



**NESNELERİN İNTERNETİNDE SERVİS KALİTESİNİN
İYİLEŞTİRİLMESİ**

Metehan GÜZEL

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TEMMUZ 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

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

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

Metehan GÜZEL
01 / 07 / 2022

NESNELERİN İNTERNETİNDE SERVİS KALİTESİNİN İYİLEŞTİRİLMESİ (Doktora Tezi)

Metehan GÜZEL

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Temmuz 2022

ÖZET

Nesnelerin İnterneti (Internet of Things / IoT) kavramı, her türlü elektronik cihazın internete bağlanarak yüksek çeşitlilikteki uygulamaları insan hayatına kazandırdığı bir geleceği öngörür. IoT’de sağlanan hizmetler ve uygulamalar IoT’nin en temel bileşenlerinden biri olan Servis Odaklı Mimari (Service Oriented Architecture / SOA) uyarınca servisler olarak özetlenir ve IoT servisi isimlendirmesi ile sunulur. IoT ağı kullanan kullanıcılara sunulan IoT servislerinin sağlanması gereken kalite standartlarına servis kalitesi (Quality of Service) ismi verilir. IoT’de servislerin QoS kısıtları altında sağlanması oldukça önemlidir. Bu tez kapsamında IoT’de sağlanan QoS seviyelerinin iyileştirilmesi problemine odaklanılmıştır. Öncelikle, IoT’de QoS sağlanması problemine dair bir kaynak taraması yapılmıştır. Belirtilen problemin çözümüne dair önerilen IoT mimarileri, yapıları ve protokolleri özetlenmiş, IoT ağı farklı katmanlarına yönelik QoS faktörleri tanımlanmış ve elde edilen tarama sonuçları özgün bir derleme makalesi olarak yayınlanmıştır. Ardından, IoT ağı nesne, ağ ve uygulama katmanlarında QoS iyileştirilmesine dair çalışmalar yapılmıştır. Yapılan ilk çalışma IoT ağı en alt katmanı olan nesne katmanında veri bütünlüğünün sağlanmasına yöneliktir. Nesne katmanında meydana gelebilecek veri kayıplarının önüne geçmek adına, uyarlamalı ağ tabanlı bulanık çıkarım sistemi ve derin öğrenme temelli özgün eksik veri öngörü modelleri önerilmiştir. Önerilen modeller gerçek bir IoT veri seti üzerinde test edilmiştir. Gözlemlendiği üzere önerilen modeller eksik verileri etkin bir şekilde doldurabilmektedir. Ağ katmanına dair yapılan çalışmada, ağ gecikmesini düşürmek adına sis hesaplama ve yazılım tanımlı ağlar teknolojilerini temel alan bir yaklaşım önerilmiştir. Gerçek bir test senaryosunda simüle edilen yöntemin öncelikli veri akışlarında ağ gecikmesini düşürdüğü gözlemlenmiştir. Uygulama katmanına yönelik yapılan çalışma kapsamında basit IoT servislerinin birleştirilip karmaşık uygulamaların elde edildiği servis birleştirme süreçlerinde QoS, enerji etkinliği ve IoT servisleri arasında adil iş yükü dağıtımının sağlanmasına odaklanılmıştır. Belirtilen hedefler altında servislerin birleştirilmesi amacıyla NSGA-II algoritması temelli özgün bir optimizasyon modeli önerilmiştir.

Bilim Kodu : 92407
Anahtar Kelimeler : Nesnelerin İnterneti, Servis Kalitesi, Nesnelerin İnterneti Servisleri, Servis Kompozisyonu, Sis Hesaplama
Sayfa Adedi : 154
Danışman : Prof. Dr. Suat ÖZDEMİR

IMPROVING QUALITY OF SERVICE IN INTERNET OF THINGS
(Ph. D. Thesis)

Metehan GÜZEL

GAZİ UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
July 2022

ABSTRACT

Internet of Thing (IoT) concept envisions a future where all electronic devices are connected via the internet to bring a wide variety of applications to human life. In accordance with the Service Oriented Architecture, functionalities provided by IoT are encapsulated as services and named as IoT services. Quality of Service (QoS) is one of the crucial components of IoT. QoS indicates the quality requirements demanded by the users of IoT networks and IoT services must meet the QoS requirements. In this thesis, the problem of improving QoS in IoT is studied. Firstly, the existing studies about QoS and IoT are reviewed. QoS aware frameworks, architectures and protocols are summarized, layer specific QoS metrics are identified and the findings are published as a survey paper. Then, separate researches focusing in perceptron, network and application layers of the IoT. The first research, which aims to improve QoS of perceptron layer of IoT, regards to improve data completeness. To overcome missing data problem, novel missing data estimation models based on Adaptive Neuro Fuzzy Inference System (ANFIS) and Deep Learning (DL) are developed and presented. The proposed models are evaluated using a real-world IoT dataset. The evaluation results indicate that the proposed models can estimate missing sensor readings in high accuracy. The second research focuses in the network layer of IoT and aims to minimize network latency. To decrease network latency, a novel approach based on Fog Computing (FC) and Software Defined Networking (SDN) is proposed. In the proposed approach FC is used to decrease overall network latency. SDN on the other hand, is used to prioritize time-critical network packages. The last study mentioned in this thesis focuses in the application layer of IoT. Service composition (SC) is the process of composing complex IoT application using basic IoT services. In the last study, the problem of providing QoS in SC process, is investigated. An energy-aware approach that can ensure fair load distribution between IoT services under QoS constraints is proposed. The SC problem is formulated as a multi-objective optimization problem and solved with a novel model that uses Non-dominated Sorting Genetic Algorithm (NSGA-II).

Science Code : 92407

Key Words : Internet of Things, Quality of Service, IoT Services, Service Composition, Fog Computing

Page Number : 154

Supervisor : Prof. Dr. Suat ÖZDEMİR

TEŞEKKÜR

Doktora çalışmam sürecinde bana yol gösteren ve beni her daim destekleyen danışman hocam Prof. Dr. Suat ÖZDEMİR'e, tez izleme komitelerinde verdikleri önerilerle çalışmalarına büyük katkılar veren komite hocalarım Prof. Dr. Sevil ŞEN ve Doç. Dr. Mehmet DEMİRCİ'ye, 2008 senesinden beri gerek lisans öğrencisi, gerek lisans üstü öğrencisi ve akademik personel olarak bir parçası olduğum Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünün üzerimde emeği olan tüm hocalarına, tez metni ve yayınlarımda bana görsel tasarlanması sürecinde destek veren arkadaşım Andaç ALTAK'a, tezin biçimlendirilmesinde ve redaksiyonunda bana yardım eden Arş. Gör. Özlem ALPAY'a, savunma sunumunda yardım eden Öğr. Gör. Aybike ŞİMŞEK DİLBAZ'a, yoğun tez sürecimde her daim desteklerini hissettiğim annem, babam ve kardeşime ve bu süreçte belki de en çok yordüğüm insan olan canım kendime sonsuz teşekkürlerimi sunarım.

Bu tez çalışmasını 1001 – Bilimsel ve Teknolojik Araştırma Projelerini Destekleme Programı kapsamında 118E212 numaralı proje ile destekleyen TÜBİTAK'a teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. NESNELERİN İNTERNETİ VE SERVİS KALİTESİ.....	7
2.1. Nesnelerin İnterneti	7
2.1.1. Öncül teknolojiler.....	8
2.1.2. IoT ağ mimarileri	10
2.1.3. IoT karakteristik özellikleri.....	13
2.1.4. IoT iletişim protokolleri	14
2.1.5. IoT uygulama alanları	17
2.1.6. Destekleyici teknolojiler	20
2.1.7. IoT geleceği.....	23
2.2. Nesnelerin İnternetinde Servis Kalitesi.....	24
2.2.1. İletişim protokolleri.....	24
2.2.2. Farklı QoS metrikleri	25
2.2.3. Kullanılan teknikler.....	28
2.3. Bölüm Değerlendirmesi ve Tartışma	40
3. NESNELERİN İNTERNETİNDE VERİ BÜTÜNLÜĞÜNÜN YAPAY SİNİR AĞLARI TEMELLİ YÖNTEMLER İLE SAĞLANMASI.....	43

	Sayfa
3.1. İlgili Çalışmalar.....	44
3.1.1. İstatistiki yaklaşımlar	44
3.1.2. Optimizasyon tabanlı yaklaşımlar	45
3.1.3. Makina öğrenmesi tabanlı yaklaşımlar	46
3.2. Veri Seti.....	49
3.3. Önerilen Yaklaşım.....	51
3.3.1. ANFIS temelli eksik veri tamamlama yaklaşımları	54
3.3.1. DL temelli eksik veri tamamlama yaklaşımı.....	61
3.4. Test Sonuçları.....	64
3.5. Değerlendirme	68
4. ÖNCELİKLENDİRME İLE FARKLI QOS SEVİYELERİNİN SAĞLANMASI.....	69
4.1. Önerilen Yaklaşım.....	70
4.2. Veri Analizi ve Senaryo	71
4.2.1. Veri kaynağı	71
4.2.2. IoT temelli mimari	73
4.2.4. Veri işlemleri.....	79
4.2.5. Veri akışları.....	82
4.2.6. Veri paketlerinin önceliklendirilmesi.....	83
4.4. Test Sonuçları.....	86
4.5. Değerlendirme	88
5. IOT UYGULAMALARININ QOS KISITLARI ALTINDA ADİL VE ENERJİ ETKİN KOMPOZİSYONU	89
5.1. İlgili Çalışmalar.....	92
5.2. Konsept Tasarımı	97
5.2.1. Ağ modeli.....	98
5.2.2. Uygulama modellemesi.....	99

Sayfa

5.2.3. QoS modellemesi	100
5.3. Önerilen Yaklaşım.....	102
5.3.1. Servis birleştirme süreci	103
5.3.2. Çok objektifli optimizasyon	108
5.3.3. NSGA-II.....	109
5.3.4. Problemin optimizasyon formülasyonu	112
5.3.5. Genetik operatörler.....	113
5.3.6. Değerlendirme	113
5.4. Test Sonuçları.....	113
5.4.1. Sonuçların QoS perspektifinden değerlendirilmesi.....	116
5.4.2. Sonuçların enerji etkinliği ve adil iş yükü dağılımı perspektifinden değerlendirilmesi	117
5.4.3. Çalışma zamanları karşılaştırılması	118
5.4.4. Değişen talep sayılarının etkileri.....	120
5.4.4. Değişen ağ büyüklüklerinin etkileri	122
5.4.5. Test sonuçlarının toplu değerlendirmesi	124
5.5. Bölüm Değerlendirmesi	124
6. SONUÇ VE ÖNERİLER.....	127
KAYNAKLAR	131
ÖZGEÇMİŞ	153

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. IoT’de uygulama katmanında kullanılan iletişim protokolleri [16].....	16
Çizelge 2.2. Nesne katmanında kullanılan QoS metrikleri	26
Çizelge 2.3. Ağ katmanında kullanılan QoS metrikleri	27
Çizelge 2.4. Uygulama katmanında kullanılan QoS metrikleri	28
Çizelge 2.5. QoS hedefli önbellekleme çalışmaları	34
Çizelge 2.6. QoS hedefli iş yükü dengeleme çalışmaları.....	39
Çizelge 3.1. Taranan eksik veri çalışmaları	48
Çizelge 3.2. Veri setinin istatistiki analiz sonuçları.....	50
Çizelge 3.3. Tanımlanan eksik veri senaryoları	51
Çizelge 3.4. GP-ANFIS modelinin en etkin parametre konfigürasyonu	59
Çizelge 3.5. SCL-ANFIS modelinin en etkin parametre konfigürasyonu	60
Çizelge 3.6. FCM-ANFIS modelinin en etkin parametre konfigürasyonu	61
Çizelge 3.7. LSTM sembol tablosu.....	62
Çizelge 3.8. DL modelinin en etkin parametre konfigürasyonu	64
Çizelge 3.9. ANFIS ve Dİ temelli eksik veri tamamlama modellerinin hata oranları ...	66
Çizelge 4.1. Test senaryosunda kullanılan istasyonlar	73
Çizelge 4.2. IoT temelli mimaride istasyonların görevleri	75
Çizelge 4.3. Kullanılan olağan dışı değer aralıkları.....	80
Çizelge 4.4. CPP öngörü modellerinin hata oranları	81
Çizelge 4.5. SP öngörü modellerinin hata oranları	82
Çizelge 4.6. Kirlenici önceliklendirme sınırları.....	84
Çizelge 4.7. Bulut sunucudaki paketlerin öncelik seviyelerine göre dağılımı.....	84
Çizelge 4.8. Sis sunuculardaki paketlerin öncelik seviyelerine göre dağılımı.....	85

Çizelge	Sayfa
Çizelge 4.9. Kirleticilerin önceliklendirme şartları.....	87
Çizelge 4.10. Test sonuçları.....	88
Çizelge 5.1. SC alanındaki çalışmalar	96
Çizelge 5.2. Notasyonlar	97
Çizelge 5.3. Geliştirilen QoS modellemesinde tanımlanan QoS ister tipleri.....	102
Çizelge 5.4. Simülasyon parametreleri	114
Çizelge 5.5. Optimal parametre konfigürasyonları	116
Çizelge 5.6. Test sonuçları.....	116
Çizelge 5.7. Farklı kritik QoS metrik sayılarında elde edilen sonuçlar	117
Çizelge 5.8. Modellerin çalışma zamanları.....	119
Çizelge 5.9. Farklı uygulama talep hızlarına dair test sonuçları	121
Çizelge 5.10. Farklı ağ büyüklüklerine dair test sonuçları	123

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. IoT Ağ Mimarileri, (a) 3 Katmanlı Konvansiyonel Mimari, (b) 4 Katmanlı Endüstriyel Mimarisi, (c) 5 Katmanlı Servis Odaklı Modern Mimari.	12
Şekil 2.2. Veri önbellekleme örnek senaryosu.....	29
Şekil 2.3. Yük dengeleme örnek senaryosu	36
Şekil 3.1. Intel Berkeley laboratuvar, algılayıcı konumları	49
Şekil 3.2. Spearman korelasyon analizi sonuçları.....	50
Şekil 3.3. Önerilen eksik veri tamamlama yaklaşımına dair akış şeması	52
Şekil 3.4. Örnek veri akışı ve eksik veri senaryolarının kullanılması.....	53
Şekil 3.5. 5 katmanlı örnek ANFIS yapısı	55
Şekil 3.6. LSTM bloğu görsel temsili	63
Şekil 4.1. İstanbul ilindeki hava kalitesi istasyonları ve konumları [230].....	74
Şekil 4.2. Tasarlanan FC temelli IoT ağ mimarisi	74
Şekil 4.3. Dördüncü istasyonda yapılan CPP öngörü modellerinin giriş ve çıkışları	77
Şekil 4.4. Yedinci istasyonda yapılan SP öngörü modellerinin giriş ve çıkışları	79
Şekil 4.5. Veri akışları a) T2, T3 b) T1(PM10), c) T1(O3), d) T1(NO ₂), f) T1(SO ₂)....	83
Şekil 5.1. Tasarlanan ağ yapısına dair örnek görsel.....	99
Şekil 5.2. Önerilen yaklaşım	103
Şekil 5.3. SC süreci görsel örnek I.....	104
Şekil 5.4. SC süreci görsel örnek II	105
Şekil 5.5. SC süreci görsel örnek III	106
Şekil 5.6. SC süreci görsel örnek IV	107
Şekil 5.7. Pareto cepheler.....	109
Şekil 5.8. Kalabalıklık mesafesi hesaplanması	110
Şekil 5.9. NSGA-II yeni popülasyon üretme prosedürü	111

SİMGELER VE KISALTMALAR

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

Simgeler	Açıklamalar
App	Uygulama
b_i	LSTM’de Giriş Kapısı Eğilim Vektörü
b_c	LSTM’de Hücre Eğilim Vektörü
b_f	LSTM’de Unutma Kapısı Eğilim Vektörü
b_o	LSTM’de Çıkış Kapısı Eğilim Vektörü
c	Servis Sınıfı
C	Bulut Sunucu
cs	S Servisinin Servis Sınıfı
C_t	LSTM’de Eski Hücre Durumu
$\bar{C}_t$	LSTM’de Yeni Hücre Durumu
EP_S	S Servisinin Enerji Profili
EP_T	T Nesnesinin Enerji Profili
F	Sis Sunucu
f_{ij}	ANFIS’te Terimlerin Ağırlıklandırılmış Toplamı
f_t	LSTM’de Unutma Kapısı
h_t	LSTM’de t Zamanına Dair Çıkış Vektörü
i_t	LSTM’de Giriş Kapısı
N	IoT Ağ
o_t	LSTM’de Çıkış Kapısı
p_{ij}	ANFIS’te Ağırlıklandırma Parametrelerinden Biri
r_{ij}	ANFIS’te Ağırlıklandırma Parametrelerinden Biri
S_T	T Tarafından Hostlanan Servislerin Listesi
S_{T,n}	T Tarafından Hostlanan n’inci Servis
SLA_{App}	App’in SLA Kontratı
SSLA_{App}	App’in SSLA Kontratı
T	IoT Nesnesi
tanh	LSTM’de Hiperbolik Tanjant Fonksiyonu

Simgeler**Açıklamalar**

q_{ij}	ANFIS'te Ağırlıklandırma Parametrelerinden Biri
$QoSV_s$	S Servisinin QoS Vektörü
$QoSA_{s,n}$	S Servisinin n'inci QoS İsteri
x_t	LSTM'de t Zamanına Dair Giriş Vektörü
W_c	LSTM'de Hücre Ağırlık Matrisi
W_f	LSTM'de Unutma Kapısı Ağırlık Matrisi
WF_{App}	App'in Birleştirilmesine Dair Taslak (Workflow)
W_i	LSTM'de Giriş Kapısı Ağırlık Matrisi
$w_{i,j}$	ANFIS'te Atış Gücü
$\bar{w}_{i,j}$	ANFIS'te Normalize Edilmiş Atış Gücü
W_o	LSTM'de Çıkış Kapısı Ağırlık Matrisi
σ	LSTM'de Mantıksal Sigmoid Fonksiyonu
μ_A	ANFIS'te Üyelik Fonksiyonu

Kısaltmalar**Açıklamalar**

5G	Fifth Generation / Beşinci Nesil
6G	Sixth Generation / Altıncı Nesil
AANN	Auto-Associative Neural Network / Otomatik İlişkilendirmeli Sinir Ağı
ACO	Ant Colony Optimization / Karınca Kolonisi Optimizasyonu
AMQP	Advanced Message Queuing Protocol / Gelişmiş Mesaj Kuyruklama Protokolü
ANFIS	Adaptive Neuro Fuzzy Inference System / Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi
ANN	Artificial Neural Network / Yapay Sinir Ağı
AR	Accept Ratio / Kabul Oranı
BSOA	Backtracking Search Optimization Algorithm / Geri Adımlı Arama Optimizasyonu Algoritması
CBF	Content Based Forwarding / İçerik Tabanlı Yönlendirme
CC	Cloud Computing / Bulut Hesaplama
CIR	Cluster Influence Range / Küme Etki Menzili
CN	Number of Clusters / Küme Sayısı

Kısaltmalar**Açıklamalar**

COAP	Constraint Application Protocol / Kısıtlı Uygulama Protokolü
CP	Convex Programming / Konveks Programlama
CPP	Cross Pollutant Prediction / Kirleticiler Arası Öngörü
CPS	Cyber-Physical Systems / Siber Fiziksel Sistemler
CRAN	Cloud RAN / Bulut tabanlı RAN
CRAPSO	Chaotic Reduced Adaptive Particle Swarm Optimization / Kaotik Eksiltilmiş Uyarlanabilir Parçacık Sürü Optimizasyonu
CSA	Cuckoo-Search Algorithm / Guguk Kulu Arama Algoritması
DDS	Data Distribution Services / Veri Dağıtım Servisleri
DL	Deep Learning / Derin öğrenme
DRL	Deep Reinforcement Learning / Derin Pekiştirmeli Öğrenme
DQLBA	Deep Q-learning Based Algorithm / Derin q-öğrenme Temelli Algoritma
DPBACO	Dual Re-Populated Bayesian Ant Colony Optimization / Çift Popülasyonlu Bayes Karınca Kolonisi Algoritması
EASE	Energy Aware and Latency Guaranteed Dynamic Resource Caching / Enerji Farkında ve Gecikme Garantili Dinamik Kaynak Ayrılması
EC	Edge Computing / Uç Hesaplama
ELM	Extreme Learning Machine / Aşırı Öğrenme Makinası
EM	Expected Maximization / Beklenti Maksimizasyonu
Expo	Exponent / Üs
EQSA	Energy-Centered and QoS-Aware Services Selection Algorithm / Enerji-Odaklı ve QoS-farkında Servis Seçim Algoritması
FC	Fog Computing / Sis Hesaplama
FCM	Fuzzy c-Means Clustering / Bulanık c-Means Kümeleme
FIS	Fuzzy Inference System / Bulanık Çıkarım Sistemi
FNDSA	Fast Non-Dominated Sorting Algorithm / Hızlı Domine Edilmeyen Sıralama Algoritması

Kısaltmalar**Açıklamalar**

FRAN	Fog RAN / Sis tabanlı RAN
FSCA-EQ	Fast Energy-Centered and QoS –Aware Service Composition / Hızlı Enerji Merkezli ve QoS Farkında Servis Birleştirme
GA	Genetic Algorithm / Genetik Algoritma
GKR	Gaussian Kernel Regression / Gauss Çekirdek Regresyonu
GP	Grid Partitioning / Izgara Bölümleme
GIoT	Green IoT / Yeşil IoT
GWO	Grey Wolf Optimization / Gri Kurt Optimizasyonu
HDCC	High Dimentional Data Clustering / HDDC
IETF	Internet Engineering Task Force / Internet Mühendisliği Görev Gücü
IIoT	Industrial IoT / Endüstriyel IoT
ILP	Integer Linear Programming / Tamsayı Lineer Programlama
IMFT	Input Membership Function Type / Giriş Üyelik Fonksiyon Tipi
IoT	Internet of Things / Nesnelerin İnterneti
IS	Isolation Forest
HIoT	Health IoT / Sağlıksal IoT
HTTP	Hyper Text Transfer Protocol / Hiper Metin Gönderim Protokolü
k-NN	k-Nearest Neighbour / k-En Yakın Komşu
LEARN	Latency Aware Popular Resource Re-caching / Gecikme Farkında Tekrarlı Popüler Veri Önbelleklemesi
LM	Laprangian Multiplier / Laprangian Çarpımı
LP	Linear Programming / Lineer Programlama
M2M	Machine-2-Machine / Cihazlar Arası
MCDM	Multi-criteria Decision Making / Çok Kriterli Karar Verme
MCE	Multi-Cloud Environment / Çok Bulut Sunuculu Ortam
MDP	Markov Decision Process / Markov Karar Süreci
MIV	Minimum Improvement Value / Minimum Gelişme Değeri

Kısaltmalar**Açıklamalar**

MLE	Maximum Likelihood Estimation / En Büyük Olabilirlik Kestrimi
MLPN	Multi-Layer Perceptron Networks / Çok Katmanlı Perceptron Ağlar
MNI	Maximum Number of Iterations / Maksimum Yineleme Sayısı
MOGA	Multi-Objektive Genetic Algorithm / Çok Objektifli Genetik Algoritma
MST	Minimum Spanning Tree / Minimum Tarama Ağacı
MQTT	Message Queuing Telemetry Transport / Mesaj Kuyruklamalı Telemetri Gönderimi
NMF	Number of Membership Functions / Üyelik Fonksiyonu Sayısı
NSGA-II	Non-dominated Sorting Genetic Algorithm / Hakimiyet Sıralama temelli Genetik Algoritma
OMFT	Output membership Function Type / Çıkış Üyelik Fonksiyon Tipi
PA	Pulse Algorithm / Nabız Algoritması
PCBF	Priority Enabled Content Based Forwarding / Önceliklendirme Etkinleştirilmiş İçerik Tabanlı Yönlendirme
PCBF-SDN	Priority Enabled Content Based Forwarding via SDN / SDN ile Önceliklendirme Etkinleştirilmiş İçerik Tabanlı Yönlendirme
PNN	Probabilistic Neural Networks / Olasılıksal Sinir Ağları
PRSNN	Partial Recurrent Spike Neural Network / Kısmi Yinelemeli Yükselen Yapay Sinir Ağları
PSO	Particle Swarm Optimization / Parçacık Sürü Optimizasyonu
RAN	Radio Access Networks / radyo Erişimli Ağlar
RFID	Radio Frequency Identification / Radyo Frekansı Tanımlama
RMSE	Root-Mean-Square Error / Kök Hata Oranı
RR	Reject Ratio / Ret Oranı
SAIoT	Scalable Anomaly-Aware Approach / Ölçeklenebilir Anomali Farkında Yaklaşım
SAW	Simple Additive Weighting / Basit Ağırlıklandırma Toplama

Kısaltmalar**Açıklamalar**

SC	Service Composition / Servis Birleştirme
SCL	Subtractive Clustering / Eksiltmeli Kümeleme
SDN	Software Defined Networking / Yazılım Tanımlı Ağlar
SF	Scaling Factor / Ölçekleme Faktörü
SFLA	Shuffled Frog Leaping Algorithm / Karıştırılmış Kurbağa Zıplama Algoritması
SLA	Service Level Agreement / Servis Seviyesi Anlaşması
SOA	Service Oriented Architecture / Servis Odaklı Mimari
SOM	Self-Organizing Maps / Kendini Organize Eden Haritalar
SP	Spatial Prediction / Konumsal Öngörü
SVR	Support Vector Regression / Destek Vektör Regresyonu
TCP	Transport Control Protocol / Gönderim Kontrol Protokolü
TLBO	Teaching Learning Based Optimization / Öğretme Öğrenme Temelli Optimizasyon
UDP	User Datagram Protocol / Kullanıcı Datagram Protokolü
QoS	Quality of Service / Servis Kalitesi
XMPP	Extensible Messaging and Presence Protocol / Genişletilebilir Mesajlaşma ve Mevcudiyet Protokolü
WOA	Whale Optimization Algorithm / Balina Optimizasyon Algoritması
WSN	Wireless Sensor Networks / Kablosuz Algılayıcı Ağlar

1. GİRİŞ

Nesnelerin İnterneti (Internet of Things / IoT) kavramının önerdiği yaklaşım internete bağlanabilen tüm elektronik cihazların kendilerine ait IP adresleri ile internet üzerinden birbirleri ile iletişimde olmaları ve ortak hedeflere yönelik çalışmalarıdır [1-4]. IoT ağı oluşturan elektronik cihazlara literatürde nesne (thing, object) ismi verilir. Nesneler temel algılama veya bir işlevi yerine getirme, ağ üzerinden iletişim kurma ve hesaplama yeteneklerine sahiptir [1]. IoT kavramının önerdiği üzere nesneler bu yetenekleri, ortak hedefleri yerine getirmek için insan müdahalesini minimize edecek şekilde kullanır [2]. Günlük kullanılan elektronik cihazlar (telefon, tablet, bilgisayar), elektronik ev eşyaları, algılayıcılar (sensor), işleticiler (actuator) gibi tüm cihazlar internete bağlı oldukları sürece IoT ağların bir parçası olabilmektedir. Bu sebeple IoT kavramının insan hayatına yüksek seviyede nüfuz etmesi ve insan hayatını devrimsel seviyede değiştirmesi beklenmektedir [3]. IoT kavramı, IoT ağı oluşturan nesneleri, nesnelerin birbiri ile bağlanmasını sağlayan ağ cihazlarını ve altyapılarını, nesneler arası kurulan cihazlar arası (Machine-2-Machine / M2M) karakterdeki bağlantıyı ve sunulan anlamlı (semantic) hizmetleri kapsayan geniş bir kavramdır. Günümüzde IoT teknolojileri üstüne inşa edilmiş uygulamalar birçok alanda kullanılmaktadır. Akıllı evler / ortamlar (evler, şehirler, kampüsler) [5-8], akıllı tarım uygulamaları [9-12], Endüstri 4.0 temelli endüstriyel üretim tesisleri [13, 14], çevre takip uygulamaları [15] IoT temelli uygulamalarının kullanım alanlarına birkaç örnektir.

IoT kavramının yapı taşlarından biri olan Servis Odaklı Mimari (Service Oriented Architecture / SOA) uyarınca IoT'de sunulan işlevler servisler olarak özetlenir [2]. Bu özetlemenin amacı IoT'nin heterojen yapısıdır. IoT nesneleri farklı donanımlara sahiptir ve çeşitli iletişim protokolleri ile birbirine bağlanır. Bu çeşitlilik ortamında nesnelerin sunduğu işlevlerin ortak bir formatta özetlenmesi ağın ortak çalışabilme yeteneğini önemli bir seviyede artırır. IoT'nin bir diğer yapı taşı Servis Kalitesi (Quality of Service / QoS) kavramıdır. QoS, IoT ağ tarafından sağlanan bir IoT servisinin kalitesini gösteren bir ölçüttür. IoT'de donanım ve protokol bazında gözlemlenen çeşitlilik QoS bazında da gözlemlenir [16, 17]. Farklı IoT uygulamalarının farklı QoS istekleri vardır. Örnek vermek gerekirse, gerçek zamanlı bir IoT uygulamasında servis gecikmesi kritik bir QoS isteri iken bir algılama uygulamasında algılanan değerin doğruluğu daha kritiktir. Bir uygulamanın veya servisin başarılı bir şekilde sağlandığını kabul etmek için, ilgili servisin veya uygulamanın QoS isteklerine uygun bir şekilde sağlanması gereklidir [17]. Aksi takdirde,

uygulama veya servis sağlanmış sayılmaz. Bu sebeple, IoT’de QoS sağlanması konusunda çalışmalar hem akademi dünyasında hem de sektörde sıkça yapılmaktadır.

Bu tez çalışmasında, IoT’de QoS sağlanması problemi üzerine çalışılmıştır. Öncelikle bu alanda yapılan çalışmalar incelenmiştir. Bu konuda yapılan inceleme prensip olarak kabul ettiğimiz QoS ister çeşitliliğini gözlemleyebilmemizi sağlamıştır [16]. Ek olarak, inceleme sonuçları IoT ağ mimarisinin farklı katmanlarında gözlemlenen QoS isterlerinin farklı olduğu gözlemlenmiştir [16]. Dolayısıyla, IoT’de QoS sağlanabilmesi için ağın farklı katmanlarında QoS sağlanmasının gerekli olduğu ortaya çıkmıştır. Bu çıkarımlar uyarınca ağın farklı katmanlarında QoS sağlanmasına dair üç ana problem belirlenmiştir ve bu problemlerin her birine dair çalışmalar yapılmıştır. IoT ve IoT’de QoS sağlanması konusunda yapılan inceleme ve inceleme sonucu hazırlanan ve yayınlanan derleme makalesinin [16] literatüre yaptığı katkılar aşağıda sıralanmıştır.

- Yapılan incelemede IoT’de QoS desteğinin sağlanması konusunda yapılan çalışmalar QoS sağlanması için önerilen ağ mimarileri ve yapıları, iletişim protokolleri ve algoritmaları, IoT ağın farklı katmanlarına odaklanan çözümler olmak üzere üç ana bakış açısından incelenmiş ve gruplandırılmıştır.
- Yapılan incelemede doğrudan veya dolaylı olarak QoS sağlanmasına etki eden güncel çalışmalar kullanılmıştır.
- İnceleme sonucu gözlemlenen QoS isterleri kategorize edilmiş ve sunulmuştur.
- İnceleme sonucu gözlemlenen güncel problemler ve geleceğe yönelik olası çalışma önerileri verilmiştir.

IoT mimarileri üzerine yapılan incelemeler IoT ağ yapısının üç katmandan oluştuğunu göstermektedir. Çözülen ilk problem IoT mimarisinin en alt katmanı olan nesne katmanında QoS sağlanmasına dairdir. Nesne katmanında talep edilen QoS isterlerinden biri veri bütünlüğüdür [16]. IoT ağlar en ücra konumlara kadar uzanır ve genellikle düşük maliyetli cihazlardan oluşur. Bu sebeple, donanım hatalarından ve iletişim kesilmelerinden kaynaklı veri kaybı problemlerine sıkça rastlanır [18]. Bu durumlar yaygın olsa dahi, veri kaybı bu verileri kullanan IoT uygulamalarında servis kalitesi düşüşüne sebep olabilmektedir. Bu problemin çözümü için Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi (Adaptive Neuro Fuzzy Inference System / ANFIS) ve Derin Öğrenme (Deep Learning / DL) temelli eksik veri tamamlama modelleri geliştirilmiştir. Sıcaklık, nem oranı ve ışık yoğunluğu tipinde veri

toplayan Intel Berkeley Research Lab veri seti üzerinden önerilen modeller test edilmiştir. IoT ağı nesne katmanında veri bütünlüğünün sağlanması için yapılan ANFIS ve DL temelli veri tamamlama yaklaşımlarından oluşan çalışmanın [18] literatüre katkıları şunlardır:

- Eksik her veri tipi için ilgili senaryolar oluşturulmuştur.
- Her eksik veri senaryosuna dair üçü ANFIS, biri DL temelli olmak üzere toplam dört eksik veri tamamlama modeli geliştirilmiştir.
- Mevzubahis her model Intel Berkeley Research Lab veri seti kullanılarak ayrı ayrı test edilmiştir.

IoT ağı ikinci katmanı nesneler arası iletişimin sağlandığı ağ katmanıdır. Yapılan inceleme sonucunda ağ katmanında en çok karşılaşılan QoS isterinin gecikme olduğu gözlemlenmiştir. Bu sebeple ikinci problem olarak seçilen ağ katmanında düşük gecikme sağlanması problemine bir çözüm önerilmiştir. Önerilen yaklaşım Yazılım Tanımlı Ağlar (Software Defined Networking / SDN) ve Sis Hesaplama (Fog Computing / FC) teknolojilerini kullanmaktadır. Yaklaşım uyarınca gecikme isterinin kritik olduğu paketlerin önceliklendirilerek, ağ kaynaklarının kritik veri akışlarına öncelikli olarak sağlanmasına dayanır. Bu amaçla, paket önceliklendirme için SDN teknolojisi, genel olarak ağ yönetilebilirlik seviyesinin ve ağ verimliliğinin artması için de FC kavramı kullanılmıştır. Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı'nın Hava Kalitesi İzleme Ağı'nın İstanbul ilindeki hava kalitesi takip istasyonlarının konumları kullanılarak FC temelli bir IoT ağ tanımlanmış, ağ üzerine bir hava kalitesi öngörü senaryosu oturtulmuştur. Senaryo uyarınca, sis düğümleri ve nesneler bir saat sonraki hava kalitesi seviyesini öngörürler. Yüksek hava kirliliğini işaret eden öngörü sonuçları IoT ağ üzerinden önceliklendirilerek gönderilir. Bu yaklaşımda, daha kritik paketlerin ağ kaynaklarını daha öncelikli kullanması ve dolayısıyla daha düşük gecikme ile hedefe varması amaçlanmıştır. Senaryonun gerçekçi olması için istasyonlarının 2019 yılında yaptığı gözlemler kullanılarak bir veri akışı oluşturulmuştur. Çalışmanın [19] literatüre katkıları aşağıda verilmiştir:

- Çalışma bildiğimiz kadarıyla FC temelli IoT ağlarda SDN destekli veri önceliklendirme konusunda yapılan dair ilk çalışmalardan biridir.
- Çalışma kapsamında veri önceliklendirme ile farklı QoS seviyelerinin sağlanmasına dair bir model önerilmiştir.

- Önerilen modelin test edilmesi hava kalitesi öngörüsü senaryosu kullanılmış. Gerçekçi bir ağ modeli ve veri akışı oluşturulmuştur.

Çözülen üçüncü problem IoT ağın servis/uygulama katmanına yöneliktir. Nesneler kaynakları (enerji, hesaplama, depolama) kısıtlı cihazlardır. Bu sebeple nesnelerin sağladığı işlevler genellikle basittir. Ancak, IoT ağın kullanıcıları çok daha karmaşık işlevler talep edebilir. Bu ve benzeri durumlarda, basit IoT servisleri birleştirilip talep edilen işlevi yerine getirebilecek karmaşık servisler kullanılır. Bu operasyona servis birleştirme (Service Composition / SC) süreci denir. SC sürecinde basit servisler, kullanıcının istediği işlevi sağlayacak şekilde birleştirilerek talep edilen uygulama elde edilir. IoT’de SC sürecinde ele alınması gereken en önemli kriter QoS’tir. SC problemi kullanıcının QoS isterlerine uyacak şekilde çözümlenmelidir. QoS isterlerinin sağlanması gerekli olsa da günümüzde artık yeterli gelmemektedir. Yeşil IoT (Green IoT / GIoT) kavramı enerji etkin ve sürdürülebilir IoT ağların geliştirilmesini öngören bir kavramdır. SC problemi çözülürken enerji etkinliği göz önünde bulundurulmalıdır. Literatürde sadece QoS ve hem QoS hem enerji etkinliği hedeflerine sahip SC yaklaşımları bulunmaktadır. SC problemi bu iki hedefle NP-hard karmaşıklıkta bir problemdir ve bu sebeple genellikle optimizasyon algoritmaları ile çözülür. IoT’de QoS sağlanması alanında yapılan inceleme [16] sonucunda iş yükü paylaşımı (load-balancing) tekniğinin daha yüksek QoS sağlanmasına yaptığı katkı görülmüştür. Adil iş yükü paylaşımı genel olarak ağ verimliliğini arttırmakla beraber, sıklıkla kullanılan nesnelerde olabilecek donanımsal problemlerin de önüne geçebilir [17]. Bu sebeple, yapılan üçüncü çalışmada QoS ve enerji etkinliği objektifleri ile yürütülen SC sürecine ek hedef olarak adil iş yükü dağılımı objektifi eklenerek QoS-farkında, enerji etkin ve adil bir SC yaklaşımı önerilmiştir. IoT’de SC problemi üç objektifli bir optimizasyon problemi olarak formüle edilmiş ve Hakimiyet Sıralama temelli Genetik Algoritma (Non-dominated Sorting Genetic Algorithm / NSGA-II) ile çözülmüştür. Çalışmanın [17] literatüre yaptığı katkılar aşağıda verilmiştir:

- IoT’de SC probleminde adil iş yükü dağılımı problemi incelemiştir. Adil iş yükü IoT’de SC probleminin bir objektifi olarak tanımlanmıştır. Bildiğimiz kadarıyla, IoT’de SC probleminde IoT servisleri arasında adil iş yükü sağlanmasına dair yapılan ilk çalışma tarafımızdan yapılan çalışmadır.
- Toplam enerji tüketimi IoT’de SC probleminin bir objektifi olarak tanımlanmıştır.

- Bildiğimiz kadarıyla bu çalışma IoT’de SC probleminde QoS, enerji etkinliği ve IoT servisleri arasında adil iş yükü dağılımı objektifleri sahip ilk uygulama bu çalışmadır.
- IoT’de QoS-farkında, enerji etkin ve adil SC problemi için NSGA-II temelli bir yaklaşım önerilmektedir.

Bu tez kapsamında öncelikle IoT’de QoS sağlanması konusunda detaylı bir literatür taraması yapılmıştır. Literatür taramasının sonuçları Bölüm 2’de verilmiştir. Literatür taraması süreci sonunda, ağın farklı katmanlarında talep edilen QoS isterlerine göre üç problem belirlenmiştir. İlk problem IoT’nin nesne katmanına dairdir. Nesne katmanında talep edilen QoS isterlerinden veri bütünlüğü isterinin sağlanması için eksik verileri tahmin edip tamamlayan ANFIS ve DL temelli yaklaşımlar tanımlanmıştır. Bu yaklaşımlar gerçek bir IoT veri seti üzerinde test edilmiştir. Bu probleme ve problemin çözümüne dair detaylı bilgi Bölüm 3’te verilmiştir. Literatür taraması sonucunda görülmüştür ki, IoT’nin ağ katmanında en çok talep edilen QoS isteri ağ gecikmesine dairdir. Ağ gecikmesinin kritik uygulamalarda düşürülmesi için önceliklendirme tabanlı bir yaklaşım önerilmiştir. Yaklaşımın uygulanması için SDN ve FC kavramları kullanılmıştır. Testlerde gerçekçi sonuçlar almak adına Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı’nın Hava Kalitesi İzleme Ağını oluşturan hava kalitesi izleme istasyonlarının konumları ve verileri kullanılarak gerçekçi bir ağ yapısı ve veri akışı elde edilmiş ve önerilen yaklaşımın test edilmesi için kullanılmıştır. Ağ katmanı problemine ve probleme önerilen yaklaşıma dair detaylı bilgi Bölüm 4’te verilmiştir. Çözülen üçüncü problem IoT ağın en üst katmanı olan servis/uygulama katmanına yöneliktir. Bu katmanın özelleşmiş bir problemi olan SC probleminin çözümüne dair NSGA-II temelli QoS-farkında, enerji etkin ve IoT servisleri arasında adil iş yükünü sağlayan bir yaklaşım önerilmiştir. Uygulama katmanı problemine dair detaylı bilgi Bölüm 5’te verilmiştir. Bölüm 6’da bu tez çalışmasında elde edilen bulgular ve sonuçlar özetlenmiş ve geleceğe dair öneriler verilmiştir.



2. NESNELERİN İNTERNETİ VE SERVİS KALİTESİ

Bu bölümde, IoT ve IoT’de QoS sağlanması konusunda yapılan inceleme sonucunda elde edilen çıkarımlar paylaşılacaktır. Öncelikle IoT kavramının temelleri, ardından IoT’de QoS sağlanmasına dair literatürde yapılan çalışmalardan bahsedilecektir.

2.1. Nesnelerin İnterneti

IoT kavramı günlük elektronik cihazların tanımlama, çevresini gözlemleme, iletişim kurma ve veri işleme yetenekleri ile donatılması ve ortak görevleri yerine getirmek için kullanılmasına önerir [1, 3]. IoT’nin temelini oluşturan fikir parça parça ele alındığında çok da yeni değildir [1]. IoT’de nesne adı verilen elektronik cihazlar ortak bir görevi yerine getirmek için beraber çalışır ve bu süreçte iletişim yeteneğine sahip nesneler makinalar arası M2M iletişim kurar. Ortak çalışma yaklaşımı kablosuz algılayıcı ağlarda (Wireless Sensor Networks / WSN) ve siber fiziksel sistemlerde (Cyber Physical Systems / CPS) bir süredir kullanılmaktadır. İletişim yeteneğine sahip nesneler ise radyo frekansı tanımlama (Radio Frequency Identification / RFID) etiketlerinin geliştirilmesinden beri çok çeşitli iş alanlarında kullanılmaktadır. Benzer bir şekilde M2M iletişim internetin ilk çıkışından beri olan bir teknolojidir, en basitinden konvansiyonel internet mimarisinde alıcı ile sunucu arasındaki iletişim M2M karakteristiktir. Ancak IoT kavramı bahsedilen teknolojilerden belli başlı kritik noktalarda ayrılmaktadır.

- CPS ve WSN karakterindeki ağlarda cihazlar ve iletişim protokolleri çeşitlilikten uzaktır. Çoğunlukla tek bir çeşit iletişim protokolü üzerinden, sayıca çok olsa dahi kısıtlı çeşitlilikte, belirli bir göreve özel üretilmiş cihazlar kullanılır. IoT’de ise cihaz çeşitliliği ve protokol çeşitliliği oldukça yüksektir. IoT internete bağlanma yeteneği olan tüm cihazların kullanımını öngörür. Bu cihazlar günlük kullanılan akıllı ev aletlerinden telefonlara, arabalardan RFID etiketlerine ve algılayıcılara kadar ulaşan geniş bir çeşitlilik yelpazesine sahiptir [1-4]. Bu çeşitlilik sayesinde, IoT ağlar belirli bir göreve özel geliştirilen öncüllerinden daha geniş bir uygulama türü spektrumuna sahiptir.
- IoT nesnelerinin sahip oldukları algılama ve veri işleme yetenekleri, nesnelerin bulundukları ortamda akıllı kararlar verebilmelerini mümkün kılar. Ek olarak, iletişim yetenekleri ile diğer nesneler ile iletişim kurup bilgi paylaşımı ile aldıkları kararları diğer nesneler ile koordine edebilirler. Bu yetenekler sayesinde merkezi bir sunucu, dolayısıyla

merkezi sunucuya ulaşmak için gerekecek ek yüksek iletişim maliyeti olmadan IoT nesneleri bulundukları ortamda akıllı kararlar verebilir [2].

- CPS ve WSN'ler oldukça yerel teknolojilerdir. Dolayısıyla belirli coğrafi bir bölgede, düşük çeşitlilikte uygulamalarda kullanılabilirler. Ancak IoT kavramının öngördüğü vizyonda internete erişen tüm nesneler insan müdahalesi gerektirmeden, kendi kendilerine organize olarak çalışabilir. Bu sebeple IoT ağlarda yapılacak uygulamalar öncül teknolojilerine göre oldukça çeşitli ve ulaşılabilir [3, 4].
- IoT'nin aktif bir çalışma konusu olmasının arkasındaki en büyük gerekçe IoT ağların insan hayatına nüfuz etme yeteneği ve potansiyelidir. Günümüzde elektronik cihazlar insan yaşam ortamının her yerine erişmiş durumdadır. Ortalama bir insan her anında bir elektronik cihaza yakındır. Şüphesizdir ki elektronik cihazlarla bu kadar iç içe girmiş bir yaşam tarzında, elektronik cihazların IoT ağların bir nesnesine dönüştürülmesi ve insan hayatını geliştirecek uygulamalar için kullanılması fikri değer ve ilgi görecektir. Bu gerekçeden yola çıkarak IoT konusunda çalışmalar yapılmakta, akıllı ortamlar, akıllı sağlık ve benzeri çok sayıda alanda IoT ağların insan hayatını geliştirmek için hali hazırda kullanılmaktadır [1, 3, 4].

2.1.1. Öncül teknolojiler

IoT kavramının gerçek dünyaya uygulanabilmesi için bir takım öncül teknolojilerin geliştirilmesi gerekmiştir. Bu teknolojiler adresleme ve tanımlama, algılama, iletişim ve hesaplama [2] teknolojileridir. Bu teknolojilere ek olarak, web servisleri alanından ilham alınarak geliştirilen SOA, IoT ağların geliştirilmesinde önemli bir kilometre taşı olmuştur [1, 2].

Adresleme ve tanımlama teknolojileri

Tanımlama teknolojileri IoT nesneleri ile servislerin eşleştirilmesini açısından önem taşır ancak yeterli değildir. Tanımlama için kullanılan yöntemler genellikle yereldir ve ağa bağlı tüm cihazları ayırt etmek için yeterli gelmez. Bu sebeple IoT nesnelerini ayırt etmek için adresleme ve tanımlama beraber kullanılır [4]. Tanımlama için elektronik ürün kodları ve yaygın kodlar kullanılmaktadır [20]. Adresleme ve tanımlama teknolojilerinin kullanılması ile birlikte, IoT nesnelerinin ağ üzerinde hatasız bir şekilde kimliklendirilmesi yapılabilmektedir.

Algılama teknolojileri

IoT nesnelerinin en temel yeteneklerinden biri çevrelerini algılayıcılar yardımı ile algılamaktır [4]. Algılanan veri, belirlenen merkezlerde toplanır ve algılanan ortamın sanal bir temsili oluşturulur [3]. Bir servis veya uygulama tarafından talep geldiğinde temsil üzerinde analiz yapıp sonuçlar ilgili adrese gönderilir [4]. Çoğu uygulamada doğru karar verilebilmesi algılanan verinin doğruluğuna bağlıdır, bu sebeple algılama teknolojileri IoT ağlar için kritik öneme sahiptir.

İletişim teknolojileri

Nesnelerin algıladıkları verileri ilgili adreslere göndermek ve servis taleplerine cevap vermek adına ağ üzerinden iletişim kurma yeteneğine sahip olması gereklidir. IoT ağlar dünyanın ücra köşelerine kadar yayılmaktadır. Bu sebeple, gürültü ve veri kaybının yüksek olduğu iletişim kanallarında çalışabilecek iletişim teknolojileri gereklidir. Aynı zamanda, IoT nesnelerinin büyük bir kısmı enerji ihtiyaçlarını bataryalarından karşıladıkları için iletişim teknolojilerinin düşük enerji seviyelerinde çalışabilmesi gereklidir [4].

Heterojen yapısı sebebiyle IoT’de yüksek çeşitlilikte iletişim teknolojileri kullanılmaktadır. Kısa mesafeli iletişimde ZigBee, Bluetooth, WiFi ve OWC sıkça kullanılmaktadır [4, 21]. Uzun mesafede ise 5G, LTE ve LPWAN türevleri sıkça kullanılmaktadır [21, 22]. Önümüzdeki dönemde ise IoT ağların 6G bağlantı teknolojisini kullanarak çok daha yaygınlaşması beklenmektedir [23, 24].

Hesaplama teknolojileri

Nesnelerin hesaplama yapabilme yetenekleri bulundurdıkları işlem birimlerinden (FPGA, mikrokontrolcü, mikroişlemci vb.) gelir [4]. İşlem birimlerinin boyut olarak küçülmesi, maliyetlerinin düşmesi ve daha düşük enerjide çalışabilmelerinden ötürü IoT ağların boyutları ve sayısı artmıştır. Örnek vermek gerekirse IoT’de en çok kullanılan elektronik platformlardan birkaçı Arduino, Raspberry Pi, BeagleBone Black, Intel Galileo, Zolertia olarak listelenebilir [25, 26].

İşletim sistemleri IoT nesnelerinin performanslı çalışabilmesi için oldukça önemlidir. Bahsedildiği üzere IoT nesneleri düşük kaynaklara sahiptir. Bu sebeple işletim sisteminin

düşük kaynaklarda ve düşük enerji seviyesinde etkin bir şekilde çalışması gereklidir. Aynı zamanda, IoT'nin heterojen karakteristiğinden dolayı esnek bir yapıda olması da gereklidir [26]. Bu şartlara uygun işletim sistemleri geliştirilmiş ve IoT nesnelerinde kullanılmıştır. Örnek vermek gerekirse, Contiki, RIOT, RTOS, TinyOS, LiteOS ve Zephyr en çok kullanılan işletim sistemlerinden birkaçıdır [25-27].

Servis odaklı mimari

IoT kavramının en temel yapıtaşlarından biri SOA'dır. SOA uyarınca, IoT'de nesnelerin sunduğu işlevsellikler servis olarak kullanıcılara sunulur. SOA, IoT öncesi bulut bilişimde ve WSN'lerde başarılı bir şekilde uygulanmıştır [14]. Bu sebepten ötürü, IoT kavramının ilk ortaya çıkışından beri en karakteristik özelliklerinden biri olarak kabul edilmiştir. Tanım olarak bir servis, bir özellik veya fonksiyonu sağlamak adına bulunan, veri ve veriye dair davranış bütünüdür [14]. SOA sayesinde karmaşık ve tekil sistemler, basit ve çok parçadan (servis) oluşan sistemlere parçalanabilmektedir [3, 14, 28].

2.1.2. IoT ağ mimarileri

IoT ağlar katmanlı mimariye sahiptir ancak ağlarda farklı sayıda katmanlar bulunabilmektedir. IoT kavramının ortaya çıktığı ilk dönemlerinde 3 katmanlı bir mimari takip edilmekteydi. Bu katmanlara aşağıdaki gibi sıralanabilir.

- Algılayıcı Katman / Nesne Katmanı (Perceptron Layer) nesnelerin ve tanımlayıcıların bulunduğu mimarinin en alt katmanıdır.
- Ağ Katmanı (Network Layer) ağ mimarisinin alt ve üst katmanlarını birbirine ve algılayıcı katmandaki nesneleri birbirine bağlayan, iletişimden sorumlu orta seviyedeki katmandır.
- Uygulama Katmanı (Application Layer) anlamsallaştırmanın sağlandığı ve ağın kullanıcılarına uygulamaların sunulduğu katmandır.

Erken IoT ağlarda iş gören 3 katmanlı mimarinin zamanla birlikte yetersiz kaldığı görülmüştür. IoT cihazlarının donanım ve bağlantı çeşitliliği arttıkça IoT ağ üzerinde uygulama geliştirmek programcılar için daha zor bir hale gelmiştir. Nesnelerin ve nesnelerin sunduğu işlevlerin servis olarak soyutlanması gerekliliği açığa çıkmıştır. Bu sebeple SOA'nın daha yüksek seviyede desteklendiği mimariler önerilmiştir ve SOA ile direkt ilgili

katmanlar mimarilere eklenmeye başlamıştır. Bunun ilk örneklerini endüstriyel IoT ağlarda görmekteyiz. Endüstriyel ortamlarda genellikle aşağıda katmanları verilen 4 katmanlı bir mimari takip edilir.

- Algılayıcı Katman (Sensing Layer) algılayıcılardan ve nesnelerden oluşan en alt seviyedeki katmandır.
- Ağ Katmanı (Network Layer) algılama katmanını servis katmanına bağlayan ağ elemanlarını içeren katmandır.
- Servis Katmanı (Service Layer) servislerin seçilimi ve uygulamaların servis birleştirme ile oluşturulduğu katmandır.
- Arayüz Katmanı (Interface Layer) son kullanıcının ağ fonksiyonlarına ulaştığı ve uygulamaları kullandığı katmandır.

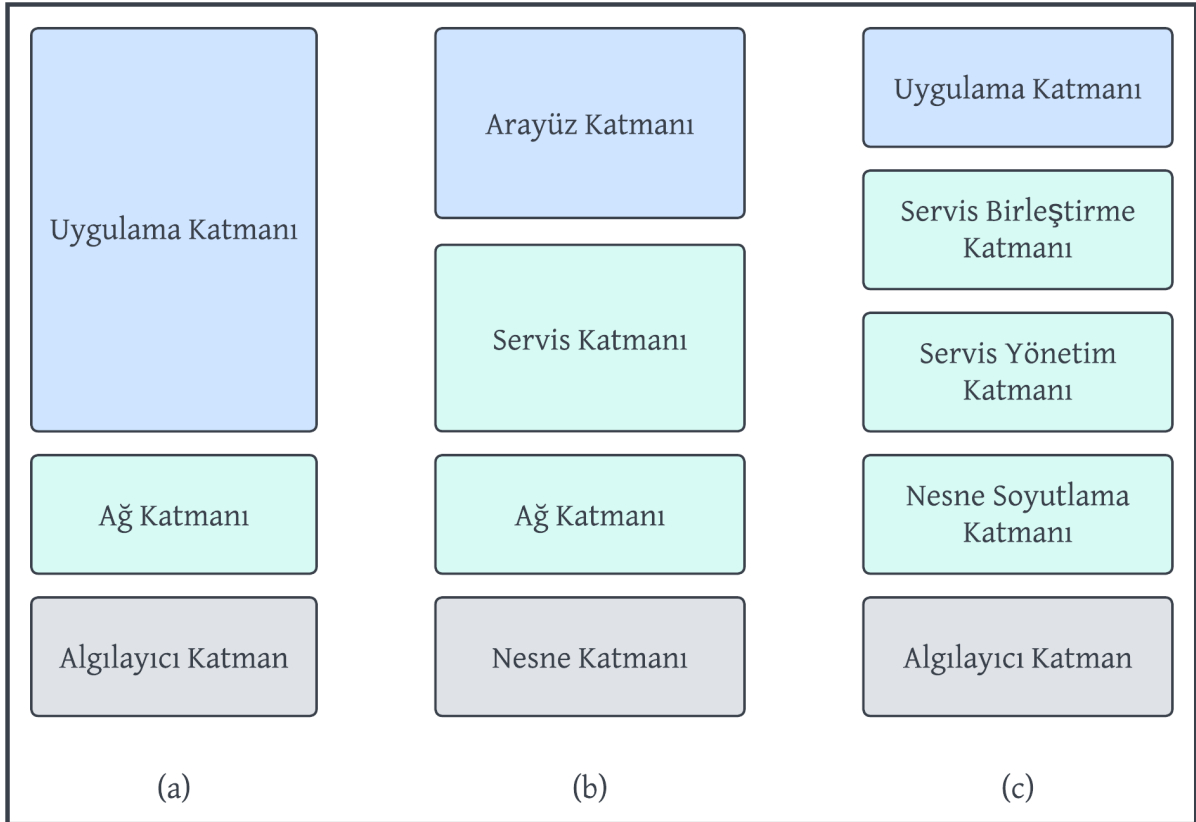
Endüstriyel uygulamalarda tekil bir servis katmanı iş görmektedir. Bunun sebebi endüstriyel ortamlar kurulan IoT ağlarda donanımsal çeşitliliğin ve uygulama çeşitliliğinin düşük olmasıdır. Bu durumun aksine genel amaçlı IoT ağlar çok daha heterojen bir yapıdadır. Bu heterojen karakteristiğın desteklenmesi, yönetimi ve kontrolü daha kolay yapılabilen, ticari fayda yaratabilecek ve esnek bir mimari elde etmek amacıyla servisler ile ilgili katmanının bölünmesine gidilmiştir. Bu aşamada 5 katmanlı servis odaklı mimarilere geçildiği gözlemlenmiştir [4]. 5 katmanlı ağ mimarisini oluşturan katmanlar ve katmanların işlevleri aşağıda sıralanmıştır.

- Nesne Katmanı (Object Layer) nesnelerin ve birleştiricilerin (actuator) bulunduğu katmandır. Veri algılanır, birleştirilir ve basit seviyelerde veri ön işleme tabi tutulur. Akabinde üst katmana gönderilir.
- Nesne Soyutlama Katmanı (Object Abstraction Layer) nesnelerde bulunan servisleri üst katmanın kullanabileceği şekilde soyutlayan katmandır. Soyutlama işleminin amacı oldukça farklı donanımsal özellikte olan IoT servislerini servis yönetimi ve birleştirmede problem çıkmaması adına tekilleştirmektir. Donanım, bağlantı teknolojisi ve protokolü farklılıklarını giderir.
- Servis Yönetim Katmanı (Service Management Layer) servis talepleri ile servisleri eşleştiren, programcıların donanımdan bağımsız kod geliştirmesini sağlayan katmandır. Aynı zamanda ağ yönetimini de üstlenir.

- Servis Birleştirme Katmanı (Service Composition Layer) IoT servislerinin birleştirilerek karmaşık uygulamalar haline getirilmesine izin veren katmandır.
- Uygulama Katmanı (Application Layer) görselleştirme yapılan ve ağ servislerinin/uygulamalarının son kullanıcıya sunulduğu katmandır. Ağdan toplanan veri üzerinde yapılan çıkarımla, bilgi keşfi uygulamaları, gözlemleme ve tasarım uygulamaları bu katmanda yapılır.

Yukarıda verilen 5 katmanlı mimari servis odaklı IoT mimarilerini en iyi özetleyen ve IoT alanında en kabul görmüş mimaridir. Bu mimari temelinde, IoT ağın kullanım alanına göre farklılaşmalar da görülmektedir. Ticari odaklı IoT ağlarda servis birleştirme katmanı uygulama katmanı olarak tanımlanıp, beşinci katmanı ticari değer çıkarımına adanabilmektedir.

Bahsedilen IoT ağ mimarileri Şekil.2.1’de verilmiştir.



Şekil 2.1. IoT Ağ Mimarileri, (a) 3 Katmanlı Konvansiyonel Mimari, (b) 4 Katmanlı Endüstriyel Mimari, (c) 5 Katmanlı Servis Odaklı Modern Mimari.

2.1.3. IoT karakteristik özellikleri

IoT kavramının öngördüğü ağların karakteristik özellikleri bulunmaktadır. Bir ağın IoT ağ olarak tanımlanabilmesi için bu özellikleri sağlaması gereklidir.

Heterojen yapı

IoT ağların en karakteristik özelliklerinden biri çeşitliliktir. Bir IoT ağı oluşturan donanımsal ve yazılımsal olarak çeşitlilik gösteren bileşenlerinin koordineli bir şekilde çalışabilmesi gereklidir. IoT ağlar farklı cihazları, iletişim teknolojilerini ve protokollerini desteklemelidir [28].

Ölçeklenebilirlik

İnsan hayatını her alanına nüfuz eden IoT ağlarda nesne sayıları astronomik sayılara ulaşabilmektedir [29]. Bu sebeple IoT ağların ölçeklenebilir yapıda olması gereklidir [28].

Yaygın kapsama alanı

Nesnelerin verimli bir şekilde çalışabilmesi ve ağ işlevlerinin kesintiye uğramaması adına yaygınlığı ve bağlantı kalitesi yüksek bir ağ yapısına ihtiyaç vardır [28].

Enerji etkinliği

Nesneler enerji açısından kısıtlı olduğu için iletişim ve hesaplama işlemlerinin enerji harcamaları optimize edilmelidir [28]. Aynı zamanda, GIoT kavramının öngördüğü üzere IoT ağların karbon ayak izinin azaltılması sürdürülebilirlik açısından ve dünyanın geleceği açısından kritiktir [30-33].

Konum tespit ve takip yetenekleri

Konum farkındalığı ve tanımlanabilirlik IoT nesneleri için önemlidir [28]. Konum tabanlı uygulamaların başarısı ve konumsal akıllı kararların alınması bu yeteneklere bağlıdır. Aynı zamanda, nesnelerin kendi konumlarındaki değişikliği takip edebilmesi de gereklidir [4].

Öz örgütlenme yeteneği

IoT ağlar heterojen ve dinamik yapıdadır. Bu sebeple nesne veya ağ katmanlarında olabilecek değişikliklere ve hatalara otonom şekilde tepki verebilmesi, IoT ağ tarafından sağlanan işlevlerde problem çıkmaması adına önemlidir. İnsan müdahalesi gerekliliği ağ mimarisi aşamasında minimize edilmelidir. Nesneler kendi aralarında organize olabilmeli, konum ve bağlam farkında bir şekilde servis yönetimi yapabilmelidir [28].

Veri yönetimi

IoT ağlarda üretilen veri miktarı oldukça yüksektir ve bu veri üzerinde işlem yapılması gereklidir. Yapılan işlem veri transferi veya analizi olabilir. Büyük miktarda verinin transfer işlemlerini sağlayabilecek ağ kapasitesi sağlanmalıdır. Gerekli yerlerde veri birleştiriciler ile yapı desteklenmelidir. Ağın veri işleme kapasitesinin de yeterli olması gereklidir. IoT uygulamalarının büyük çoğunluğu toplanan verinin işlenmesi ve anlamlandırılmasını gerektirir [4, 28].

Güvenlik

Nüfuz edici yapısıyla IoT nesneleri insan hayatının her aşamasına girer. Bu sebeple IoT ağların güvenlik ve gizlilik mekanizmalarının olması oldukça önemlidir [28].

2.1.4. IoT iletişim protokolleri

Uygulama katmanındaki iletişim protokolleri, uygulamalar / servisler ile nesneler arasındaki iletişimi düzenler. Literatürde bu amaçla kullanılan çok sayıda protokol tanımlanmıştır. IoT ağların karakteristik özelliklerinden dolayı kullanılacak iletişim protokollerinin taşıması gereken temel özellikler vardır. Bunlara örnek vermek gerekirse ölçeklenebilirlik, düşük paket kaybı, kaynakların verimli kullanılması ve enerji etkinliği verilebilir [34]. Belirtilen ortak gereksinimlere ek olarak iletişim protokolünün seçiminde uygulamanın spesifik gereksinimleri, kullanılan nesnelerin kaynak kısıtları ve IoT ağın altyapısı gibi faktörler göz önüne alınmaktadır [16].

Takdir edileceği üzere, IoT uygulamalarının çeşitliliği göz önüne alındığında tekil bir iletişim protokolünün kullanılması yeterli gelmemektedir. Çok sayıda protokol IoT ağlarda

kullanılmaktadır. Bu bölümde en temel iletişim protokolleri konu ile ilgili temel derleme makaleleri [4, 16, 34, 35] kullanılarak verilmiştir.

HTTP: HyperText Transfer Protocol

HTTP günlük kullanılan internetin altyapısını oluşturan protokoldür. Yaygınca desteklenen bir protokol olduğundan dolayı IoT uygulamalarında yaygınca kullanılmaktadır [36]. Ancak, IoT kavramının şafağından önce tasarlanan bir protokol olduğundan dolayı HTTP'nin bazı özellikleri IoT'de uygulamalarının gereksinimlerini karşılamada yetersiz kalmaktadır. HTTP metin tarzında veri gönderen bir protokoldür ve IoT nesnelerinin kısıtlı kaynakları göz önünde geliştirilmemiştir. Bu sebeple HTTP kullanan IoT nesnelerinin verileri metin formatına çevirmek zorunda kalmakta, hali hazırda kısıtlı olan kaynaklarını gereksiz yere harcamaktadır [37]. Bu sebepten ötürü kullanımı yaygın olsa da, IoT'nin karakteristiklerine daha uygun protokollere ihtiyaç vardır [16].

MQTT: Message Queuing Telemetry Transport

MQTT, M2M iletişim için geliştirilmiş düşük kaynaklar üzerinde çalışabilen bir protokoldür. Yüksek ağ gecikmeleri ve dar bant genişliğine sahip ağlarda sorunsuz çalışabilmektedir [35]. Aracı (broker) üzerinden çalışan yayımla / abone ol (publish / subscribe) tekniğini kullanılır [16, 34]. Taşıdığı veri türünde herhangi bir kısıt olmayan MQTT, TCP-tabanlı asenkron bir protokol olduğu için IoT uygulamaları için oldukça uygundur [16].

DDS: Data Distribution Services

DDS görev kritik ağlar için geliştirilmiş, aracısız yayımla / abone ol mimarisine yüksek performanslı bir protokoldür [4, 16]. Veri odaklı bir protokol olup çoklu gönderim (multicast) desteklemektedir [16, 35]. Bu sebeple IoT ağlarda M2M iletişim için kullanılmaktadır. Aracısız mimarisi sebebiyle kaynak kısıtlarına sahip IoT nesneleri için uygundur, yüksek ölçeklenebilirliğe sahiptir [35].

AMQP: Advanced Message Queuing Protocol

AMQP mesaj odaklı ağlar için geliştirilmiş bir iletişim protokolüdür [4]. Aracılı yayımla / abone ol mimarisi ile birlikte konvansiyonel talep / cevap (request / response) mimarisini de

kullanabilir [16]. TCP benzeri güvenilir gönderim protokollerini kullandığı için sağladığı iletişim oldukça güvenilirdir, mesaj iletim garantileri mevcuttur. IoT ağlar için oldukça uygun, kaynak tüketimi az ve ağdaki çeşitliliği destekleyen bir protokoldür [35].

XMPP: Extensible Messaging and Presence Protocol

XMPP, XML veri akış protokolüne dayanan bir anlık mesajlaşma protokolüdür [35]. Yayınla / abone ol ve talep / cevap mimarilerini destekler [16]. Yakın gerçek zamanlı uygulamalarda kullanılmaktadır ancak mesaj iletim garantisi sunmaz. Aynı zamanda metin tabanlı bir protokol olduğu için hem kaynak kısıtlı nesnelere ek işlem gücü yüklemektedir [16], hem de ağ kullanımı yüksektir [4]. Bu sebeplerle, hali hazırda IoT’de kullanılmasına rağmen IoT ağların karakteristiklerine daha uygun varyasyonlarının geliştirilmesi gereklidir [4, 16].

COAP: Constraint Application Protocol

UDP-tabanlı bir protokol olan COAP düşük kaynağa sahip cihazlar ve bu cihazlardan oluşan ağlar için geliştirilmiş bir protokoldür. İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force / IETF) isimli bağımsız bir grup tarafından bir standart haline getirilmiştir. Hem aracı tabanlı yayınlama / abone olma yapısını hem de konvansiyonel talep / cevap mimarisini destekler [16].

COAP, DDS, HTTP, XMPP ve AMQP protokollerinin özeti Çizelge 2.1’de verilmiştir.

Çizelge 2.1. IoT’de uygulama katmanında kullanılan iletişim protokolleri [16]

Protokol	Yıl	İletişim Mimarisi	Paket Taşıma Protokolü	Veri Formatı
HTTP	1997	Talep / Cevap	TCP	Metin
MQTT	1999	Aracılı Yayınla / Abone Ol	TCP / UDP	İkili Veri
DDS	2001	Aracısız Yayınla / Abone Ol	TCP / UDP	İkili Veri
AMQP	2003	Talep / Cevap, Aracılı Yayınla / Abone Ol	TCP / SCTP	İkili Veri
XMPP	2004	Talep / Cevap, Aracısız Yayınla / Abone Ol	TCP / UDP	Metin
COAP	2010	Talep / Cevap, Aracılı Yayınla / Abone Ol	UDP / SCTP	İkili Veri

2.1.5. IoT uygulama alanları

IoT kavramı milyarlarca cihazı birbirine bağlamakta ve gün geçtikçe insan hayatına daha yüksek seviyelerde nüfuz etmektedir. Bu derece bir nüfuz, insan hayatını yeniden tasarlamaktadır [38]. Gün geçtikçe IoT nesnelerinin sayısı artmakta ve insan hayatı daha fazla sayıda IoT uygulamasına dayanan daha teknolojik destekli forma evrilmektedir [39]. IoT uygulamaları insan hayatındaki çeşitli problemlere çözümler bulmaktadır. Dolayısıyla, problemler çeşitlendikçe IoT uygulamaları da çeşitlenmektedir. Bu bölümde IoT uygulamaları kullanıldıkları alana göre kategorize edilecek ve sunulacaktır.

Akıllı şehirler

1990'lar itibarı ile kullanılmaya başlayan “akıllı şehir” kavramı, bilişim teknolojileri ile şehir sakinlerinin hayat kalitesinin artırılması, hizmetlerin daha etkin bir şekilde verilmesi ve bunlar yapılırken ekonomik, çevresel ve sosyal anlamda sürdürülebilirlik sağlanmasını öngörür [40]. İnsan nüfusunun yüzde 70'inin şehirlerde yaşadığı göz önüne alındığında akıllı şehir kavramının önemi daha çok ortaya çıkmaktadır [41]. Bilişim teknolojileri ile desteklenen şehirler artık bir tercihten ziyade bir gereklilik haline dönüşmüştür [41]. Talep edilen akıllı şehir uygulamalarının hayata geçirilmesi için IoT kosepti oldukça uygundur. Servis çeşitliliği ve geniş erişimi ile IoT nesneleri çoğu akıllı şehir uygulamasının geliştirilmesinde kullanılmaktadır [42].

Akıllı şehirlerde yüksek çeşitlilikte uygulamalar kullanılmaktadır. Akıllı trafik uygulamaları taşıt park yerlerinin yönetilmesinde [43], insan müdahalesi getirmeyen trafik ışıkları ve ışıklandırmaları için yaygınca kullanılmaktadır. Akıllı trafik uygulamaları Amsterdam, Barcelona, Dubai ve Singapore gibi çok sayıda şehirde kullanılmaktadır [44].

Şehir yönetiminin sağlaması gereken sağlaması beklenen hizmetlerde IoT uygulamalarının kullanılması sağlanan servislerin kalitesini arttırmaktadır. Ulaşım sistemlerinde [6], atık yönetimi hizmetlerinde [41, 45] ve enerji sistemlerinde [46] IoT destekli uygulamalar kullanılmaktadır [6, 47, 48].

Akıllı evler ve binalar IoT uygulamalarının çokça kullanıldığı alanlardır. IoT uygulamaları sayesinde enerji etkinliği başta olmak üzere [49] çok sayıda ek işlev otomatikleştirilebilmektedir. Akıllı evlerde bitki sulama, ışıklandırma [49], iklimlendirme

işlemleri otomatikleştirebilmekte, hırsızlık ve yangın gibi istenmeyen olaylara karşı tespit ve koruma sistemler işlevleri IoT uygulamaları tarafından sağlanmaktadır [5]. Toplumda hayatını tek başına idame ettirme yeteneğine sahip olmayan bireyleri için IoT uygulamaları hayat kalitesini arttırmaktadır. Engelli [50] ve yaşlı [51] bireylerin hayatını kolaylaştırmak adına geliştirilen akıllı ev uygulamaları literatürde bulunmaktadır.

Akıllı endüstri

Endüstri 4.0, akıllı teknolojilerin, yapay zekanın ve ağ teknolojilerinin üretim süreçlerinde etkin bir şekilde kullanılmasını ve insan faktörünün üretim süreçlerinde minimize edilmesini öngörür [13]. Sağladığı RFID etiketler, algılayıcılar ve kablosuz bağlantı özelliğine sahip nesneler sayesinde IoT kavramı endüstriyel IoT (Industrial IoT / IIoT) uygulamalarının tasarlanmasında ve hayata geçirilmesi problemine çözümler sunmaktadır [14]. Üretim makinaları, akıllı cihazlar, ekipmanlar gibi çok sayıda cihazı birbirine bağlayan IIoT, sistemin algılama işlevini geliştirir, üretim süreçleri üzerinde daha iyi kontrol sağlar, makine bakımlarını otomatikleştirir, olası makine hatalarını öngörür ve veri analizi yeteneği sayesinde üretim süreçlerinin verimini artırır [52, 53].

Akıllı sağlık uygulamaları

IoT temelli sağlık uygulamalarının (Health-IoT / HIoT) hastaları takip etme ve gerçek zamanlı analiz etme yeteneği vardır [54]. Belirli bir göreve dedike, giyilebilir algılayıcılar ile tıbbi ölçümler daha yüksek kesinlikle ve daha hızlı yapılabilmektedir [54, 55]. Bu da ilgili sağlık personelinin olası sağlık sorunlarına daha hızlı müdahale edebilmesini sağlamaktadır.

Algılayıcılar aracılığı ile toplanan yüksek miktardaki biyometrik veri, konvansiyonel tıbbi çeşitli açılardan geliştirmektedir. Toplanan yüksek miktardaki veri olası sağlık sorunlarını önceden öngörebilen, yüksek doğruluğa sahip yapay zeka temelli modeller geliştirilebilmesini mümkün kılmaktadır [56]. HIoT'un daha uygulanabilir kıldığı bir diğer yenilik de kişiselleştirilmiş sağlık uygulamalarıdır [55, 57]. Her hasta özelinde karar almayı gerektiren kişiselleştirilmiş sağlık uygulamalarında, çok sayıda ölçümün eksiksiz olarak yapılması doğru kararlar alınması açısından önemlidir [58]. Giyilebilir algılayıcılar bu ihtiyaç duyulan devamlı ve düşük hata oranına sahip veriyi sağlayabilmektedir [55, 59].

Hastane personellerinin, ekipmanlarının ve araç/gereç/ilaç stoklarının HIoT temelli uygulamaları ile takip edilmesi sağlık kurumlarında verimliliği arttırmaktadır. HIoT uygulamaları bahsedilen avantajlara ek olarak sağlık hizmetlerinin maliyetini düşürmekte ve hastaların aldığı hizmetlerin kalitesini arttırmaktadır [56, 60]. Bu potansiyelinden dolayı HIoT uygulamalarının tüm IoT uygulamaları arasında en yüksek ekonomik etkiye sahip olacağı düşünülmektedir [55, 57, 58].

Akıllı lojistik

Lojistik sektörü modern hayatın görünürlüğü az olsa da en önemli sektörlerinden biridir [61, 62]. Bu sebeple, lojistik süreçlerinin optimize edilmesi adına akıllı lojistik fikri öne sürülmüştür. Modern bilişim teknolojilerinin lojistik alanında kullanılmasını öngören “akıllı lojistik” IoT uygulamalarının kullanıldığı alanlardan biridir [63, 64]. Paketleme, taşıma, depo yönetimi, yükleme, ürün ve filo takibi gibi lojistik operasyonların alt operasyonlarında IoT uygulamaları sıkça kullanılmaktadır [44, 62, 63].

Akıllı tarım

IoT uygulamalarının modernize ettiği bir diğer alan da tarımdır [12]. Tarım alanında çok sayıda ve çeşitte operasyon IoT ağları ve uygulamaları ile yapılmaktadır. Bunlardan kısaca bahsetmek gerekirse hayvan, bitki ve ortamların gözlemlenmesi en başta gelen alandır. Tarım topraklarında su kalitesi, nem ve kimyasal takibi, ahır ve seralarda hava, sıcaklık ve gaz takibi IoT uygulamaları sayesinde otomatikleştirilmektedir [11]. Hastalık ve zararlıların tespiti, öngörülmesi ve önlenmesi, sulama, gübreleme ve hasat sistemlerinin otomatikleştirilmesi vb. uygulamalar IoT uygulamaları ile mümkündür ve uygulanmaktadır [11, 12].

IoT uygulamalarının mümkün kıldığı bir diğer alan ise hassas tarımdır. Tarımın geleceği olarak tanımlanan bu alan, her bitki ve hayvana en yüksek verimi vermesi için ihtiyacına özel ilgi gösterilmesini öngörür [9, 10].

Afet yönetim uygulamaları

Kriz ve afetler ister doğal olsun (deprem, sel, orman yangını, tsunami) ister insan kaynaklı (yangın, patlama, terörist saldırı), insan hayatı kayıplarına ve ekonomik zararlara sebep

olmaktadır [65]. Bu gibi durumlarda IoT uygulamaları insan hayatının kurtarılması ve olası maddi kayıpları azaltmak amaçlı olarak kullanılmaktadır. Gerek öngörülebilir afetlerin erken tespiti (sel, orman yangını, volkanik boşalım) [66], gerekse olası kriz durumlarında etkilenen insanların etkin bir şekilde tahliye edilmesi ve yaralılara ulaşılması adına IoT uygulamaları kullanılmaktadır [67].

Çevre gözlem uygulamaları

İklim değişikliği sebebiyle çevre unsurlarının (deniz, nehir, göl, orman vb.) takibi oldukça önem kazanmıştır. Geniş alanlara dağılma özelliği sayesinde, büyük alanların algılayıcılar ile gözlenmesi IoT uygulamaları için oldukça kolaydır. Bu sebeple, orman yangını tespiti [68, 69], su kaynaklarının ve denizlerin takibi [70-73], hava kalitesi ve kirliliği tespiti ve öngörüsü [15], tehlike altındaki hayvanların yasadışı avcılığa karşı korunması [69] gibi kritik alanlarda IoT ağlar kullanılmaktadır.

2.1.6. Destekleyici teknolojiler

IoT ağlar, milyonlarca nesnenin iletişim halinde bulunduğu oldukça büyük bir ağlardır. Ağların boyutuna ek olarak, nesnelerin çeşitliliği ve ürettikleri astronomik miktardaki veri göz önüne alınırsa, IoT'nin kavramsal olarak mümkün kıldığı fonksiyonların gerçekleştirilmesi için ek teknolojilerin gerekliliği açıktır. Bu bölümde IoT ağlarda sıkça kullanılan destekleyici teknolojiler gruplanarak açıklanacaktır.

Destekleyici ağ teknolojileri

Bahsedilecek ilk grup teknoloji ağ yapısına ve yönetimine dairdir. Bahsedildiği üzere IoT ağlar milyonlarca nesneyi birbirine bağlayan çok büyük yapılar olabilmektedir. Giyilebilir cihazlar, telefonlar, ev aletleri, algılayıcılar ve internet erişimi olan her cihazın bir nesne olarak iş gördüğü bir gelecekte nesne sayılarının on milyarları aşip yüz milyarlaraya ulaşması beklenmektedir. Bu büyüklükteki bir ağın yönetimi ek teknolojilere ihtiyaç duymaktadır.

Bahsedilecek ilk teknoloji bilgisayar ağlarının yazılımsal kontrolünü sağlayan SDN teknolojisi. Modern internet çok sayıda ve çeşitte ağ cihazından oluşmaktadır [74]. Bu cihazlar belirli uygulamalara ve protokollere özelleşmiş şekildedir ve donanımsal olarak yüklenmiş gömülü kurallar ile yönetilirler [75]. Bahsedilen bu yapı donanım temellidir ve

internetin ilk tasarım gereksinimlerini yerine getirmek konusunda başarılıdır. Bu gereksinimler yüksek kapasite ve düşük gecikmedir. İçlerinde IoT kavramının de sayılabileceği yeni kavramların oluşması ile birlikte bahsedilen konvansiyonel donanım temelli mimarinin eksikleri ortaya çıkmaya başlamıştır. Bunlar yönetilebilirlik, esneklik ve genişletilebilirlik olarak özetlenebilir. Bu eksiklerin kapatılması adına SDN öne sürülmüştür. SDN uyarınca ağın kontrol ve veri iletim katmanları ayrılmıştır [74]. Programlanabilir olan ağ cihazları merkezi bir kontrolcü tarafından kontrol edilir ve bu sayede ağ üzerinde daha yüksek seviyede yönetilebilirlik ve esneklik sağlanır [76]. IoT ağların en karakteristik özelliklerinden olan büyük boyut ve cihaz/protokol çeşitliliği SDN ile çok daha kolay bir şekilde yönetilebilir [76]. IoT ağların bir diğer özelliği ise dinamizmdir. Ağa devamlı olarak yeni nesneler bağlanabilir veya ağdan ayrılabilir. Bu dinamizmin konvansiyonel ağ yapısı ile desteklenmesi mümkün değildir, bu sorunun çözümü için SDN uygun bir çözümdür [76]. IoT ağlarda SDN kullanımına dair çok sayıda çalışma mevcuttur. Daha detaylı bilgi için SDN ile ilgili derleme makalelerine bakılabilir [74-78].

FC, IoT ağların ölçeklenebilirliğine yönelik bir diğer kavramdır. Bulut hesaplama (Cloud Computing / CC) IoT uygulamalarının ilk dönemlerinden itibaren IoT'nin paralel bir şekilde ilerlediği teknolojilerden biridir. IoT uygulamalarında veri işlenmesi, veri saklanması ve kullanıcı arayüzlerinin sağlanması gibi birçok işlev bulut sunucularda yapılmaktadır. Ancak IoT ağların boyutları, ağa bağlı nesne sayısı arttıkça merkezi konumda bulunan bulut sunucularının kullanımı ölçeklenebilirlik problemleri yaratmaya başlamıştır. Ayrıca toplam ağ kullanımının da ağlar büyüdükçe arttığı gözlemlenmiştir. Bu problemlerin çözümü için Cisco tarafından FC kavramı önerilmiştir. FC uyarınca bulut katmanı ile nesne katmanı arasına bir cihaz katmanı daha eklenir. Sis katmanı denen bu katman, nesne ile bulut katmanlarını bağlar, hesaplama yeteneğine sahip sis sunucu adı verilen cihazlardan oluşur ve coğrafi olarak dağıtık bir şekilde bulunur. Nesneler direkt bulut sunuculara bağlanmaz ve kendilerinden sorumlu olan sis sunucularına bağlanır. Bulut sunucularda bulunan görevler olabildiğince sis sunucularına taşınır. Bu sayede IoT ağlarda daha düşük gecikme, daha düşük ağ kullanımı, daha iyi gizlilik ve güvenlik, konum farkındalığı, coğrafi dağıtıklık, çeşitlilik ve ortak çalışabilirlik sağlanır [79, 80]. Daha detaylı bilgi için FC ile ilgili derleme makalelerine bakılabilir [29, 81, 82].

Sis hesaplama benzer bir şekilde, veri işlerini ağın nesne katmanına yaklaştıran uç hesaplama (Edge Computing / EC) IoT ağların ölçeklenebilirliğini arttıran bir başka

teknolojidir. EC uyarınca, nesnelerin ürettiği veri üretim konumuna yakın bir konumda işlenir. Bu sayede, FC ile benzer bir şekilde daha verimli ağ kullanımı ve yüksek veri gizliliği sağlanır. Daha detaylı bilgi için EC ile ilgili derleme makalelerine bakılabilir [77, 78, 82-84].

Destekleyici veri teknolojileri

Bahsedildiği üzere IoT ağlar astronomik miktarlarda veri üretir. Destekleyici ağ teknolojileri üretilen verinin iletimi konusunda fayda sağlarken, destekleyici veri teknolojileri üretilen veriden bilgi ve değerli içerik çıkarımına odaklanır.

IoT nesnelerinden elde edilen veri uygulamaların çalışması için kritik öneme sahiptir. Bu sebeple verilerin etkin bir şekilde işlenmesi gereklidir. Verinin büyüklüğü göz önüne alınca bu boyutta verinin FC/EC'nin öngördüğü şekilde işlenebilmesi için spesifik çözümlere ihtiyaç vardır. Bu konuda geliştirilen teknoloji büyük veri teknolojileridir. IoT özelinde geliştirilen bir teknoloji olmasa da büyük veri, IoT veri karakteristiğine oldukça uygundur. Büyük veri tanımına göre bir verinin büyük veri olarak tanımlanabilmesi için yüksek hacimli, hızlı üretilen, çeşitli, dinamik, değerli ve değişken olması gereklidir [85, 86]. Sayılan ölçütler IoT veri karakteristiği ile örtüşmektedir ve bu sebeple IoT verisi sıkça büyük veri teknolojileri ile işlenmektedir [87].

Makina öğrenmesi ve yapay zeka teknolojileri IoT ağların akıllılaştırılması için kullanılan teknolojilerdir. Gerek ağın operasyonel yeteneklerinin geliştirilmesi olsun, gerek son kullanıcıya yeni işlevler sağlamak olsun, yapay zeka IoT'de önemli görevler yerine getirir [86, 88, 89]. Öngörü ve tespit alanındaki çoğu IoT uygulaması alt yapılarında makine öğrenmesi ve daha spesifik olmak gerekirse derin öğrenme temelli modellere sahiptir [86].

Blockchain Teknolojisi

Blockchain teknolojisi elektronik kayıtların değiştirilmesini önlemek adına geliştirilmiş bir teknolojidir [90]. Dağıtık defter teknolojisine sahip olan blockchain yapısında veriler blok olarak tutulur ve yeni eklenen her blok kendisinden önce gelen bloğun verisinin özetlenmiş halini tutar [91]. Bu sayede, geçmişteki kayıtlarda olan değişiklik her daim fark edilebilir. Aynı zamanda, kayıtlar çok sayıda kullanıcıda tutulduğu için veri üzerinde değişiklik

yapmak oldukça zordur [91-93]. Blockchain teknolojisi ilk olarak kripto-paralar alanında kullanılmaya başlamıştır. Ardından kullanım alanlarına akıllı kontratlar eklenmiştir. Günümüzde ise artık uygulamalar da blockchain mimarisini kullanmaktadır [93].

Blockchain teknolojisi, IoT ağları birçok alanda geliştirme potansiyeline sahiptir. Bunlardan ilki IoT verilerinin değiştirilemez bir şekilde saklanmasıdır. Bahsedildiği üzere Blockchain teknolojisinin temelinde veri güvenliği vardır ve IoT verisinin güvenliğini sağlamak konusunda oldukça uygulanabilir [90, 94]. Blockchain'in dağıtık yapısı kritik IoT verilerinin merkezi bir sunucuya ihtiyaç duyulmadan dağıtılabilmesini mümkün kılmaktadır [94]. Akıllı kontratlar sayesinde nesneler kendi aralarında otomatik ödemeler yaparak daha düşük maddi külfete neden olacak ve daha hızlı bir şekilde operasyonlarını yapabilir [94]. Bahsedilenlere ek olarak IoT ağlardaki beraber çalışabilirlik yeteneğini de Blockchain teknolojisi ile geliştirilebilir [90].

2.1.7. IoT geleceği

IoT ağlar ve uygulamaları hayatın her alanına nüfuz ederek günlük hayatımızı baştan aşağıya değiştirmektedir. Ancak, bu derece yaygınlaşan bir teknolojinin sürdürülebilirliği ve çevre duyarlılığı gezegenimizin geleceği için oldukça önemlidir. Bilindiği üzere insanoğlunun bir sonraki büyük sınavı iklim değişikliğine karşıdır. İklim değişikliğine engel olmak için yapılabileceklerden en basit ve etkililerden biri enerji tüketiminin optimize edilmesidir. Daha düşük enerji tüketimi daha az karbon ayak izi demektir. Bu fikirler ortaya atılan GIoT kavramı IoT ağlarda enerji etkinliğinin gerekliliğine dairdir [30-32]. Günümüzde IoT konusunda yapılan akademik ve sektörel çalışmaların büyük bir kısmında yapılan uygulamanın enerji etkinliği irdelenmektedir [31]. Hatta, son zamanlarda enerji etkinliğinin bir QoS isteri olarak tanımlandığı çalışmalar mevcuttur. Bu bize GIoT kavramının önemini göstermektedir. Geleceğin IoT ağları ve uygulamaları tasarlanırken işin sürdürülebilirlik ve enerji etkinliği kısmı atlanmamalıdır [17, 32].

Beşinci ve altıncı nesil kablosuz iletişim ağlarının, kısaltmaları ile 5G ve 6G, IoT ağlarda devrimsel değişikliğe sebep olması beklenmektedir. 5G ile birlikte gelen daha yüksek bant genişliği, daha düşük gecikme, daha düşük maliyet ve daha yüksek enerji avantajları IoT ağların daha geniş coğrafi alanlarda daha kaliteli hizmetler vermesini sağlayacaktır [22, 95]. 6G'nin ise bu avantajları yapay zeka destekli ve IoT odaklı bir şekilde geliştireceği, 6G

sayesinde dünyadaki ve hatta dünya dışı cihazların (uydu ve uzay cihazları) birbirine daha iyi bir şekilde bağlanabilecek olması beklenmektedir [23, 24]. Yeni kablosuz bağlantı teknolojileri sayesinde IoT ağların yeteneklerinin artacağı ve yepyeni ufuklar açacağı şüphesizdir.

2.2. Nesnelerin İnternetinde Servis Kalitesi

IoT’de ağ kullanıcısının talep ettiği bir servisin sunulmasında en önemli kriter kullanıcı tarafından belirtilen QoS isterleridir. QoS isterleri, servisi talep eden kullanıcının belirttiği, servis sağlanırken garanti altına alınması gereken özellikleri ve kalite göstergelerini ifade eder. QoS isterleri karşılanmadan sunulan bir servis sunulmuş kabul edilmez. Bu sebepten ötürü, IoT ağlarda QoS desteğinin ağın performans ölçütü olduğunu ifade edersek yanlış olmaz [16].

QoS, IoT ile ortaya çıkan bir kavram değildir ve internet servislerinin sunulmasında uzun yıllardır kullanılmaktadır. Ancak internet servislerinde QoS yönetimi IoT’ye kıyasla kolaydır, IoT ağların karakteristik özelliklerinden dolayı IoT ağda QoS sağlanması zor bir uğraştır. Yüksek nesne ve ağ elemanı sayıları, nesnelerin donanımsal çeşitliliği, kullanılan iletişim protokollerinin çeşitliliği, bölgesel olarak kullanılan destekleyici teknolojiler ve IoT uygulamalarının çeşitliliği gibi IoT ağı geleceğin teknolojisi yapan birçok unsur QoS sağlanması aşamasında zorluk olarak karşımıza çıkmaktadır. Ağ büyüklüğü QoS yönetimini zorlaştırırken, ağın heterojen yapısı ortak kullanılabilecek bir QoS modellemesinin geliştirilmesine engel olmaktadır. Bu sebeplerden ötürü IoT’de QoS sağlanması aktif bir araştırma konusu olmuştur ve literatürde sıkça ele alınmaktadır.

Bu bölümde, IoT’de QoS sağlanması konusunda yapılan literatür taraması sunulacaktır.

2.2.1. İletişim protokolleri

IoT’de kullanılan iletişim protokolleri QoS’i farklı seviyelerde desteklemektedir. Bu bölümde en yaygın IoT protokollerinin QoS destekleri incelenecektir.

En düşük seviyede QoS desteği sağlayan protokoller HTTP ve XMPP protokolleridir. İki protokolde de özel olarak tanımlanmış herhangi bir QoS mekanizması bulunmaz. Protokollerin alt yapısında kullanılan iletim katmanı protokolünün veri iletim garantisi

sağlanan tek QoS desteğidir. TCP kullanan HTTP protokolünde de kısıtlı olsa dahi veri iletim garantisi olduğu gözlemlenebilir. XMPP protokolünde ise hem TCP hem UDP kullanılabilir. TCP kullanan XMPP protokolünde HTTP ile benzer şekilde kısıtlı bir QoS desteği vardır, ancak UDP kullanan XMPP protokolünde herhangi bir QoS desteği bulunmamaktadır.

MQTT protokolü veri transfer garantisini üç seviyede sunmaktadır.

- QoS 0 seviyesi her mesajın bir en fazla bir kere gönderilmesini sağlar.
- QoS 1 seviyesi her mesajın en azından bir kere gönderildiğinin garanti edildiği seviyedir. Ancak, çoklu gönderim olabilir.
- QoS 2 seviyesi her mesajın hedefine ulaştığını ve çoklu gönderim olmadığını garanti eder.

MQTT QoS seviyelerinden QoS 0 veri kaybının önemli olmadığı uygulamalar için uygundur. QoS 1 veri kaybı konusunda hassas olan uygulamalara yöneliktir, ancak çoklu gönderime sebep olabilir. QoS 2 seviyesi hem verinin hedefe ulaşmasını garanti eder hem de çoklu veri gönderilmesini engeller. Bu sebeple en üst düzey QoS seviyesini sağlar ancak her mesaj gönderimi için dört mesaj gönderimi yapar. Bu açıdan ağı verimsiz kullanır ve yavaştır.

CoAP ve AMQP protokolleri isimlendirmede farklı olsa da oldukça benzer iki seviyeli bir mesaj gönderim garantisi sunar, erişim garantili ve erişim garantisi olmayan seviyeler. AMQP protokolü CoAP'a ek olarak farklı mimarileri desteklediği için verimlilik ile alakalı QoS metriklerine kısıtlı olsa dahi destek verebilmektedir [16].

IoT iletişim protokolleri arasında en yüksek QoS seviyesini DDS sağlamaktadır. 20'nin üzerinde QoS istisini desteklemektedir. Bunlar arasında güvenilirlik, zaman kısıtları, ağ verimliliği, önceliklendirme ve kaynak verimliliği gibi metrikler bulunmaktadır [16].

2.2.2. Farklı QoS metrikleri

Bir önceki bölümde iletişim protokolleri QoS açısından değerlendirilmiştir. İletişim protokollerindeki QoS metrikleri basit seviyededir ve ağ üzerinden mesajların transferi ile ilgilidir. Ancak, IoT'de ağ katmanında çok daha üst seviyeli QoS metrikleri de

bulunmaktadır. Ayrıca, ağ katmanına ek olarak nesne katmanı ve uygulama katmanında tanımlanan çok sayıda QoS metriği bulunmaktadır. Bu metriklerin listelenmesi adına literatür taraması yapılmıştır ve metrikler katmanlarına göre gruplanmıştır. Nesne katmanında kullanılan QoS metrikleri doğruluk, kapsama alanı, kesinlik ve enerji etkinliği üzerinedir. Nesne katmanında kullanılan QoS metrikleri ve bu metrikleri kullanan çalışmalar Çizelge 2.2’de verilmiştir.

Çizelge 2.2. Nesne katmanında kullanılan QoS metrikleri

QoS Metriği	Kullanan Çalışmalar
Kesinlik (Accuracy)	[96-98]
Kapsama Alanı (Coverage)	[98]
Enerji Tüketimi	[98-104]
Hassasiyet (Precision)	[96, 97]

Ağ katmanında ise yüksek çeşitlilikte QoS faktörleri göze çarpmaktadır. En çok gözlemlenen QoS metrikleri paket gecikmesi, paket taşıma kapasitesi, enerji etkinliği, paket kayıp oranı ve ağ üzerindeki yüküdür. Gözlemlenen metrikler Çizelge 2.3’de ilgili makale referansları detaylı ile verilmiştir.

Çizelge 2.3. Ağ katmanında kullanılan QoS metrikleri

QoS Metriği	Çalışmalar
Kesinlik (Accuracy)	[105, 106]
Veri Hızı (Data Rate)	[107-109]
Hata Oranı (Error Rate)	[96, 110, 111]
Tekrar Gönderim (Retransmission)	[112]
Bant Genişliği (Bandwidth)	[98, 102, 106, 111, 113-116]
Ağ Verimliliği (Channel/Network Utilization)	[112, 117]
Bağlantı Seviyesi (Connectivity)	[118]
Maliyet (Cost)	[99, 119-121]
Kapsama Alanı (Coverage)	[122]
Gecikme (Latency)	[37, 98, 101-103, 105-107, 110, 112, 113, 115, 117, 121-138]
Enerji Tüketimi/Verimliliği (Energy Consumption/Efficiency)	[98, 114, 121, 124, 130, 135, 139-142]
Adalet (Fairness)	[113]
Heterojenlik (Heterogeneity/Interoperability)	[104, 143]
Jitter	[98, 102, 112, 125]
Ağ Yüğü (Network Load)	[101, 106, 112, 117, 144, 145]
Kayıp Oranı (Loss Ratio)	[100, 102, 106, 112, 117, 125, 128, 129, 135, 136, 138, 146-148]
Güvenilirlik (Reliability)	[133, 138, 149]
Verim/Kapasite (Throughput)	[33, 98, 100-103, 106, 111-114, 117, 124, 127, 131, 136, 139, 146, 150]
Ağ Ömrü (Network Lifetime)	[98, 135]

Uygulama katmanında ise en çok gözlemlenen metrikler maliyet, uçtan uca gecikme (servis zamanı), enerji etkinliği, güvenilirlik ve kaynakların verimli kullanımı olarak gözlemlenmiştir. Gözlemlenen metrikler Çizelge 2.4'te ilgili makale referansları ile verilmiştir.

Çizelge 2.4. Uygulama katmanında kullanılan QoS metrikleri

QoS Metriği	Çalışmalar
Kesinlik (Accuracy)	[151]
Uygunluk (Availability)	[102, 104, 142]
Maliyet (Cost)	[98, 102, 107, 116, 122, 134, 152-160]
İçerik Kalitesi (Content Quality)	[110, 161]
Zaman Sınırı (Deadline)	[33, 159, 162]
Enerji Tüketimi (Energy Consumption)	[33, 122, 134, 149, 159, 163-168]
Servis Zamanı (Service Delay)	[98, 102, 115, 143, 144, 146, 148, 155, 157, 159, 160, 163, 165, 166, 168-173]
Adalet (Fairness)	[116]
Yük (Load)	[98, 102, 164]
Gizlilik (Privacy)	[156]
Güvenilirlik (Reliability)	[98, 102, 104, 122, 151, 154, 156, 162]
Kaynak Verimliliği (Resource Utilization)	[103, 104, 111, 113, 120, 145, 147, 157, 158, 165, 167, 168, 171, 172, 174]
Cevap Zamanı (Response Time)	[111, 166]
Ölçeklenebilirlik (Scalability)	[104, 149]
Güvenlik (Security)	[102, 104, 156]

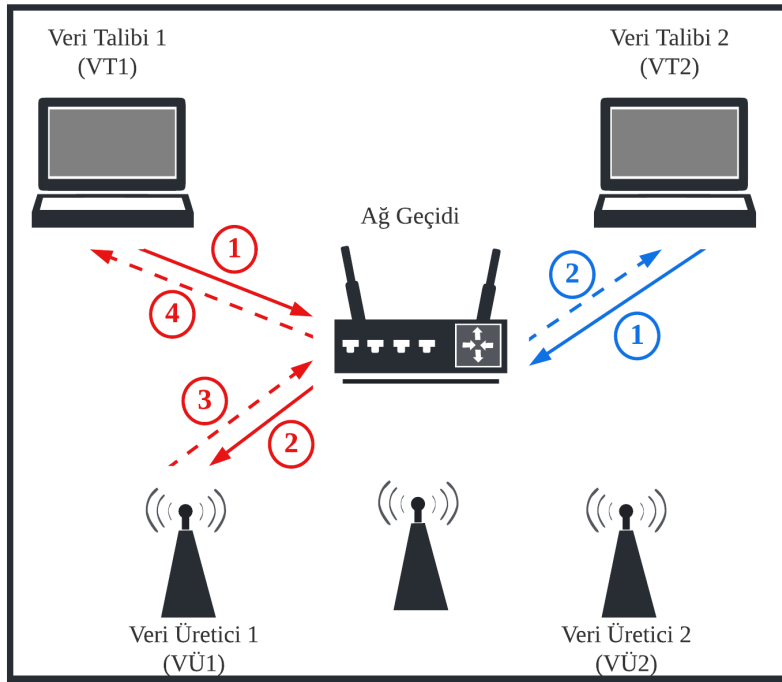
2.2.3. Kullanılan teknikler

IoT servislerinin belirli QoS seviyelerini garanti altına alacak şekilde sağlanması karmaşıklığı yüksek bir problemdir. Bu sebeple QoS-kritik problemlerin optimizasyon problemi olarak formüle edilip çözülmesi gereklidir [17]. Ancak çözüm uzayını etkin bir şekilde arayabilen optimizasyon algoritmaları bile IoT ağların büyüklüğü, heterojen yapısı ve dinamizmi sebebiyle yetersiz kalabilmektedir. Bahsedilen karmaşıklıkta problemlerin çözülebilmesi ve dolayısıyla QoS garantileri altında çalışabilen bir IoT ağ sağlanabilmesi için ek tekniklerin kullanılabilmesi gereklidir. Bu bölümde IoT ağların servisleri etkin bir

şekilde sağlmasına yardım eden teknikler verilecektir. Verilen teknikler ağ ve donanım kullanımını optimize ederek ve yönetimi kolaylaştırarak ağın QoS desteğini artırır [16].

Veri önbellekleme (content caching)

Veri önbellekleme tekniği popüler IoT verisinin kopyalarının ağ cihazlarında veya seçili IoT nesnelerinde tutulmasını önerir. IoT ağda önbellek kullanımı yapılmadığı zaman her veri talebinde veri kaynağına gidilmesi gerekir, veri çok daha uzak bir konumdan getirilir. Dolayısıyla hem gecikme yaşanır hem de ağ kullanımı artar. Önbellek kullanımına dair bir görsel Şekil 2.2’de verilmiştir.



Şekil 2.2. Veri önbellekleme örnek senaryosu

Şekilde iki senaryo verilmiştir, sırasıyla önbellekli ve önbelleksiz olmak üzere. Kırmızı oklarla gönderilen ve VT1, VÜ1 ve ağ geçidi arasında geçen birinci senaryo kapsamında VT1 VÜ1’den veri talep eder. VT1 veriyi ağ geçidinden talep eder. Ardından ağ geçidi veri talebini VÜ1’e iletir. VÜ1 talebi alır ve talep edilen veriyi ağ geçidine gönderir. Ağ geçidi VÜ1’den gelen veriyi VT1’e iletir. Verinin VT1’e ulaşması ile birlikte veri aktarımı toplamda 4 iletimde tamamlanmış olur. Mavi oklarla gönderilen ve VT2, VÜ2 ve ağ geçidi arasında geçen ikinci senaryo kapsamında VT2 VÜ2’den veri talep eder. VT2 talebi ağ geçidine gönderir. Ağ geçidi verinin kendi önbelleğinde saklandığını tespit eder ve VT2’ye

gönderir. Veri transferi VÜ2 ile iletişime gerek kalmadan toplam 2 iletimde tamamlanır. Senaryoların da gösterdiği üzere IoT verisinin önbellekte saklanması iletim sayısını ve gecikmeyi azaltır, bant genişliğinin daha verimli kullanılmasını sağlar, özetle ağ kullanımını optimize eder. Literatürde verinin önbellekte saklanması (önbellekleme) problemi NP-hard bir problem olarak tanımlanmaktadır [16].

Çalışmalarında [152] Zhu ve arkadaşları önbellekleme problemini Markov Karar Süreci (Markov Decision Process / MDP) olarak formüle etmişlerdir. Veri tazeliği ve iletişim maliyeti arasındaki dengeyi tanımlamış ve maliyet fonksiyonu olarak kullanmışlardır. Problemin çözümü için Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning / DRL) algoritmasını kullanmışlardır.

Zhang ve Liu çalışmalarında [175] veri önbellekleme sürecinin optimizasyonuna eğilmişlerdir. Çalışmanın odağında IoT ağların karakteristik özelliklerinden biri olan heterojen yapısı konulmuştur. Makalede ele alınan senaryoda veri nesnelerde veya uç sunucularda önbelleklenebilir. Nesnelerin ve uç sunucuların önbellekleme için kullanılabilir bellekleri de gerçek dünyaya uygun olması amacıyla değişken olarak tanımlanmıştır. Problemin çözümü için olasılıksal bir yaklaşım benimsenmiştir. Öncelikle, problem bir optimizasyon problemi olarak formüle edilmiştir ve Lagrangian Çarpımı (Lagrangian Multiplier / LM) kullanılarak çözülmüştür. Ardından, olasılıksal bir algoritma ile optimal önbellekleme olasılıkları elde edilmiştir. Son adım olarak, nesne ve uç sunucuların veri saklama kapasiteleri ağ modeline eklenmiştir ve problem konveks programlama (Convex Programming / CP) ile problem çözülmüştür.

Radyo erişimli ağlar (Radio Access networks / RAN) teknolojisi beşinci nesil ağlar için önerilen mimariler arasında önemli bir tanesidir. Çalışmalarında Yao ve arkadaşları [120] bulut tabanlı RAN mimarilerinde (Cloud Radio Access Networks / C-RAN) önbellekleme için sezgisel tabanlı bir yaklaşım önermişlerdir. Ele alınan önbellekleme optimizasyonu problemi iki alt probleme bölünmüştür, sırasıyla veri için bellekte yer ayrılması ve verinin yerleştirilmesi problemleri. Problem tamsayı lineer programlama (Integer linear programming / ILP) ile bileşik bir optimizasyon problemi olarak formüle edilmiş, iki farklı sezgisel algoritma ile çözülmüştür.

FC kavramı çekirdek ağ fonksiyonlarını ağın uçlarına yaklaştırarak ağların QoS desteğini arttırmaktadır. FC ile RAN teknolojisi bir araya getiren sis tabanlı RAN mimarileri (Fog Radio Access Networks / F-RAN) önbellekleme işlemini ağın uçlarına getirerek ağ kullanımını optimize eder, dolayısıyla QoS seviyelerini arttırır. Çalışmaların Jiang ve arkadaşları [169] F-RAN mimarisi için bir önbellekleme yaklaşımı önermişlerdir. Önerilen yaklaşım, talep edilen verilerin önbellekte bulunma ihtimalini maksimize etmeye yöneliktir. Öncelikle kullanıcılar analiz edilir ve kullanıcı tercihleri ortaya çıkarılır. Ardından, kullanıcı tercihleri ve ağ topolojisi kullanılarak verilerin popülerliği çıkarılır. Son adımda derin q-öğrenme tabanlı algoritma (Deep Q-Learning Based Algorithm / DQLBA) ile kullanıcı tercihleri ve veri popüleritesi kullanılarak en optimal önbellekleme stratejisi elde edilir.

Çalışanlarında Hua ve arkadaşları [176] FC temelli bir önbellekleme yaklaşımı önermişlerdir. Önerilen yaklaşım verinin hem ağ hem de kullanıcı cihazlarında tutulmasını önerir. Önbelleklenecek olan veri popüleritesine göre, önbellekleme konumları ise veri iletim hatlarına olan yakınlıklarına göre seçilir. Ağ tıkanıklıklarının önlenmesi amacıyla proaktif ve reaktif mekanizmalar tanımlanmıştır. Test sonuçlarına göre önerilen yöntemin iletim ağlarında dengeli yük dağıttığı ve önbellekleme başarısının arttığı gözlemlenmiştir.

Khodaparas ve arkadaşlarının önerdiği yaklaşım [177] önbellekleme işlemi için SDN teknolojisinin kullanılmasına dairdir. Önerilen yöntemde göre nesneler kümelere ayrılır, küme içi kararlar küme yöneticisi nesne tarafından verilir. Global kararlar için SDN tabanlı bir önbellekleme kontrolcüsü kullanılır. Karar alımlarında enerji seviyeleri, bellek miktarları, içerik değeri ve ağ kullanımı gibi çok sayıda kriter göz önüne alınır.

Son zamanlarda yapılan çalışmalar enerji tüketimini bir QoS metriği olarak tanımlamaktadır [16, 17]. Enerji tüketiminin minimize edilmesi IoT için kritiktir. Nesnelerin sıkça çalışması ve veri iletimi yapması dedike enerji kaynağı olmayan ve batarya ile çalışan nesnelerin enerjisinin bitmesine ve deaktive olmalarına sebebiyet verebilir. Önbellekleme enerji tüketiminin minimize edilmesi amacıyla da kullanılmaktadır. Çalışmalarında Yao ve Ansari [99] enerji toplayan IoT ağlar için bir önbellekleme yapısı önermişlerdir. Enerji taşıyıcıları ve ağ geçitleri arasında bir Stackelberg oyunu formüle edilmiştir. Ağ geçitleri önbellekleme operasyonunu optimize etmeye odaklanırken enerji taşıyıcıları enerji tüketimini minimize etmeye çalışır. Önerilen yapı iki objektif arasında dengeleyen, en optimize stratejiyi elde etmeye odaklanır.

Zhang ve arkadaşları çalışmalarında [119] enerji tüketimini azaltmak ve QoS seviyesini yükseltmek amacıyla çoklu ajan temelli bir yaklaşım önermişlerdir. Uç sunucular önbelleklerini daha verimli kullanabilmek ve daha popüler verileri saklayabilmek adına birbirleri ile işbirliği yaparlar. Bu amaçla DRL tabanlı bir yöntem önerilmiştir. Her uç sunucu bir ajan olarak tanımlanır. Her ajan yerel önbellekleme kararlarını vermekle sorumludur. Global yönetim için ise bir merkezi sunucu vardır. Merkezi sunucu ajanların yerel kararlarını değerlendirir ve kararlara dair görüş gönderir. Ajanlar merkezi sunucudan gelen görüşleri kullanarak kendi karar verme mekanizmalarını geliştirirler.

Bir diğer çalışmalarında [126] Yao ve arkadaşları FC temelli IoT ağlar için bir önbellekleme mekanizması önermişlerdir. Önerilen yöntemde problem ILP ile formüle edilir, MDP ile modellenir ve DRL ile çözülür. Çözülme sırasında veri iletim süresi, veri tazeliği ve bellek miktarı kısıtları altında minimize edilir.

Ağ cihazlarında veri önbelleklenmesi toplam ağ trafiğini ve nesnelerin enerji tüketimini azaltır. Çalışmalarında [130] Zhang ve arkadaşları ortak çalışma prensibine dayanan bir önbellekleme yaklaşımı sunmuşlardır. Önerilen yaklaşımda ağ cihazları önbelleklerinde veri saklar. Bu sayede nesnelerin uyku süresi arttırılır, fazla enerji tüketiminde azalacak olan çalışma ömürleri artar ve dolayısıyla ağın operasyonel olarak kalacağı süre de artar. Değişken olan veri talep hızlarına adapte olmak adına kayan pencere tabanlı bir mekanizma tanımlanmıştır.

Çalışmalarında Sun ve arkadaşları [163] CoAP iletim protokolüne dayanan bir önbellekleme mekanizması önermişlerdir. Önerilen yöntem CoAP'ın yayınla / abone ol mekanizmasını bir aracı üzerinde tanımlayarak verinin saklanması ve taleplere cevap olarak gönderilmesini sağlar. Aracıda oluşabilecek tıkanmaları ve veri popülerliğini tespit edebilmek için enerji farkında ve gecikme garantili dinamik kaynak ayrılması (energy aware and latency guaranteed dynamic resource caching / EASE) algoritmasını önermişlerdir.

Prerna ve arkadaşları çalışmalarında [110] IoT nesneleri arasındaki M2M iletişimde önbellekleme problemine eğilmişlerdir. Problem çok etiketli sınıflandırma problemi olarak formüle edilmiş, karar ağacı (Decision Tree) ve rastgele orman (Random Forest) algoritmaları ile çözülmüştür. Önerilen yöntem enerji tüketimini ve gecikmeyi optimize eder.

Chen ve arkadaşları [141] QoS kısıtları altında önbellekleme problemine eğilmişlerdir. Sıkça kullanılan derin öğrenme modellerinden biri olan uzun kısa süreli bellek (long short term memory \ LSTM) ile veri talepleri öngörülmüştür. Ardından sezgisel bir algoritma önbellekleme konumlarının ve içeriklerinin belirlenmesi amacıyla kullanılmak üzere geliştirilmiştir. Test sonuçları önerilen yöntemin QoS seviyelerinde kayıp olmadan enerji etkinliğini arttırdığını ortaya koymuştur.

Çalışmalarında [161] Han ve arkadaşları yüksek kaliteli video yayınlarında önbellekleme problemine eğilmişlerdir. Önerilen yöntem uç sunucularda başarılı iletim oranını maksimize etmeye odaklanmaktadır. Önerilen yöntem uyarınca öncelikle yerel önbellekleme kararları alınır, ardından yerel kararlar kullanılarak global kararlar yinelemeli bir şekilde alınır.

Şekil 2.2’de önbellekleme sürecinin ağ kullanımını optimize ettiğini ve ağ gecikmesini düşürdüğünü gösteren bir senaryo verilmiştir. Ancak önbellekleme sürecinin IoT’de QoS seviyelerine katkıları çok daha fazladır. IoT’de QoS odaklı önbellekleme hakkında literatürde bulunan çalışmalar Çizelge 2.5’te listelenmiştir.

Çizelge 2.5. QoS hedefli önbellekleme çalışmaları

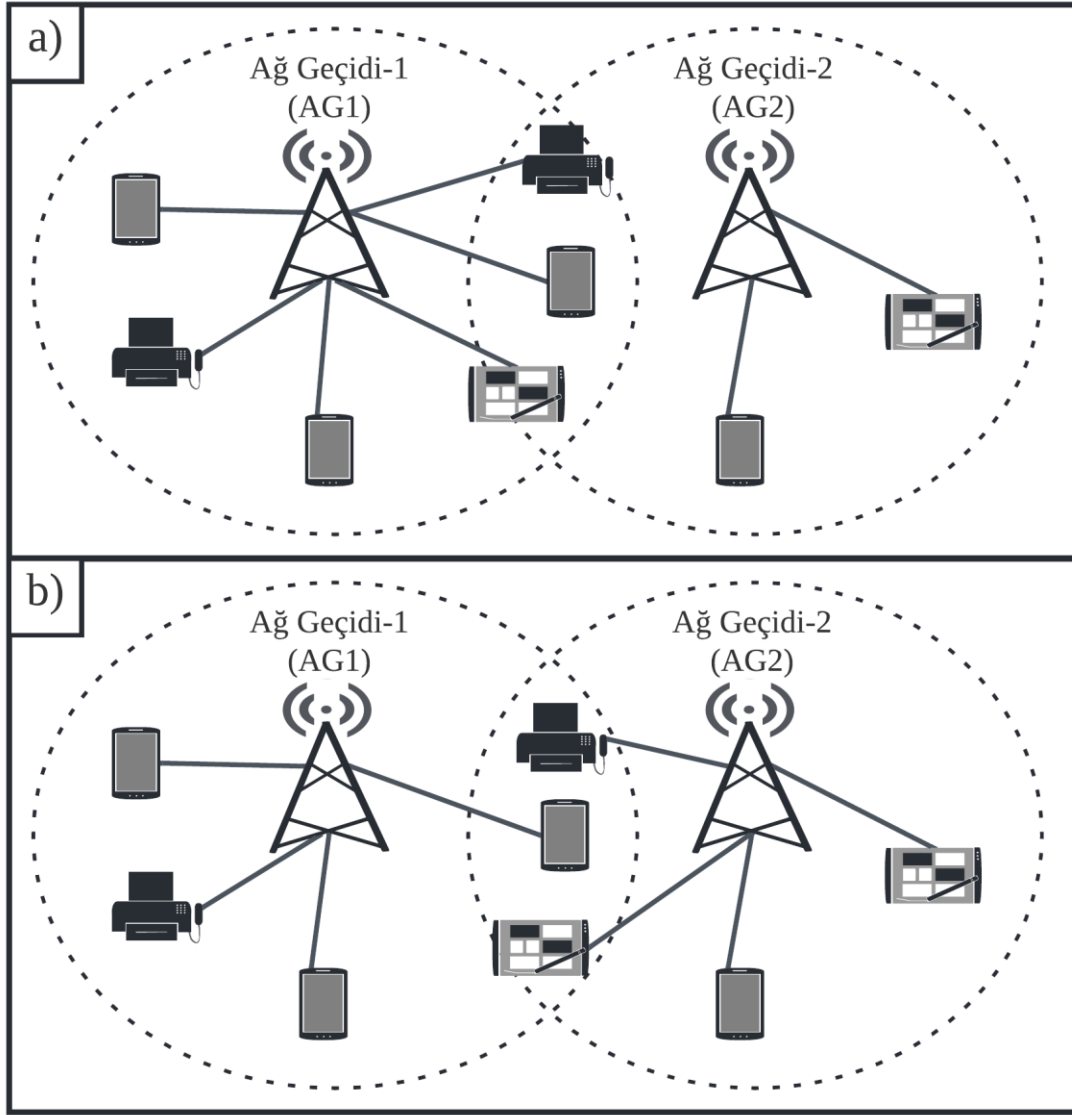
Çalışma	Önbellekleme Konumu	Platform	QoS Metrikleri
[152]	Uç Sunucular	IoT Ağlar	Ağ Metrikleri
[175]	Uç Sunucular, Nesneler	IoT Ağlar	Ağ Metrikleri
[120]	Ağ Cihazları	C-RAN	Ağ Metrikleri
[169]	Sis Sunucular, Nesneler	F-RAN	Ağ Metrikleri
[176]	Sis Sunucular	ICN-IoT Ağlar	Ağ Metrikleri, Gecikme
[177]	Nesneler	CCN-IoT Ağlar	Ağ Metrikleri, Gecikme
[99]	Ağ Geçitleri	Enerji Destekli IoT Ağlar	Enerji Tüketimi
[119]	Uç Sunucular	IoT Ağlar	Enerji Tüketimi, Ağ Metrikleri
[126]	Ağ Geçitleri	IoT Ağlar	Enerji Tüketimi, Ağ Metrikleri
[130]	Ağ Cihazları	ICN-IoT Ağlar	Enerji Tüketimi, Gecikme
[163]	Aracı Sunucu	IoT Ağlar	Enerji Tüketimi, Gecikme
[110]	Ağ Cihazları, Ağ Geçitleri	IoT Ağlar	Enerji Tüketimi, Gecikme
[161]	Uç Sunucular	IoT Ağlar	İçerik Kalitesi
[141]	Ağ Cihazları	Blokchain tabanlı IoT Ağlar	Enerji Tüketimi

Çizelgede görüldüğü üzere veri önbellekleme yüksek oranda gecikme, bant genişliği, kapasite (throughput) gibi IoT'nin ağ katmanına yönelik QoS metriklerine odaklanmaktadır. Ancak çizelgenin ortaya çıkardığı üzere enerji verimliliği önbellekleme sürecinin

kazandırdığı bir başka faktördür. Çizelgede gözlemlenebilen bir diğer değerli çıkarım önbellekleme konumlarının çeşitliliğidir. Önbellekleme konumlarının nesneler, uç ve sis düğümleri, ağ geçitleri, araçlar ve diğer ağ cihazları olabildiği görülmektedir. IoT ağlarda çoğu QoS metriği son kullanıcının veriye veya servise ne kadar hızlı ulaşabildiğine dairdir. Önbellekleme verinin son kullanıcıya kısa bir sürede ulaştırılabilmesini sağlar. Ancak, IoT ağı oluşturan cihazların bellek kapasiteleri sınırlıdır. Bu sebeple önbellekleme konumları ve önbellekte saklanacak verinin seçimi çözülmesi gereken bir problemdir. Bu bölümde verdiğimiz ve çizelgede özetlediğimiz çalışmalar bu ifadeyi doğrular niteliktedir. Çizelgeden elde edilen bir diğer çıkarım ise dağıtık ağ yapısına dairdir. Görüldüğü üzere dağıtık ağ mimarileri (sis veya uç sunucular) önbellekleme işlemini güçlendirmektedir.

Yük Dengeleme

IoT ağlar büyük miktarda veri üretir. Bir önceki bölümde bahsedilen önbellekleme süreci sıkça talep edilen veriyi son kullanıcıya yakın yerlerde tutarak QoS iyileştirmesi yapar. Yük dengeleme ise iletişim bağlantıları ve cihazlar arasında iş yükünü dengeli bir şekilde paylaştırarak QoS seviyelerini iyileştirmeyi amaçlar. İş yükünün sunucular üzerinde dengeli bir şekilde paylaşılması olası tıkanmaları önler, dolayısıyla QoS'in gerilemesine sebep olacak durumları engeller. Benzer bir şekilde iletim ağlarının dengelenmesi hata oranını ve gecikmeyi azaltır. Bu sebeplerden dolayı ağ bağlantı ve cihazlarında yük dengeleme uygulamaları IoT'de bir gerekliliktir ve sıkça çalışılan bir konudur.



Şekil 2.3. Yük dengeleme örnek senaryosu

Şekil 2.3'te yük dengelemeye dair örnek iki senaryo verilmiştir. Senaryolar sekiz adet cihazın 2 ağ geçidine bağlanması gereken bir ofis ortamında geçmektedir. Resmin üst kısmında verilen senaryoda dengesiz bağlantı görülmektedir. Senaryoda AG1'e 6 cihaz bağlanmışken AG2'ye 2 cihaz bağlanmış durumdadır. Bu durumda AG2'nin kaynakları büyük kısmı atıl durumdayken AG1'in kaynak sıkıntısı çekmesi beklenir. AG1 üzerinden ağa bağlanan cihazlar daha yüksek gecikme ve hata oranlarına sahip olacaktır. AG1 kaynak sıkıntısı çekerken AG2'de kaynaklar boş durmaktadır. Resim alt kısmında ise iş yükünün dengelendiği bir senaryo verilmiştir. AG1 ve AG2'ye dörder cihaz bağlanmıştır. İş yükü dengeli olduğundan dolayı kaynaklar eşit bir şekilde paylaşılmış ve kaynak verimliliği yükselmiştir. Bu sebepten ötürü ikinci senaryoda birinci senaryoya kıyasla daha yüksek QoS

seviyesi beklenir. IoT’de QoS odaklı yük dengeleme literatürde sıkça çalışılan bir problemdir ve NP-hard karmaşıklığa sahip olarak kabul edilir [16].

Çalışmalarında Montazerolghaem ve Yaghmaee SDN destekli IoT ağlar için bir yük dengeleme yaklaşımı önermişlerdir [113]. Yaklaşım kapsamında problem bir ILP problemi olarak formüle edilmiştir. Bulanık mantık ve zaman serisi analizine dayanan sezgisel bir model kullanılarak formüle edilen problem çözülmüştür. Önerilen model IoT sunucularının yükleri öngörür ve bulanık kurallara dayanan proaktif kararlar alır.

Al-Jamali ve arkadaşları çalışmalarında [178] SDN ağlarda kısmi yinelemeli yükselen sinir ağları (partial recurrent spike neural network / PRSNN) temelli bir yük dengeleme yaklaşımı önermişlerdir. Yöntem veri akışını öngörür ve olası problemlerde akışa müdahale eder. Ağın kontrol katmanına iki kontrolcü yerleştirilir. Bunlardan ilki veri akışını yönetir, ikincisi ise IoT nesnelerinden oluşan kümeleri yönetir. Önerilen yöntem ağ cihazlarındaki kuyrukları yönetmek için akıllı bir kuyruk modeli de önermektedir. Test sonuçları önerilen yöntemin ortalama QoS seviyelerini dikkate değer bir seviyede yükselttiğini ortaya koymaktadır.

Banaie ve arkadaşları FC temelli IoT ağlar için bir yük dengeleme mekanizması önermişlerdir [165]. Ağ geçitlerinde tıkanmaları önlemek adına yeni bir kuyruk modeli önerilmiştir. Ağ geçitlerine gelen akışların dengelenmesi için ise çok kriterli karar verme temelli (multicriteria decision-making / MCDM) bir yöntem kullanmışlardır.

Fan ve Ansari çalışmalarında FC temelli IoT ağlar için bir yük dengeleme mekanizması önermişlerdir [171]. Mekanizmanın amacı IoT veri akışlarının gecikmelerinin minimize edilmesidir. Bu amaçla nesneler ile sis sunucuları eşleştiren bir ilişkilendirme şeması yapılmıştır. Önerilen mekanizma uyarınca şema, sunucuların iş yükü ve bağlantıların yükleri kullanılarak kararlar alınır.

Önbellekleme bölümünde anlatıldığı üzere aracılar (broker) IoT verisinin son kullanıcıya yakınlaştırılmasında kullanılmaktadır. Aracıların kullanılmasıyla birlikte toplam ağ trafiği azaltılır, son kullanıcının tecrübe ettiği gecikme düşer ve genel QoS yükselir. Ancak aracılar üzerindeki yüklerin dengesiz olması bu avantajları yok edip QoS seviyelerinde düşüşe sebep olabilir. Bu sebeple aracılar üzerinde bir yük dengeleme uygulaması gereklidir. Sun ve Ansari çalışmalarında [168] aracılar üzerindeki yüklerin dengelenmesi amacıyla gecikme

farkında tekrarlı popüler veri önbellekleme (latency aware popular resource re-caching / LEARN) algoritmasını önermişlerdir. LEARN yüksek yüke sahip araçlardaki verileri daha düşük iş yüküne sahip araçlara kaydırır.

Servisler IoT kavramının temel taşıdır. SOA uyarınca nesnelerin sunduğu işlevler servisler olarak enkapsüle edilir. Son kullanıcıların talepleri sonucunda IoT servisleri kullanılır. Ancak, kullanıcı talepleri ve IoT servislerinin yükleri yönetilmez ise nesneler aşırı yük yüzünden aşırı yüklenebilir. Bu ve benzeri istenmeyen durumların oluşmasını önlemek adına IoT servislerinin arasında da bir yük dengeleme mekanizması tesis edilmelidir. Aksi takdirde QoS seviyelerinde düşüşler gözlenecektir. Çalışmalarında Yu ve arkadaşları [115] servis zincirleri için trafik dengeleme yaklaşımı önermişlerdir. Yaklaşımında SDN yardımı ile servislerin güvenilirliği ve QoS seviyeleri göz önünde bulundurularak ağ yönetilir. Yakın-optimal bir planlama elde etmek için hızlı yaklaşım algoritması (fast approximation scheme) temelli bir yöntem kullanılmıştır. Bir sonraki çalışmalarında [179] ise IoT servisleri için QoS farkında bir yük dengeleme yaklaşımı sunmuşlardır. Yük dengeleme problemi lineer programlama (Linear Programming / LP) ile formüle edilmiş ve graf tabanlı bir algoritma ile çözülmüştür. Algoritma IoT servisleri arasındaki bağımlılıklar ve QoS seviyelerini göz önüne alarak karar verir.

Çalışmalarında Nezami ve arkadaşları [180] IoT servislerinin yerleştirilmesinin QoS ve maliyet farkında yapılabilmesi adına EPOS Fog sistemini önermişlerdir. Sistem dağıtık karakterde bir çoklu ajan yaklaşımını ve işbirlikçi öğrenme yöntemlerini iki optimizasyon hedefini sağlamak için kullanır. İlk objektif iş yüklerinin dengelemesidir ve global olarak çözülür. İkinci hedef ise QoS seviyelerinin sağlanmasına da aittir ve yerel olarak çözülür. Test sonuçları önerilen sistemin iş yüklerini başarılı bir şekilde dağıttığını ve yüksek QoS seviyesi sağladığını göstermektedir.

Bu bölümde bahsedilen QoS odaklı yük dengeleme çalışmaları Çizelge 2.6'da listelenmiştir.

Çizelge 2.6. QoS hedefli iş yükü dengeleme çalışmaları

Çalışma	Yük Dengeleme Konumları	Platform	Yaklaşım
Montazerolghaem ve Yaghmaee [113]	IoT Sunucuları	SDN destekli IoT	Problem ILP ile formüle edilmiştir. Ardından bulanık mantık ve zaman serisi temelli sezgisel yaklaşım ile çözülmüştür.
Bananie ve ark. [165]	IoT Ağ Geçitleri	FC temelli IoT	Veri akışlarının dağıtılması için MCMD kullanılmıştır.
Fan ve Ansari [171]	IoT Aracıları	FC temelli IoT	Hesaplama ve iletişim yükleri için ilişkilendirme şemaları kullanılmıştır.
Sun ve Ansari [168]	IoT Aracıları	IoT	LEARN algoritması araçlarda iş yüklerinin dengelenmesi için kullanılmıştır.
Yu ve ark. [115]	IoT Servisleri	SDN destekli IoT Mobil IoT	Hızlı ilişkilendirme şeması kullanılmıştır.
Yu ve ark. [179]	IoT Servisleri	IoT	Problem LP ile formüle edilip graf temelli yaklaşımla çözülmüştür.
Nezimi ve ark. [180]	IoT Servisleri	FC temelli IoT	EPOS Fog ile iş yükü dağılım problemi global olarak çözülmüştür. Yerelde QoS garantileri sağlanmıştır.
Al-Jamali ve ark. [178]	IoT Veri Akışları	SDN destekli IoT	PRSNN ile veri akış yoğunlukları öngörülmüştür. NN temelli kümeleme ve önerilen akıllı kuyruklama yaklaşımları ile problem çözülmüştür.

Çizelgede görüldüğü üzere yük dengeleme sürecinin uygulanması genellikle SDN veya FC gibi ikincil teknolojilerin kullanılmasını gerektirmektedir. Etkin bir yük dengeleme mekanizmasının tahsis edilmesi için ağın iyi bir şekilde takip ediliyor olması gereklidir. SDN ve FC gibi teknolojiler de ağın daha iyi takip edilmesini ve yönetilmesini sağlar ve bu sebeple sıklıkla kullanılırlar. Genel tanım olarak yük dengeleme iş yükünün kaynaklar üzerine eşit şekilde paylaştırılmasına dair mekanizmaları kapsar. Kaynak tipi belirgin değildir, yük dengeleme mekanizmasına, mekanizmanın odağına veya uygulandığı ağa göre değişebilmektedir. Çizelge 2.6'da görüldüğü üzere kaynak bir cihaz (nesne, sunucu, ağ geçidi, aracı), bir servis veya bağlantı olabilir.

Değerlendirme

Önbellekleme ve yük dağılımı üzerine yapılan literatür taramaları göstermektedir eldeki kısıtlı kaynakların son kullanıcı ve IoT uygulamalarının talepleri doğrultusunda etkin bir şekilde kullanılması IoT'de QoS sağlanmasının en temel kriteridir. IoT ağların büyüklüğü ve dinamizmi göz önüne alındığında kaynakların etkin kullanımı oldukça karmaşık bir problem olmaktadır. Bu sebeple, bu alanda yapılan çalışmaların genelinde problem bir optimizasyon problemi olarak formüle edilip NP-hard karmaşıklıklarda etkin bir şekilde çözüm sağlayan optimizasyon algoritmaları ile çözülmektedir.

2.3. Bölüm Değerlendirmesi ve Tartışma

IoT ağlar ve uygulamalarının insan yaşamını devrimsel bir ölçekte değiştireceği açıktır. IoT sayesinde günlük hayatta kullanılan birçok uygulama ve cihaz çok daha akıllı hale gelmesi ve IoT devrimi öncesi varlığı hayal edilmeyen yeni uygulamaların insan hayatına girmesi beklenmektedir. Akıllı şehir uygulamaları sayesinde daha sürdürülebilir ve konforlu hayat, akıllı endüstri sayesinde daha yüksek verimde çalışan üretim ortamları, akıllı sağlık uygulamaları sayesinde daha kişiselleştirilmiş ve başarılı bir sağlık sistemi yakın bir gelecekte oldukça sıradan şeyler olacaktır.

Öngörülen IoT vizyonunun temel yapı taşlarından biri QoS'tir. QoS bir IoT uygulamasının veya servisinin uyması gereken kalite kriterlerini belirtir. Bir servisin veya uygulamanın son kullanıcıya başarılı olarak sunulması için QoS kriterlerini yerine getirilmiş olması gereklidir. Bu sebeple IoT'de QoS sağlanması üzerine çok sayıda çalışma yapılmaktadır. Ancak,

literatür taramasında görüldüğü üzere IoT’de QoS sağlanması üzerine olan literatür oldukça genştir. IoT’de tam anlamıyla QoS sağlanması adına aşılması gereken zorluklar bulunmaktadır. Aşağıda literatür taraması sonucunda gözlemlenen zorluklar ve geleceğe dair olası araştırma konuları [16] verilmiştir.

- Güvenlik IoT için oldukça kritik bir konudur. Güvenliği sağlanmayan nesneler üzerinden yapılan saldırılar IoT ağları işlevsizleştirebilir, çok yüksek gecikmelere neden olabilir ve toplanan veriyi bozabilir. Sıralanan sonuçlar uygulamaların QoS seviyelerini bozacaktır. Bu sebeple güvenlik mekanizmalarının QoS odaklı olarak tanımlanması bir gerekliliktir.
- Gizlilik IoT verisi için kritik olabilmektedir. Özellikle akıllı sağlık uygulamalarında nesnelerin topladığı veri kişisel veridir. Bu sebeple bu verilerin üçüncü partilerin (doktor ve hasta dışında bireyler) eline geçmesi engellenmelidir. Veri gizliliği literatürde kullanılan QoS metriklerinden biridir.
- Yeni Nesil İletişim Teknolojileri 5G/6G ve blockchain teknolojileri IoT ağların QoS desteklerini geliştirmektedir. Ancak, mevzu bahis teknolojiler hali hazırda oldukça yeni olduğu için bu alanda yeterince çalışma bulunmamaktadır ve araştırma boşluğu bulunmaktadır.
- Destekleyici Mimariler EC ve FC tabanlı yaklaşımları kapsar. Bu mimariler veri işleme ve saklama yeteneklerini son kullanıcıya yaklaştırarak IoT ağın daha verimli kullanılmasını, daha düşük gecikmeler ile servislerin sağlanmasını sağlar. EC ve FC sıkça kullanılan mimariler olsa dahi çoğu çalışmada önerilen yaklaşımlar QoS açısından yeterince irdelenmemektedir. Bu konuda çalışmalar yapılabilir.
- Ölçeklenebilirlik IoT’de önemli bir gereksinimdir. IoT ağların ölçeklenebilirliği konusunda çokça çalışma bulunmasına rağmen bu çalışmalar QoS perspektifini genellikle es geçmektedir. Bu sebeple, ölçeklenebilir ve QoS destekleyen IoT ağ mimarileri üzerine çalışmalar yapılmalıdır.
- Genelleyici QoS Modellemesi bu bölümde bahsedilecek son ama en önemli tartışma konusudur. Bahsedildiği üzere IoT uygulamaları ve ilgili çizelgelerde (Çizelge 2.2, Çizelge 2.3, Çizelge 2.4) görüldüğü üzere IoT uygulamalarının QoS metrikleri oldukça çeşitlidir. Ancak literatürde QoS isterlerini genelleyici bir şekilde ifade edecek, IoT’de tüm QoS isterlerine uygulanabilecek bir QoS modellemesi bulunmamaktadır. Yapılan çalışmalar spesifik QoS isterlerine odaklı olarak yapılmaktadır. Ayrıca, QoS isterleri ağın farklı katmanlarına ait olabilmektedir. QoS isterleri algılama-iletim-veri işleme sürecinin başından sonuna kadar, ağın her katmanında, uçtan uca desteklenmelidir.

IoT’de gerek anlamda QoS desteęini saęlanması iin, genelleyici bir QoS modellemesi ve modelleme zerinden geliřtirilen utan uca bir QoS desteęi saęlanmalıdır. Bu amala gerekli mimarilerin ve protokollerin geliřtirilmesi gereklidir.



3. NESNELERİN İNTERNETİNDE VERİ BÜTÜNLÜĞÜNÜN YAPAY SİNİR AĞLARI TEMELLİ YÖNTEMLER İLE SAĞLANMASI

Algılayıcı teknolojileri IoT kavramının en önemli öncül teknolojilerinden biridir. Algılayıcılar yardımı ile büyük miktarlarda IoT verisi toplanmaktadır. Toplanan veri çeşitli alanlarda (akıllı ortamlar, akıllı sağlık, akıllı tarım vb.) IoT uygulamalarında kullanılmaktadır. IoT verisi IoT kavramının karakteristik özelliklerinden dolayı büyük boyutta, yüksek hızda ve heterojen yapıdadır [18]. Bu yapıdaki verinin işlenmesi hali hazırda oldukça zordur. Ek olarak, özellikle büyük boyuttaki IoT ağlarda bataryası tükenen nesneler, zayıf bağlantı hatları, iletişim hataları ve donanımsal hatalardan kaynaklanan veri kayıpları oldukça yaygındır [181]. Bu sebeplerden ötürü veri kalitesi ve doğruluğu IoT uygulamaları için kritiktir. Veri kaybı, veri analizi sonuçlarında hatalara sebep verip IoT uygulamalarının çalışmalarını ciddi şekilde bozabilir [18].

Veri akışlarında eksik verinin tamamlanmasına dair çalışmalar literatürde bulunmaktadır. Ancak çoğu çalışma statik veri setleri üzerinedir. IoT verisi ise statik veri setlerinden çok daha zorlu bir alandır. Öncelikle veri akan veridir, değişken yapıdadır, sınırları (minimum ve maksimum) belli değildir. Bu sebeple eksik veri yerine ortalama değer veya boş değer yazılması gibi konvansiyonel yaklaşımlar uygulanabilir değildir. Akan veri alanında en büyük problemlerden biri konsept kaymasıdır. Konsept kayması veri karakteristiğinin sürekli değişmesi durumudur ve konvansiyonel eksik veri tamamlama yaklaşımları bu probleme bir çözüm sunmaz. Ancak IoT verisi işlenirken konsept kayması göz önüne alınması gereken bir problemdir. Bu sebeplerden dolayı IoT verisinin karakteristiğine uygun, olası veri kayıplarını tamamlayacak yöntemlere ihtiyaç vardır [18].

Bu bölümde anlatılacak olan çalışma [18] IoT ağlarda eksik verilerin tamamlanması için önerilen yapay sinir ağı temelli yaklaşımları içermektedir. Bu çalışma kapsamında biri derin öğrenme, üçü uyarlamalı ağ tabanlı bulanık çıkarım sistemi tabanlı dört model önerilmiştir. Modeller Intel Berkeley Laboratuvarı'nın gerçek ortamdan topladığı IoT verisi üzerinde uygulanmış [182] ve literatürde bulunan regresyon modelleri ile karşılaştırılmıştır.

ANFIS, yapay sinir ağları (Artificial Neural Network / ANN) ile bulanık çıkarım sistemlerinin (Fuzzy Inference System / FIS) avantajlarını birleştiren yöntemdir [183]. DL son zamanlarda çok ilgi çeken ve birçok alanda (bilgisayarla görme, doğal dil işleme, veri analizi) kullanılan bir yaklaşımdır [184]. DL büyük miktardaki verilerin analizi ve büyük

veri üzerinden öngörü yapma potansiyeline sahiptir. Bu sebeplerden ötürü ANFIS ve DL tabanlı eksik veri tamamlama modelleri geliştirilmiştir.

3.1. İlgili Çalışmalar

Eksik/kayıp veri, veri setlerinin ilk ortaya çıktığından beri karşılaşılan bir problemdir. Ek olarak WSN ve IoT kavramları ile birlikte üretilen veri miktarı artmış ancak ekipman kalitesi düşmüştür. Bu sebeple eksik veri problemleri de artmıştır. Bu problemin çözümü adına çalışmalar yapılmıştır. Bu bölümde bu probleme eğilen çalışmalar verilmiştir. Yapılan çalışmalar eksik veriyi tamamlama metodolojilerine göre 3 grupta incelenmiştir; istatistiki yaklaşımlar, optimizasyon temelli yaklaşımlar ve makine öğrenmesi temelli yaklaşımlar.

3.1.1. İstatistiki yaklaşımlar

Eksik verinin tamamlanması için kullanılan en eski yaklaşımlar istatistiki olanlardır. İstatistiki yaklaşımlar arasında regresyon modelleri en çok kullanılanlardır. Çalışmalarında Qin ve arkadaşları [185] eksik verilen tamamlanması için bir regresyon modeli önermişlerdir. Önerilen yöntem ortalama hata oranını (Root-Mean-Square Error / RMSE) minimize etmeye odaklanır ve her veri tamamlama sonrası kendini optimize eder. Yöntem kendini geliştirmesi ve parametrik yapıda olması ile konvansiyonel yöntemleri geride bırakmaktadır. Yöntem gerçek ve yapay veri setlerinde test edilmiştir.

Beklenti maksimizasyonu (Expected Maximization / EM) veri tamamlama için kullanılan yöntemlerden biridir. EM algoritması veri noktalarını modellerin bir karışımı olarak tutar [186]. Modellerin optimizasyonu için en büyük olabilirlik kestirimi (Maximum Likelihood Estimation / MLE) algoritması kullanılmıştır. EM algoritmasında kullanılan modeller değişebilmektedir ancak en yaygın kullanılanı Gauss dağılımı modelidir. Bu yaklaşımda her nokta, çok sayıda gauss dağılım modelinin karışımı olarak ifade edilir. Ancak veri boyutu arttıkça Gauss modeline dayanan EM algoritmasının işlem karmaşıklığı çok artmaktadır. Bu sebeple, Gauss dağılımına dayanan EM tabanlı veri tamamlama yaklaşımlarında veri boyutunu düşürecek ek mekanizmalar gereklidir. Çalışmalarında Delalleau ve arkadaşları [187] Gauss dağılımına dayanan bir EM önermişlerdir. Yüksek işlem karmaşıklığını önlemek için ağaç tabanlı bir mekanizma önerilmiştir. Ağaç yardımı ile kümeler oluşturulur ve küme-içi elde edilen ortalama değerler veri tamamlama için kullanılmıştır. Kullanılan ağaç yapısı minimum tarama ağacı (Minimum Spanning Tree / MST), kullanılan kümele

algoritması ise K-Means algoritmasıdır. Bir diğer EM tabanlı çalışma Eirola ve arkadaşları [188] tarafından yapılmıştır. Çalışma çok boyutlu veri setlerine yöneliktir ve çok boyutlu veri kümeleme (High Dimensional Data Clustering / HDDC) algoritmasını işlem karmaşıklığını azaltmak için kullanır.

3.1.2. Optimizasyon tabanlı yaklaşımlar

Optimizasyon algoritmaları çözüm uzayının geniş olduğu NP-hard ve üstü karmaşıklıkta problemler için kullanılır. Bu algoritmalar evrimsel ve yinelemeli yöntemler ile çözüm uzayını etkin bir şekilde keşfeder, optimal çözüme yakın çözümleri düşük işlem karmaşıklığı ile bulur. Eksik veri problemi bir optimizasyon problemi olmasa dahi, optimizasyon algoritmalarının düşük işlem karmaşıklığı ile çözüm bulabilme yeteneği bu algoritmaların büyük boyuttaki verilerde eksik verilerin tamamlanması için kullanılabilir olmasını sağlamaktadır. En çok kullanılan optimizasyon algoritmalar genetik algoritma (Genetic Algorithm / GA), parçacık sürü optimizasyonu (Particle Swarm Optimization / PSO) ve karınca kolonisi optimizasyonu (Ant Colony Optimization / ACO) algoritmalarıdır.

GA türlerin evrimini taklit eden bir algoritmadır[189]. Bir probleme dair çözümlerden oluşan popülasyon rastgele şekilde oluşturulur. Her çözüm belirlenen bir uygunluk fonksiyonuna göre puanlandırılır. Ardından, eldeki popülasyonun en yüksek puana sahip bireyleri birbirleri ile çaprazlanarak bir sonraki nesil oluşturulur. Puanlama ve yeni nesil oluşturma süreci algoritma bir bitiş şartını sağlayana kadar devam eder. Her nesil oluşturulmasında yerel minimumlardan kaçınmak için rastgele bireyler popülasyona eklenir ve çaprazlama işleminde mutasyonlar yapılır. Çalışmalarında [190] Garcia ve arkadaşları GA tabanlı bir eksik veri yaklaşımı sunmaktadır. Tamamlanacak olan veri bir matris olarak tanımlanır. Tamamlanması gereken veriyi X , X 'i tamamlayacak olan veri matrisini Y , tamamlanmış veriyi de Z olarak kabul edelim. Yaklaşımda X ve Z arasındaki mesafeyi minimize edecek bir uygunluk fonksiyonu tanımlanmıştır. Y 'lerden oluşan bir popülasyon GA ile bahsedilen uygunluk fonksiyonu kullanılarak optimize edilmiştir. Test sonuçları önerilen yöntemin EM tabanlı yöntemlerden daha başarılı, esnek ve tepkisel olduğunu ortaya koymuştur. GA tek uygunluk fonksiyonuna göre çalışan bir algoritmadır. Dolayısıyla, birden çok optimizasyon hedefi olduğu durumlarda yetersiz kalır. Bu noktada çok hedefli genetik algoritmalar (Multi-Objective Genetic Algorithm / MOGA) kullanılır. MOGA çok sayıda objektifi aynı anda optimize etme yeteneğine sahiptir. Çalışmalarında Lobato ve arkadaşları [191] MOGA ailesi

içerisinde en çok kullanılan algoritmalarından biri NSGA-II algoritmasına dayanan MOGAImp algoritmasını önermişlerdir. MOGAImp iki objektife sahiptir, bunlar doğruluk ve RMSE metriğidir. Bu iki objektifi puanlandıran bir uygunluk fonksiyonu geliştirilmiş ve kullanılmıştır.

ACO tabanlı yöntemler graf şeklinde ifade edilebilen problemler için kullanılabilir. Bu prensipten hareketle Priya ve arkadaşları çalışmalarında [192] eksik veri problemini bir graf problemi olarak formüle etmiş ve ACO ile çözmüşlerdir. ACO'nun yerel minimumlara düşme zaafını aşmak için çift popülasyonlu bir yaklaşım geliştirmişlerdir. Çift popülasyon kullanılması ve her iterasyonda popülasyonlar arası çözüm transferi yapılması popülasyonlardaki çeşitliliği arttırmıştır. Çift popülasyonlu bayes ACO (Dual Re-populated Bayesian Ant Colony Optimization / DPBACO) olarak isimlendirilen yöntem test sonuçlarında başarısını ortaya koymuştur.

PSO sürü hareketlerini taklit eden bir algoritmadır ve basitliği, kesinliği ve hızlı çözüm bulmak yeteneği sebebiyle kullanılmaktadır [189]. Çalışmalarında Nekouie ve Moattar [193] meme kanseri teşhis veri için bir veri tamamlama algoritması sunmuşlardır. Kaotik azaltılmış uyarlanabilir PSO (Chaotic Reduced Adaptive PSO / CRAPSO) isimli algoritma bayes ağırları kullanan PSO algoritmasının yerel minimumlara düşme zaafını aşmayı başarmıştır.

3.1.3. Makina öğrenmesi tabanlı yaklaşımlar

Günümüzde eksik veri tamamlama problemi için yapay zeka temelli yaklaşımların kullanımı artmaktadır[194]. Bu yaklaşımların kullandığı algoritmalar arasında en popüler olanlardan biri k en yakın komşu (k-Nearest Neighbour / k-NN) algoritmasıdır. k-NN bir sınıflandırma algoritması olmasına rağmen, veri çok boyutlu olduğu ve gözlem sayısı düşük olduğu durumlarda yüksek başarı ile kullanılabilir [195]. Bir veri eksik olduğu durumda k-NN ile eksik veriye benzer k adet veri noktası seçilir ve eksik verinin tahmin edilmesi için kullanılır. Bu alanda yapılan ilk çalışmalardan biri Troyanskaya ve arkadaşlarına aittir[196]. Çalışmalarında DNA mikro dizilerinde eksik olan verilerin tamamlanması için kullanılmıştır ve her veri noktası eşit bir şekilde ele alınmıştır. Garcia-Laencia ve arkadaşlarının çalışmasında [190] ise veri noktaları ağırlıklandırılmıştır. Test sonuçları ağırlıklandırma yaklaşımının, algoritma doğruluğu arttırdığını göstermiştir. Bir başka ağırlıklandırılmış k-NN kullanılan çalışma Tutz ve Ramzan [195] tarafından yayınlanmıştır. Test sonuçları

Garcia-Laencia ve arkadaşlarının çalışmasına paralel bir şekilde ağırlıklandırmanın etkisini göstermektedir.

ANN insan beyin yapısını taklit eden bir yapay zeka yöntemidir. Eksik veri problemi için ANN ve varyasyonları sıkça kullanılmaktadır. Otomatik ilişkilendirmeli sinir ağları (Auto-associative Neural Networks / AANN) sinir ağının giriş ve çıkış katmanları arasında bir darboğaz oluşturur, lineer ve lineer olmayan ilişkileri öğrenme yeteneğine sahiptir. Çalışmalarında Abdella ve Marvela [197] ve Nelwamondo ve arkadaşları [198] AANN temelli modelleri eksik veri tamamlanması için kullanmışlardır. Hata fonksiyonunun optimize edilmesi için GA kullanmışlardır. Optimizasyon ve NN kullanımını birlikte uygulayan bir başka çalışma Ravi ve Krishna tarafından yayınlanmıştır[199]. Çalışmada PSO algoritması ile geliştirilen otomatik ilişkilendirmeli NN modelinin gizli katmanı optimize edilmiştir. Eksik veri problemi için kullanılan bir başka NN varyasyonu aşırı öğrenme makinasıdır (Extreme Learning Machine / ELM). Eğitim sırasında geri besleme (Back-propagation) mekanizması içermediği için diğer NN varyasyonlarına göre çok daha düşük eğitim süresine sahiptir [200, 201]. Çalışmalarında Sovilj ve arkadaşlarının yayınladığı yaklaşım [202], EM ve ELM algoritmalarını beraber kullanılmasını önerir. EM ile çok sayıda veri tamamlama yapılır ve tamamlanmış değerler üzerinde son doğrulamaları ELM yapar. ELM kullanan bir başka yaklaşım Lana ve arkadaşları tarafından [203] yayınlanmıştır. Çalışmada eksik veri tamamlama amacıyla ELM kullanılır. ELM kendi içerisinde nöronlar arası ağırlıkları optimize eden bir yaklaşıma sahiptir değildir. Çalışma, bu eksikliğin GA ile kapatılmasını önerir. Bahsedilenlere ek olarak çeşitli NN varyasyonları eksik veri tamamlama problemi için kullanılmaktadır. Örnek vermek gerekirse çok katmanlı perseptron ağlar (Multi-Layer Perceptron Networks / MLPN)[204], kendini organize eden haritalar (Self Organizing Maps / SOM) [205, 206], olasılıksal sinir ağları (probabilistic neural networks / PNN) [207] en çok kullanılan NN varyasyonlarıdır.

Kümeleme yaklaşımları eksik veri için kullanılmaktadır. Eksik veri aynı kümeyi paylaştığı veri noktalarından tahmin edilebilir. Kümeleme algoritmaları arasında bulanık c-means kümeleme (Fuzzy c-means Clustering / FCM) eksik veri tamamlama problemi için sıkça kullanılmaktadır [208-210].

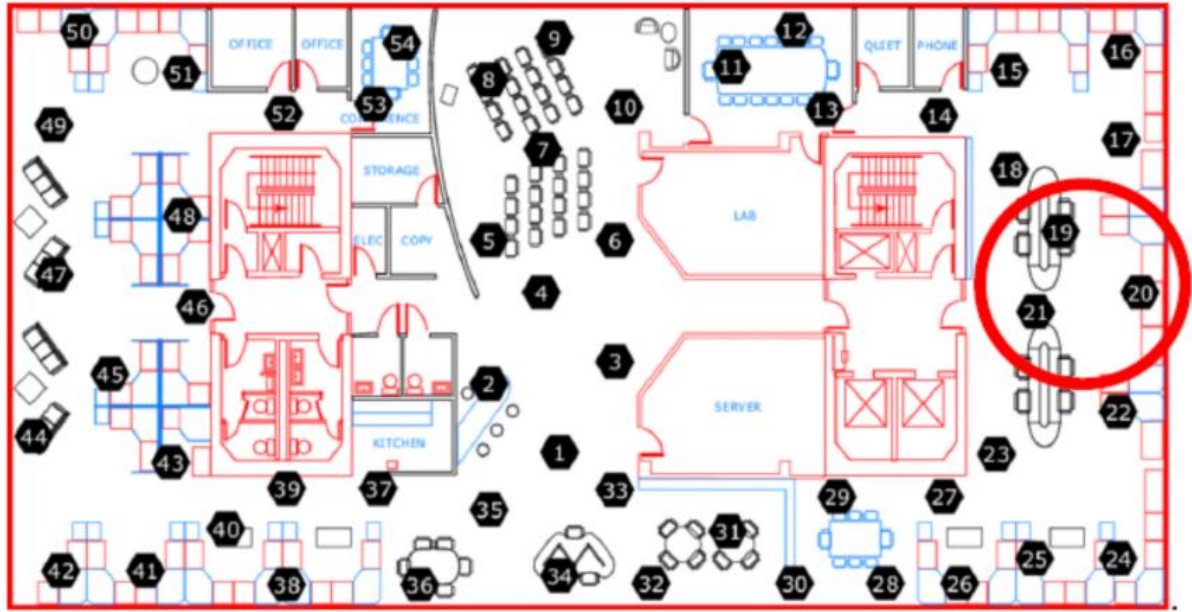
Bu bölümde bahsedilen çalışmalar Çizelge 3.1’de özetlenmiştir.

Çizelge 3.1. Taranan eksik veri çalışmaları

Çalışma	Yaklaşım	Algoritma
Qin ve ark. [185]	İstatistiki	Regresyon Algoritmaları
Delalleau ve ark. [187]	İstatistiki	EM, K-means
Eirola ve ark. [188]	İstatistiki	EM, HDDC
Garcia ve ark. [190]	Optimizasyon	GA
Lobato ve ark. [191]	Optimizasyon	NSGA-II
Priya ve ark. [192]	Optimizasyon	ACO
Nekouie ve Moattar [193]	Optimizasyon	PSO
Troyanskaya ve ark. [196].	Makine Öğrenmesi	k-NN
Tutz ve Ramzan [195]	Makine Öğrenmesi	Ağırlıklandırılmış k-NN
Garcia-Laencia ve ark. [190]	Makine Öğrenmesi	Ağırlıklandırılmış k-NN
Abdella ve Marvela [197]	Makine Öğrenmesi	AANN, GA
Nelwamondo ve ark. [198]	Makine Öğrenmesi	AANN, GA
Ravi ve Krishna [199]	Makine Öğrenmesi	AANN, PSO
Sovilj ve ark. [202]	Makine Öğrenmesi	ELM, EM
Lana ve ark. [203]	Makine Öğrenmesi	ELM, GA
Silva-Ramirez ve ark. [204]	Makine Öğrenmesi	MLPN
Folguera ve ark. [205]	Makine Öğrenmesi	SOM
Saitoh [206]	Makine Öğrenmesi	SOM
Nishanth ve ark. [207]	Makine Öğrenmesi	PNN
Li ve ark. [208]	Makine Öğrenmesi	FCM
Sefidian ve Daneshpour [209]	Makine Öğrenmesi	FCM
Zhang ve ark. [210]	Makine Öğrenmesi	FCM

3.2. Veri Seti

Bu bölümde anlatılan çalışmada Intel firmasının Berkeley'deki araştırma laboratuvarına yerleştirdiği algılayıcılar ile topladığı Intel Berkeley Research Lab veri seti [182] kullanılmıştır. Laboratuvara yerleştirilen 54 algılayıcı 2004 yılında 28 Şubat ile 5 Nisan arası veri toplanmıştır. Toplam 2.3 milyon algılayıcı okumasında ortamın nem oranı, sıcaklık değeri ve ışık yoğunluğu ölçülmektedir. Sıcaklık değeri celsius, nem oranı yüzde, ışık yoğunluğu ise lux türündedir. Verinin toplandığı Berkeley laboratuvarının ve algılayıcıların konumlarının gösterildiği görsel Şekil 3.1'de verilmiştir.



Şekil 3.1. Intel Berkeley laboratuvar, algılayıcı konumları

Bu bölümde anlatılan çalışmada Şekil 3.1'de kırmızı daire içerisinde gösterilen 19, 20 ve 21 numaralı algılayıcılar tarafından toplanan sıcaklık, nem ve ışık değerleri kullanılmıştır. Mevzu bahis algılayıcıların seçilmesi ve kullanılmasının nedeni belirtilen algılayıcıların topladığı verilerin doluluğudur. Algılayıcıların 29 Mayıs ile 7 Mart tarihleri arasında kaydettikleri algılayıcı okumaları 3 dakikalık pencereler olarak gruplandırıldığı zaman toplanan veri yüzde yüz doluluğa sahiptir. Diğer algılayıcılarda veri doluluğu oranı çok daha düşüktür. Bir diğer seçim nedeni ise algılayıcıların birbirine yakın konumda olmasıdır. Yakın konumda olan algılayıcıların birbirine yakın örüntüleri takip etmesi, dolayısıyla önerilen yöntemlerin performansının daha net bir şekilde ortaya koyması beklenmektedir.

Toplanan veri iki şekilde kullanılmıştır.

- Üç algılayıcıdan toplanan veri tek bir veri seti haline getirilmiş ve eksik veri tamamlama modelinin eğitimi için kullanılmıştır. Bu yaklaşım DL temelli model için kullanılmıştır.
- Üç algılayıcıdan toplanan veri farklı üç farklı modelin eğitimi için kullanılmıştır. Modelin başarısını elde etmek için farklı algılayıcıların verileri ile eğitilen üç modelin ortalaması alınmıştır. Bu yaklaşım ANFIS temelli model için kullanılmıştır.

Verinin istatistiki analizi Çizelge 3.2’de verilmiştir. Çizelgede görüldüğü üzere veri değerleri normal bir dağılım takip etmemektedir.

Çizelge 3.2. Veri setinin istatistiki analiz sonuçları

İstatistiki Parametre	Sıcaklık (C)	Nem Oranı (%)	Işık Yoğunluğu (Lux)
Minimum	15,27	14,43	0,46
Maksimum	37,66	51,72	1847,36
Ortalama	23,216	35,48	323,88
Standart Sapma	4,651	7,76	436,85

Veride normal dağılım gözlemlenemediği için veri tipleri arasında korelasyon analizi yapılmıştır. Spearman korelasyon katsayısı metoduna göre yapılan analizin sonuçları Şekil 3.2’de verilmiştir.

Sıcaklık Işık Yoğunluğu Nem Oranı	1.0	-0.66	0.55
	-0.66	1.0	-0.31
	0.55	-0.31	1.0
	Sıcaklık	Işık Yoğunluğu	Nem Oranı

Şekil 3.2. Spearman korelasyon analizi sonuçları

Veri türleri arasındaki korelasyonu gösteren Şekil 3.2’de görüldüğü üzere diğer veri türleri ile korelasyonu en yüksek olan veri türü sıcaklıktır. Sıcaklığın ardından nem oranı gelmektedir. En düşük ilişkilendirilebilir tür ise ışık yoğunluğudur.

Hem tek bir veri türüne dair veri içerisinde hem de farklı veri türleri arasında istatistiki norm dışı dağılım sebebiyle normalizasyon ihtiyacı olduğu gözlemlenmiştir. Veri [0-1] değer aralığına normalize edilmiştir. Normalize edilmiş veri, modellerin eğitiminde ve testinde kullanılmıştır.

3.3. Önerilen Yaklaşım

Bu bölümde anlatılan çalışma olası veri kayıplarını engellemek adına yapılmaktadır. Önerilen yaklaşım bir veri türündeki akışta eksik veri durumu oluştuğunda, veri kaybı zaman dilimindeki diğer veri türlerini kullanarak eksik verinin tahmin edilmesidir. Bu amaçla biri DL, üçü ANFIS tabanlı olmak üzere dört eksik veri tamamlama metodu önerilmektedir. Önerilen modellerin test edileceği üç eksik veri senaryosu tanımlanmıştır ve Çizelge 3.3’te tanımlanmıştır.

Çizelge 3.3. Tanımlanan eksik veri senaryoları

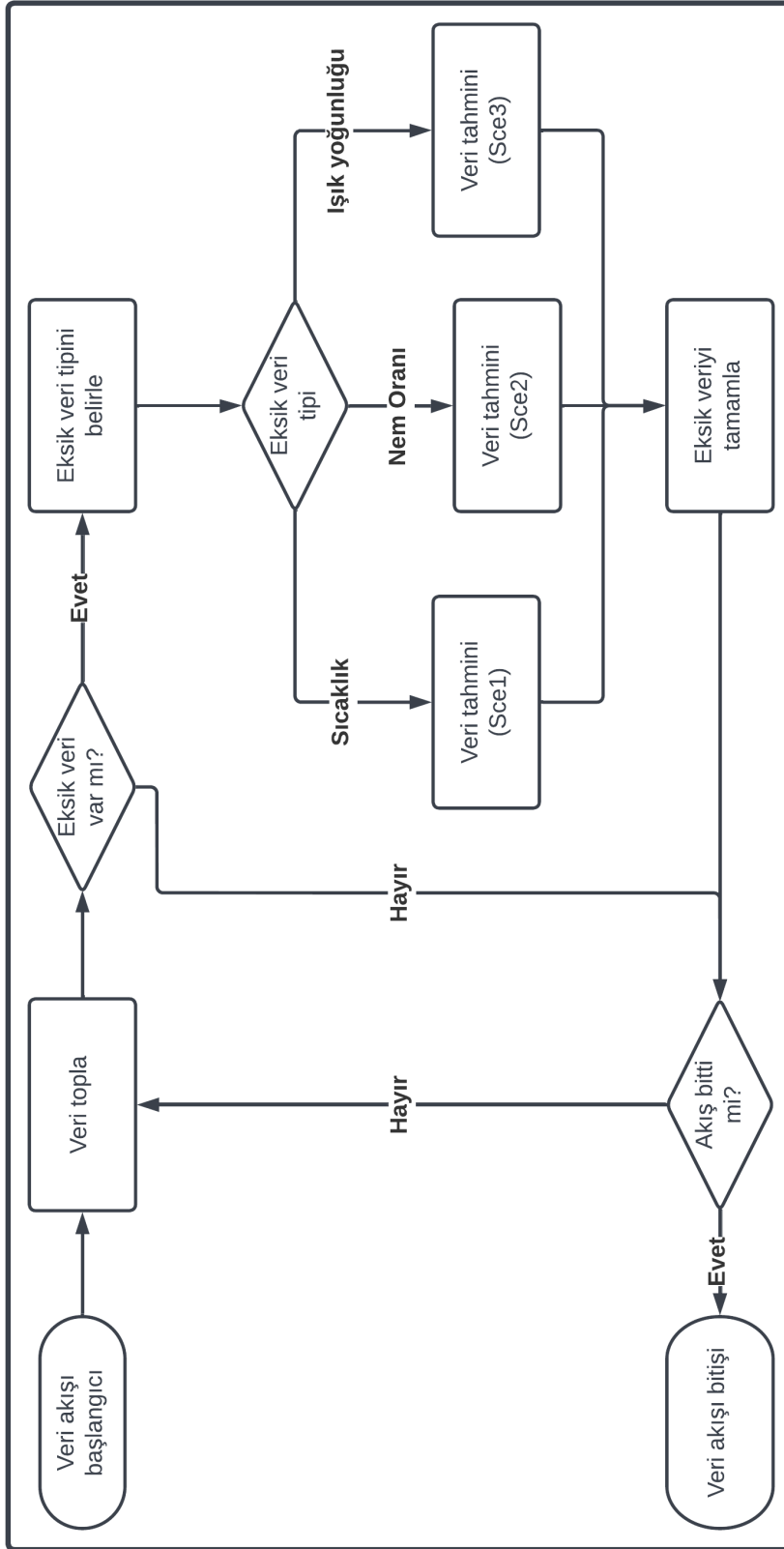
Senaryo Kısaltması	Giriş I	Giriş II	Çıkış (Eksik Veri Tipi)
Sce1	Nem Oranı	Işık Yoğunluğu	Sıcaklık
Sce2	Sıcaklık	Işık Yoğunluğu	Nem Oranı
Sce3	Sıcaklık	Nem Oranı	Işık Yoğunluğu

Eksik veri oluşması durumunda Çizelge 3.3’de verilen senaryolardan çıkış veri tipi uygun senaryo seçilerek eksik veri tamamlanır. Senaryo seçimine dair detay aşağıda verilmiştir.

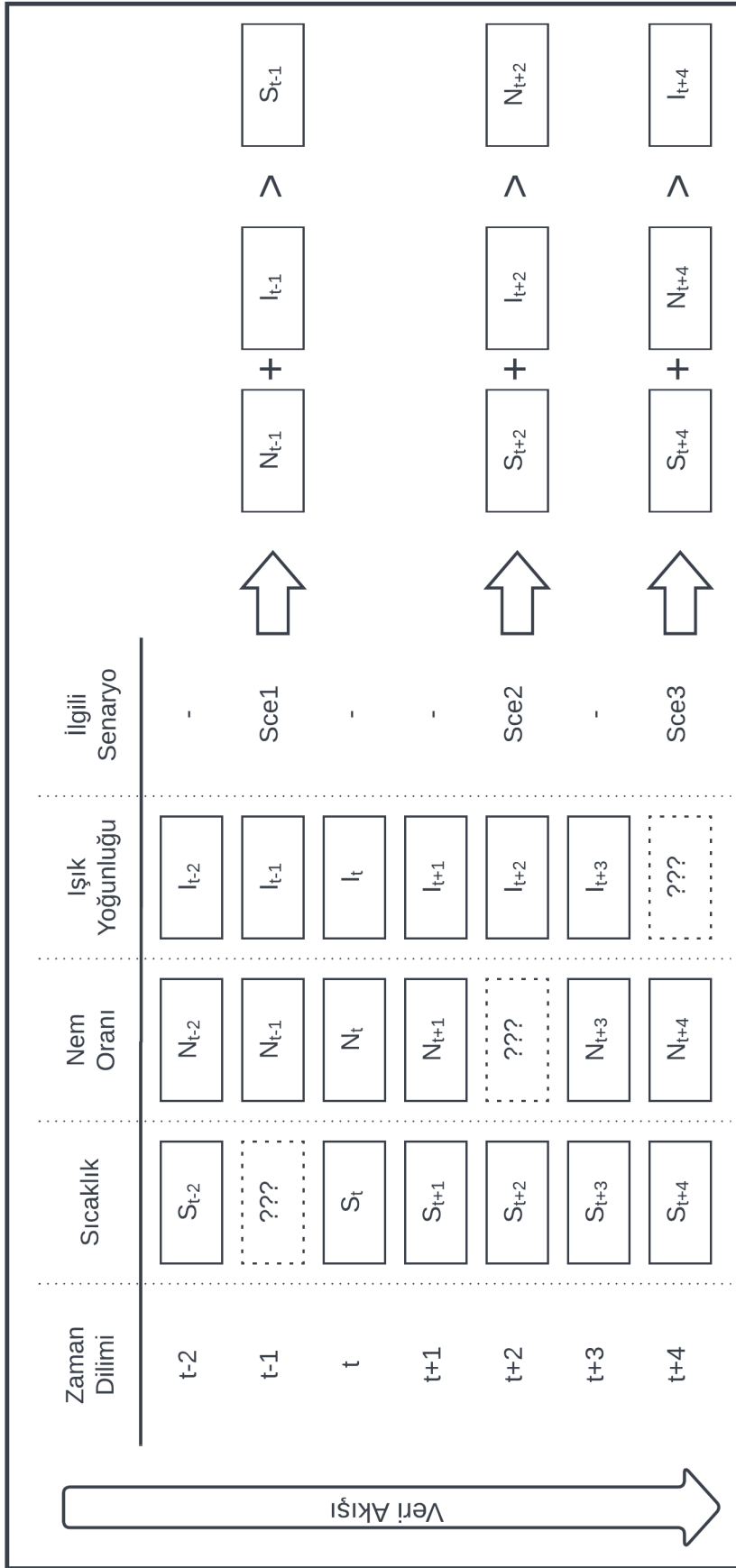
- Eksik veri tipi sıcaklık ise Sce1 kullanılır.
- Eksik veri tipi nem oranı ise Sce2 kullanılır.
- Eksik veri tipi ışık yoğunluğu ise Sce3 kullanılır.

Önerilen yaklaşımın daha iyi anlaşılmasına adına bir akış diyagramı Şekil 3.3’te verilmiştir. Bir zaman penceresinde eksik veri tespit edildiğinde akış diyagramında verilen adımlar

izlenir. Şekil 3.4'te ise önerilen yöntemin ve senaryo seçimine dair örnek bir veri akışı verilmiştir.



Şekil 3.3. Önerilen eksik veri tamamlama yaklaşımına dair akış şeması



Şekil 3.4. Örnek veri akışı ve eksik veri senaryolarının kullanılması

Şekil 3.4’de verilen zaman dilimi bir veri akışının $t-2$ ila $t+4$ zaman dilimleri arasındaki aralığını vermektedir. I_x , N_x ve S_x , x anında sırasıyla ışık yoğunluğu, nem oranı ve sıcaklık türündeki algılayıcı okumalarını ifade etmektedir. Görüldüğü üzere eksik veri durumları $t-1$, $t+2$, $t+4$ zamanlarında oluşmaktadır. Eksik veri ile karşılaşıldığında o zaman diliminde elde olan diğer algılayıcı tiplerinden eksik veri tahmin edilmektedir.

3.3.1. ANFIS temelli eksik veri tamamlama yaklaşımları

Bahsedildiği üzere önerilen eksik veri tamamlama yaklaşımlarının üçü ANFIS temeline sahiptir. Bu bölümde ANFIS algoritmasından bahsedilecektir. Ardından eksik veri tamamlama için geliştirilen üç ANFIS temelli modelden bahsedilecektir.

ANFIS

Matematiksel ve istatistiksel temele dayanan tahmin modelleri matematiksel olarak formüle edilebilen problemlerde yüksek başarıya sahiptir. Ancak, gerçek dünya problemleri kesinlikten uzak olduğu için bu kategori içinde değildir [18]. Gerçek dünya problemlerinde sıkça kullanılan yaklaşımlardan biri FIS modelleridir. FIS modelleri bulanık kurallar uyarınca giriş uzayından çıkış uzayına lineer olmayan bir ilişkilendirme yapar. Bulanık kurallar ise alan uzmanı tarafından yazılır. Dolayısıyla bir FIS modelinin başarısı kuralları yazan uzmanın uzmanlık seviyesine oldukça bağlıdır. Literatürde iki tür FIS sıkça kullanılır, bunlar Mamdani [211] ve Sugeno [212] tiplerindeki FIS’lerdir.

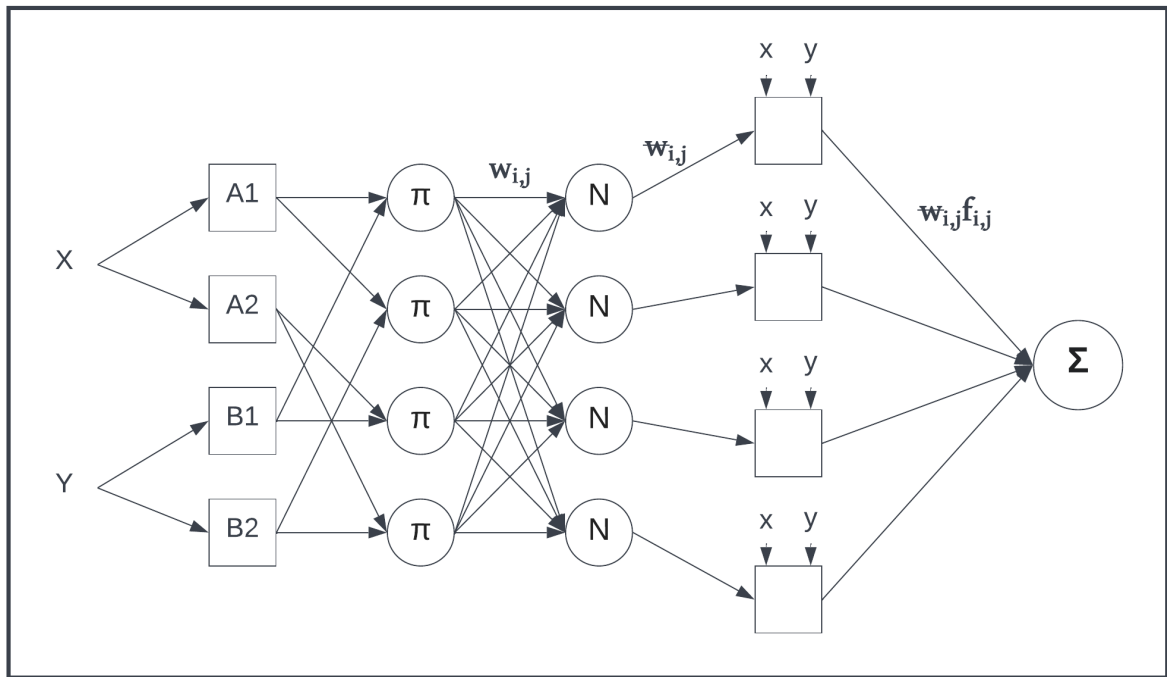
Bazı durumlarda alan uzmanı bulmak veya kuralları oluşturabilmek zor olabilmektedir. Ek olarak, kurallar bir şekilde elde edilse bile üyelik fonksiyonlarının parametrelerinin optimize edilmesi bir gerekliliktir. ANFIS [183] bu eksikleri ve zorlukları aşmak amacıyla Jang tarafından önerilmiştir. Giriş-Çıkış şeklinde ilişkilendirilmiş veriyi alan ANFIS sistemleri insan faktörünü aradan çıkararak kuralları ve parametrelerin optimizasyonunu kendi başına yapabilmektedir.

Takagi-Sugeno’nun metodu iyi bir ilişkilendirme metodu olarak tanımlanmaktadır [213]. İlişkilendirme metodu her eğer-yoksa (if-else) durumu için bir kural oluşturur. Sugeno tipi ilişkilendirme ile oluşturulan her bulanık kural tek bir çıkışa sahiptir ve bu çıkış girişlerin ve bir değişkenin lineer kombinasyonudur. Sistem çıkışı ise kural çıkışlarının ağırlıklı toplanması ile elde edilir [18].

Daha iyi anlaşılması adına iki girişe (x, y) sahip bir ANFIS sistemini ele alalım. Her girişin iki bulanık terime sahip olabileceğini varsayalım. Bahsedilen senaryoda aşağıda verilen kurallardan oluşan bir FIS elimizde olacaktır.

- Kural 1: IF (x = A₁) AND (y = B₁) THEN f₁₁ = p₁₁x + q₁₁y + r₁₁
- Kural 2: IF (x = A₁) AND (y = B₂) THEN f₁₂ = p₁₂x + q₁₂y + r₁₂
- Kural 3: IF (x = A₂) AND (y = B₁) THEN f₂₁ = p₂₁x + q₂₁y + r₂₁
- Kural 4: IF (x = A₂) AND (y = B₂) THEN f₂₂ = p₂₂x + q₂₂y + r₂₂

Verilen örnek kurallarda p_{ij}, q_{ij} ve r_{ij} ANFIS sisteminin eğitim sürecinde optimize edilen parametreler, A₁ ve A₂ x'i tanımlayabilecek bulanık terimler, B₁ ve B₂ y'i tanımlayabilecek bulanık terimler ve f_i i'inci kuralın çıkışıdır. Şekil 3.5'te verilen örneğin 5 katmanlı ANFIS yapısında ifadesi verilmiştir.



Şekil 3.5. 5 katmanlı örnek ANFIS yapısı

Resimde daire şekline sahip düğümler sabittir ve eğitim sürecinde değişmez. Kare şeklindeki düğümler ise eğitim sürecinde kalibre edilen düğümlerdir, uyarlanabilir karakterdedir. Şekil 3.5'te görüldüğü üzere ANFIS 5 katmanlı bir mimariye sahiptir.

Katman 1: Birinci katmandaki her düğüm uyarlanabilir yapıdadır ve sisteme giriş olarak verilen değerler için üyelik seviyesini belirler. İki girişli bir modelde her giriş için üyelik fonksiyonu hesaplamaları aşağıda verilmiştir.

$$O_{1,i} = \mu_{A_i}(x), \quad i = 1,2 \quad (3.1)$$

$$O_{1,j} = \mu_{A_j}(y), \quad j = 1,2 \quad (3.2)$$

Yukarıda verilen μ_{A_i} ve μ_{A_j} üyelik fonksiyonlarıdır. Bu fonksiyonlar Gauss, çan üyelik fonksiyonları veya başka fonksiyonlar olabilir. Her üyelik fonksiyonunun kendine has parametreleri olabilmektedir. Bu parametrelere öncül parametreler denir.

Katman 2: Bu katmandaki her düğüm sabittir ve Π sembolü ile tanımlanır. Bu katmandaki düğümler gelen sinyalleri çarpar ve bir sonraki katmana gönderir. Gönderilen çıkışa atış gücü (firing strength) denir. Atış gücü aşağıdaki gibi hesaplanır.

$$O_{2,i,j} = w_{i,j} = \mu_{A_i}(x) \times \mu_{B_j}(y), \quad i,j = 1,2 \quad (3.3)$$

Katman 3: Bu katmandaki tüm düğümler sabittir ve N sembolü ile etiketlenir. N etiketli düğümler girişleri normalize eder. Her i 'inci N düğümü, i 'inci kuralın atış gücünü toplam atış gücüne bölerek normalize eder. Kullanılan normalizasyon fonksiyonu aşağıda verilmiştir.

$$O_{3,i,j} = \bar{w}_{i,j} = w_{i,j} / \sum w_{i,j}, \quad i, j = 1, 2 \quad (3.4)$$

Katman 4: Bu katmandaki tüm düğümler uyarlanabilir karakterdedir ve aşağıda verilen fonksiyona göre çıkış üretirler.

$$O_{4,i,j} = \bar{w}_{i,j} f_{i,j} = \bar{w}_{i,j}(p_{ij}x + q_{ij}y + r_{ij}), \quad i, j = 1, 2 \quad (3.5)$$

Bu katmandaki parametreler olan $p_{ij}x$, $q_{ij}y$ ve r_{ij} *ardıl parametreler* olarak adlandırılır.

Katman 5: Son katman olan beşinci katman tek düğümden oluşur ve dördüncü katmandan gelen tüm sinyalleri toplar. Σ sembolü ile etiketlenen düğüm aşağıda verilen fonksiyona göre çıkış üretir.

$$\text{Çıkış} = O_5 = \sum_i \sum_j w_{ij} f_{i,j}, \quad i, j = 1, 2 \quad (3.6)$$

Bahsedildiği üzere ANFIS 5 katmandan oluşmaktadır. Bunlardan birinci ve dördüncü katmanlar uyarlanabilir düğümlerden oluşur. ANFIS sisteminin öğrenmesi birinci katmandaki öncül ve dördüncü katmandaki ardıl parametrelerin kalibre edilmesinden gelir. Eğitim sürecinde öncül ve ardıl parametrelerin kalibre edilmesi için melez algoritma (Hybrid Algorithm) kullanılır [183]. Melez algoritma iki faza sahiptir. İleri yönlü ilerleyen faz ardıl parametreleri kalibre ederken, geri yönlü faz öncül parametreleri kalibre eder. İleri yönlü fazda öncül parametreler sabitlenir ve sinyal dördüncü katmana kadar iletilir. Dördüncü katmanda ardıl parametreler en küçük kareler metodu (Least Squares Method) ile kararlaştırılır. Geri yönlü fazda ardıl parametreler sabitlenir, hata oranları birinci katmana iletilir ve öncül parametreler dereceli alçalma metodu (Gradient Descent) ile hesaplanır [18].

ANFIS Temelli Eksik Veri Tamamlama Modelleri

Bu çalışma kapsamında eksik veri tamamlanması için ANFIS temelli üç model önerilmiştir. Bu modelleri ayıran özellik kullandıkları kümeleme yaklaşımıdır. Bahsedildiği üzere Sugeno tipi ANFIS bulanık eğer-yoksa kurallarına dayanan bir ilişkilendirme mekanizmasına sahiptir. FIS temelli sistemlerde kuralları tanımlamak için kullanılan bulanık terimler uzman tarafından kararlaştırılmaktadır. Ancak ANFIS yaklaşımı uzman gerektirmez, bulanık terimlerin kararlaştırılması için kümeleme kullanılır. Girişler ve çıkışlar kümeleme algoritmaları ile kümelere bölünür ve bu kümeler bulanık terimler olarak kullanılır. Ardından kümeleme sonucu elde edilen bulanık terimler üzerinden kurallar oluşturulur. Literatürde çok sayıda kümeleme yaklaşımı bulunmaktadır. Bu çalışmada ızgara bölümlenme (Grid Partitioning / GP), eksiltici kümeleme (Subtractive Clustering / SCL) ve FCM algoritmalarını kullanan üç ANFIS temelli metot önerilmiştir.

Önerilen metotların gerçek potansiyelinin ortaya çıkarılması için, ilgili metodun kullandığı kümeleme algoritmasının optimize edilmesi önem taşımaktadır. Bu sebeple, deneysel uygulamaların ilk seti kümeleme algoritmalarının optimizasyonuna dairdir. Farklı kümeleme algoritmasını kullanan her ANFIS temelli model ayrı ayrı optimize edilmiştir. Optimizasyon sürecinde izlenen metodoloji basittir. Optimize edilen algoritmanın her parametresi ayrı ayrı optimize edilmiştir. Bir parametre optimize edilirken, optimize edilen parametre dışındaki tüm diğer parametreler varsayılan değere sabitlenmiştir. Optimize edilen parametreye farklı

değerler verilerek en iyi sonuç veren değer elde edilmiştir. Her parametre için en optimize değerler birleştirilip kullanılmıştır.

GP-ANFIS

Önerilen ilk model ızgara bölümlleme kümeleme algoritması kullanan GP-ANFIS'tir. Zadeh'in önerdiği bulanık setler [214] ve bulanık terimlere [215-217] dayanan GP, FIS sistemlerinde sıklıkla kullanılmaktadır. Bu yöntemde giriş-çıkış eşleştirmeleri bir yüzey oluşturur ve yüzey girişlerin bulanık etiketine göre bölünür. Her ızgara bir bulanık ilişki alanını ifade eder ve bir kural olarak tanımlanır. Sayısal değerler de kümelenip bulanık terimler olarak kullanılabilir [218]. Bu esneklik GP algoritmasının uzman bilgisine ihtiyaç olmadan kullanılmasını mümkün kılmaktadır. GP'nin zayıf yanı giriş sayısıdır. Kural sayısı, giriş sayısı arttıkça üstel bir şekilde artar ve çok yüksek sayıda kural oluşur. Ayrıca, GP'nin randımanlı çalışması için yüksek sayıda eğitim verisi gereklidir. GP, giriş sayısı az ve eğitim seti büyük olduğu durumlarda kullanılmalıdır.

GP algoritmasının üç parametresi bulunmaktadır. Bunlar aşağıda verilmiştir.

- Giriş Üyelik Fonksiyon Tipi (Input Membership Function Type / IMFT) bulanık sistemlerde oldukça önemlidir. Üyelik fonksiyonu bir gözlemin bir bulanık terime hangi seviyede üye olduğunu [0, 1] aralığında bir çıktı verecek şekilde hesaplar ve sonuca direkt etki eder. Bu çalışma kapsamında MatLab'da tanımlanmış olan sekiz fonksiyon (bell-shaped mf, gaussian curve mf, gaussian combination mf, triangular-shaped mf, trapezoid-shaped mf, difference between two sigmoidal mf, product of two sigmoidal ve pi-shaped mf) denenmiştir. IMFT her giriş için ayrı tanımlanmaktadır.
- Üyelik Fonksiyonu Sayısı (Number of Membership Functions / NMF) her giriş için kaç tane üyelik fonksiyonu kullanılacağını ifade eder ve ortaya çıkacak kural sayısını direkt etkiler. Her giriş için ayrı ayrı tanımlanır. Bu çalışma kapsamında [2, 10] aralığındaki değerler denenmiştir.
- Çıkış Üyelik Fonksiyon Tipi (Output Membership Function Type / OMFT) çıkış için üyelik fonksiyon tipini belirler. İki değer alabilir, bunlar sabit ve lineerdir.

Yukarıda verilen değer aralıklarında yapılan parametre optimizasyonu sonucunda elde edilen GP-ANFIS modelinin en etkin parametre konfigürasyonları Çizelge 3.4'te verilmiştir.

Çizelge 3.4. GP-ANFIS modelinin en etkin parametre konfigürasyonu

Senaryo	IMFT (Giriş I)	IMFT (Giriş II)	NMF (Giriş I)	NMF (Giriş II)	OMFT
Sce1	gaussmf	trimf	2	3	linear
Sce2	trimf	trapmf	2	2	constant
Sce3	gaussmf	gaussmf	2	2	constant

SCL-ANFIS

Önerilen ikinci metot SCL kümeleme algoritmasını kullanan SCL-ANFIS'tir. SCL her veri noktasını potansiyel bir küme merkezi olarak ele alır ve veri noktası yoğunluğuna dayanan bir olasılık çıkarır. Hali hazırda bir kümede yer alan veri noktaları atılır ve geri kalan noktalardan yeni küme merkezleri elde edilir. GP'nin aksine SCL yüksek sayıda giriş durumunda problem yaşamaz çünkü oluşacak kural sayısı SC sonucunda oluşacak küme sayısına bağlıdır [219]. Ancak, yüksek sayıda gözlemden oluşan bir eğitim seti SCL'nin işlem karmaşıklığını yükseltip çalışma zamanına zarar verebilmektedir [18]. Ek olarak, küme yarıçapının belirlenmesi SCL'nin performansına direkt etki etmektedir. Gereğinden küçük kümeler kural sayısını artırırken, gereğinden büyük kümeler çok genel kurallar oluşmasına sebep olur [220].

SCL algoritmasının dört parametresi bulunmaktadır.

- Küme Etki Alanı (Cluster Influence Range / CIR) kümelerin etki alanını gösterir. Varsayılan değeri 0.5'tir. Bu çalışma kapsamında [0.1, 0.9] aralığındaki değerler test edilmiştir.
- Ölçekleme Faktörü (Scaling Factor / SF) etki alanını ölçeklenme faktörüdür. Varsayılan değer 1.25'tir. Bu çalışma kapsamında [0.3, 1.5] arasındaki değerler test edilmiştir.
- Kabul Oranı (Accept Ratio / AR) yeni kümelerin kabulü için kullanılan bir parametredir. [0.3, 0.95] aralığındaki değerler test edilmiştir ama sonuca bir etkisi gözlemlenmemiştir.
- Ret Oranı (Reject Ratio / RR) yeni kümelerin reddi için kullanılmaktadır. [0.05, 0.3] aralığındaki değerler test edilmiştir.

Yukarıda verilen değer aralıklarında yapılan parametre optimizasyonu sonucunda elde edilen SCL-ANFIS için en etkin parametre konfigürasyonları Çizelge 3.5'te verilmiştir. AR'nin sonuca bir etkisi gözlemlenmediği için varsayılan değer olan 0.5 kullanılmıştır.

Çizelge 3.5. SCL-ANFIS modelinin en etkin parametre konfigürasyonu

Senaryo	CIR (Giriş I)	CIR (Giriş II)	CIR (Çıkış)	SF	AR	RR
Sce1	0.9	0.3	0.3	0.9	0.5	0.25
Sce2	0.7	0.3	0.7	0.9	0.5	0.2
Sce3	0.7	0.6	0.7	0.9	0.5	0.3

FCM-ANFIS

Önerilen son model kümeleme için FCM algoritmasını kullanan FCM-ANFIS'tir. Dunn tarafından önerilen [221, 222] ve Bezdek tarafından ilerletilen [223] FCM her veri noktasının farklı üyelik dereceleri ile birden çok kümeye ait olabildiği bir kümeleme metodudur [224]. FCM kümeler arasındaki keskin sınırları kaldırır [224]. Gözden kaçırılmamalıdır ki FCM parametre olarak oluşacak küme sayısına ihtiyaç duymaktadır. Bu çalışma kapsamında kullanılan MatLab'ın FCM fonksiyonu küme sayısını FCM öncesi SCL uygulayarak elde etmektedir. Ancak, çalışma kapsamında küme sayısı bir optimizasyon parametresi olarak tanımlanıp sabit olarak verilmiştir.

FCM algoritmasının dört parametresi bulunmaktadır.

- Küme Sayısı (Number of Clusters / CN) direkt olarak kural sayısını etkileyen bir parametredir. Varsayılan değeri olmayan bu parametreye FCM'in MatLab uyarlamasında CIR parametresi 0.5 olan SCL algoritması ile karar verilmektedir. Bu çalışma kapsamında SCL ile karar verilen küme sayısına ek olarak [2, 50] aralığındaki küme sayıları denenmiştir.
- Üs (Exponent / Expo) bulanık kümelerin kesişme oranına dair bir parametredir. Varsayılan değeri 2.0'dır. Bu çalışma kapsamında [1.2, 3.0] aralığındaki değerler test edilmiştir.

- Maksimum İterasyon Sayısı (Maximum Number of Iterations / MNI) eğitim sürecinin sonlandırılmasına dair bir parametredir ve algoritmanın de en fazla kaç iterasyon yapılacağını belirtir. [5, 50] aralığındaki değerler test edilmiştir.
- Minimum Gelişme Değeri (Minimum Improvement Value / MIV) eğitim sürecinin sonlandırılmasına dair bir parametredir. Varsayılan değer 1.00E-5'tir. Bu çalışma kapsamında 1.00E-4, 1.00E-5 ve 1.00E-6 değerleri test edilmiştir.

Yukarıda verilen değer aralıklarında yapılan parametre optimizasyonu sonucunda elde edilen FCM-ANFIS için en etkin parametre konfigürasyonları Çizelge 3.6'da verilmiştir.

Çizelge 3.6. FCM-ANFIS modelinin en etkin parametre konfigürasyonu

Senaryo	CN	Expo	MNI	MI
Sce1	3	1.2	15	1.00E-5
Sce2	2	1.2	50	1.00E-5
Sce3	3	1.2	10	1.00E-5

3.3.1. DL temelli eksik veri tamamlama yaklaşımı

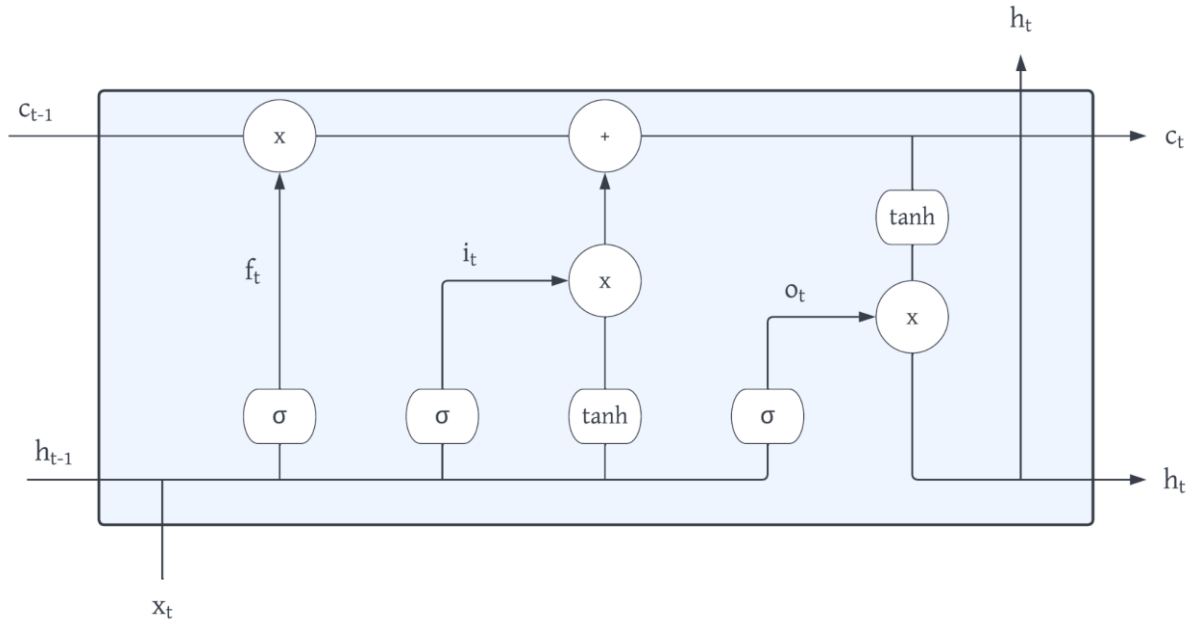
Eksik veri tamamlanması için önerilen dördüncü model DL temellidir. Bu bölümde DL, modelde kullanılan DL varyasyonu olan LSTM ve önerilen metodun parametre optimizasyon sürecinden bahsedilecektir.

LSTM mimarisi tekrarlı sinir ağları (RNN) mimarisinin bir uzantısıdır. Çeşitli eğitim problemlerinin aşılması için önerilen LSTM [225], sıralı ve zaman akışı şeklindeki verilerde yüksek başarıda öngörü ve tahmin yapabilmektedir [225]. LSTM mimarisinin gizli katmanında bellek blokları bulunur. LSTM bloğu birer giriş, unutma ve çıkış kapılarına sahiptir. Bu kapılar aracılığı ile veri blok içerisinde girişten çıkışa doğru ilerletilir [226, 227]. LSTM temelli bölümün anlatımında kullanılan semboller ve sembolize ettikleri Çizelge 3.7'te verilmiştir.

Çizelge 3.7. LSTM sembol tablosu

Sembol	Açıklama
x_t	t zamanındaki giriş vektörü
h_t	t zamanındaki çıkış vektörü
C_t	eski hücre durumu
C_t	yeni hücre durumu
i_t	giriş kapısı
f_t	unutma kapısı
o_t	çıkış kapısı
W_i	giriş kapısı ağırlık matrisi
W_c	hücre ağırlık matrisi
W_f	unutma kapısı ağırlık matrisi
W_o	çıkış kapısı ağırlık matrisi
*	eleman bazlı çarpım (product) işlemi, aynı boyuttaki iki vektörü çarpar
b_i	giriş kapısı eğilim vektörü
b_c	hücre eğilim vektörü
b_f	unutma kapısı eğilim vektörü
b_o	çıkış kapısı eğilim vektörü
σ	mantıksal sigmoid fonksiyonu
$\tanh$	hiperbolik tanjant fonksiyonu

Bir LSTM bloğunun görsel temsili Şekil 3.6’da verilmiştir.



Şekil 3.6. LSTM bloğu görsel temsili

LSTM bloğuna giren bir girişin çıkışa ulaşmasına kadar geçen ilerletme sürecinde uygulanan denklemler aşağıda verilmiştir.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.7)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.7)$$

$$\epsilon_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3.8)$$

$$C_t = f_t * C_{t-1} + i_t * \epsilon_t \quad (3.9)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.10)$$

$$h_t = o_t * \tanh(C_t) \quad (3.11)$$

Eksik veri tamamlanması için LSTM tipinde ara katmana sahip bir DL yaklaşımı kullanılmıştır. Üç eksik veri tipine dair üç farklı senaryo (Sce1, Sce2, Sce3) bulunmaktadır. Bu senaryolarda model giriş çıkış veri türleri farklı olduğu için üç senaryoya ait model için de ayrı ayrı bir parametre optimizasyonu gereklidir. Bu sebeple üç model için de aşağıda verilen değer aralıkları kullanılarak parametreler optimizasyonu ızgara arama (grid search) metodu ile yapılmıştır.

- Katman sayısı için 1, 2, 3 değerleri test edilmiştir.
- Gizli ünite sayısı için 50, 120, 240, 480 değerleri test edilmiştir.
- Yığın büyüklüğü için 60, 72, 240, 480, 1440 değerleri test edilmiştir.

Uygulanan testler sonucu Çizelge 3.8’de verilen değerlerin en iyi sonuçları verdiği gözlenmiştir.

Çizelge 3.8. DL modelinin en etkin parametre konfigürasyonu

Senaryo	Katman Sayısı	Gizli Ünite Sayısı (L1xL2)	Yığın büyüklüğü	Epoch
Sce1	2	240x20	1440	200
Sce2	2	240x240	1440	200
Sce3	2	60x60	60	200

3.4. Test Sonuçları

Önerilen modellerin başarımları RMSE metriğine göre ölçülmüştür. RMSE denklemi [18] tahmin edilen ve gerçek değer arasındaki hata oranına göre sonuç üretir. RMSE metriği oluşan hatanın karesini alarak kullanır ve bu sebepten ötürü minimum-maksimum değerleri farklı olan veri tiplerinde kullanılamaz. Bu çalışma kapsamında kullanılan değerler farklı minimum-maksimum değerlerine sahip olsa da, veriler normalize edildiği için RMSE kullanılabilmiştir.

Önerilen yöntemin başarısını ölçmek için literatürden lineer olmayan iki regresyon algoritması seçilmiştir. Bunlar destek vektör regresyon (Support Vector Regression / SVR) ve gauss çekirdek regresyon (Gaussian Kernel Regression / GKR) algoritmalarıdır. Çizelge 3.3’de verilen giriş çıkış kombinasyonları önerilen dört veri tamamlama metodu ve test için seçilen kontrol grubu algoritmalarında (SVR, GKR) kullanılmıştır. Testlerde her düğümün verisi 10-katlı çapraz doğrulama metodu uyarınca kullanılmıştır. Yani her düğüm için on test sonucu, üç düğüm olduğu göz önüne alındığında her model için toplamda 30 test sonucu elde edilmiştir. Test sonucu olarak verilen hata oranı mevzu bahis 30 sonucun ortalamasıdır. Dolayısıyla elde edilen sonuçların oldukça genel olduğu, veri seçiminden kaynaklanacak etkilerin minimize edildiği kabul edilmektedir. Buna tek istisna DL modelidir. DL modeli çalışmak için yüksek miktarda veriye ihtiyaç duyduğu için on üç düğümün verilerinin

birleşimini kullanmıştır. Eğitim için verinin yüzde 70'i, test için ise yüzde 30'u kullanılmıştır. ANFIS temelli yaklaşımlar, SVR ve GKR Matlab kullanılarak çalıştırılmıştır. DL modeli ise TensorFlow [228] ve Keras [229] ile geliştirilip çalıştırılmıştır. ANFIS temelli modellerde kullanılan parametreler Çizelge 3.4, Çizelge 3.5 ve Çizelge 3.6'da, DL temelli modelin kullanıldığı parametreler Çizelge 3.8'de verilmiştir. SVR ve GKR için Matlab'da tanımlanan varsayılan parametreler kullanılmıştır. Test sonuçları Çizelge 3.9'da verilmiştir.



Çizelge 3.9. ANFIS ve DL temelli eksik veri tamamlama modellerinin hata oranları

Senaryo	Model	Hata Oranı		Çalışma Süresi	
		Eğitim	Test	Eğitim	Test
Sce1	DL	0,0659	0,0701	24,28	0,5168
	GP-ANFIS	0,1487	0,0990	0,79	0,0086
	SCL-ANFIS	0,1581	0,0979	5,55	0,0081
	FCM-ANFIS	0,1662	0,1031	1,65	0,0067
	GKR	0,0907	0,1139	0,07	0,0007
	SVR	0,1068	0,1110	0,44	0,0014
Sce2	DL	0,0726	0,0880	26,63	0,5877
	GP-ANFIS	0,1109	0,1086	0,37	0,0074
	SCL-ANFIS	0,1040	0,1089	5,69	0,0065
	FCM-ANFIS	0,1106	0,1061	0,60	0,0055
	GKR	0,1094	0,1079	0,06	0,0006
	SVR	0,1140	0,1055	0,41	0,0015
Sce3	DL	0,1101	0,1711	320,82	0,6237
	GP-ANFIS	0,1096	0,1733	0,49	0,0063
	SCL-ANFIS	0,0905	0,1762	5,85	0,0064
	FCM-ANFIS	0,0943	0,1751	1,56	0,0062
	GKR	0,1690	0,1940	0,08	0,0007
	SVR	0,1797	0,1833	0,41	0,0012

Sce1 ve Sce3 senaryoları için, önerilen yöntemlerin test grubu algoritmalarına göre daha düşük hata oranına sahip olduğu, dolayısıyla daha yüksek başarıda tahmin yapabildiği görülmektedir. Önerilen yöntemler arasında ise DL temelli yöntemin en başarılı olduğu gözlemlenmektedir. Sce2 durumunda en iyi başarıya DL temelli yaklaşımın ulaştığı

görülmektedir. ANFIS temelli yaklaşımların ise kontrol grubu algoritmalarına göre önemli bir iyileştirme gösteremediği gözlemlenmiştir. Sonuç olarak üç eksik veri türü için de önerilen yöntemlerin kontrol grubu algoritmalarından daha başarılı olduğu görülmüştür.

Eksik veri senaryoları arasında, en kolay tahmin edilebilir senaryonun Sce1 olduğu gözlemlenmiştir. Bu bize sıcaklık-nem oranı ve sıcaklık-ışık yoğunluğu arasındaki yüksek korelasyonu göstermektedir. Benzer bir durum nem oranı verisinin eksik olduğu Sce2 senaryosu için de gözlemlenmektedir ancak sıcaklık-nem oranı ve ışık yoğunluğu-nem oranı arasındaki ilişkinin farklı karakterde olduğu görülmektedir. Önerilen algoritmalar Sce2 senaryosunda Sce1 senaryosuna göre daha yüksek hata oranına sahiptir. Buna zıt olarak kontrol grubu algoritmaları Sce2 senaryosunda Sce1 senaryosuna göre daha başarılıdır. Eksik veri senaryoları arasında ışık yoğunluğunun eksik olduğu Sce3 senaryosunda başarılar genel olarak düşüktür.

Sonuçları çalışma zamanı perspektifinden karşılaştırmak gerekirse önerilen yöntemlerin kontrol grubu algoritmaları ile karşılaştırıldığında geride kaldığı görülmüştür. GKR en düşük çalışma zamanına sahiptir. SVR ve GP-ANFIS temelli modellerin çalışma sürelerinin 3970 gözlemden oluşan bir eğitim seti için bir saniyenin altında olduğu görülmektedir. SCL ve FCM kümeleme kullanan ANFIS temelli modellerin eğitim aşamasında çalışma sürelerinin yüksek olduğu ancak bu yüksekliğin tahmin etme sürecine yansımadağı görülmüştür. ANFIS temelli modeller arasında en düşük sürede çalışan modelin GP-ANFIS olduğu gözlemlenmiştir. Sce1 ve Sce3 senaryolarında GP-ANFIS modelinin GKR ve SVR algoritmalarına göre daha düşük hata oranını kabul edilebilir bir sürede elde ettiği görülmüştür. DL temelli modellerin ise çalışma süreleri çok yüksektir. Özellikle yığın boyutu oldukça düşük olan Sce3 senaryosunda DL temelli modelin eğitimi 320.82 saniye sürmektedir.

Önerilen modellerin eksik verileri etkin bir şekilde tahmin edebildiği, IoT’de eksik veri problemi için kullanılabileceği gözlemlenmiştir. ANFIS temelli modeller düşük çalışma zamanlarında kabul edilebilir tahminler yapabilirken, en düşük hata oranına sahip tahminleri DL temelli model yapmaktadır.

3.5. Değerlendirme

IoT uygulamalarının etkin bir şekilde çalışmaları IoT ağdan toplanan veri kalitesine bağlıdır. Hatalı ve eksik veri IoT uygulamalarının çalışmasını engelleyebilir. Bu sebeple IoT ağlarda eksik ve tespit edilen hatalı verilerin tamamlanması gereklidir. QoS bakış açısından, bütünlüğünü bir ister olarak tanımlanmaktadır. IoT servis ve uygulamalarından belirli bir veri doğruluk ve doluluk oranı QoS isteri olarak talep edilebilmektedir. Bu noktalardan hareket ederek doktora çalışmalarının ilki eksik IoT verisinin tamamlanmasına dair yapılmıştır.

Çalışmanın ilk aşamasında literatür taraması yapılmıştır. Literatür taramasının ortaya çıkardığına göre bu alanda yapılan çalışmalar çoğunlukla sabit veri setlerinde yapılmaktadır ve konvansiyonel yaklaşımları (ortalama değer ile doldur, 0 ile doldur vb.) kullanmaktadır. Ancak bu yaklaşımlar IoT verisine uygun değildir. IoT verisi dinamik ve heterojen yapıdadır. Akan veri karakteristiğinde olduğu için ortalama bir değer ile doldurulamaz, çünkü başı sonu belli olmayan bir akışta ortalama hesaplanamaz. Ek olarak IoT verisinde konsept kayması durumu vardır, veri karakteristiği zaman ve koşullara göre değişebilir. Bu sebeple eksik verilerin tamamlanması adına konvansiyonel yöntemlere göre daha akıllı yaklaşımlar geliştirilmesi gerektiğine karar verilmiştir.

Bu çalışmada üçü ANFIS biri DL temeline sahip dört eksik veri tamamlama modeli geliştirilmiştir. Modeller Intel Berkeley Laboratuvarından toplanan gerçek veri üzerinde test edilmiştir. Veri üç veri tipinde toplanan algılayıcı verilerinden oluşmaktadır. Veri tipleri sıcaklık, nem oranı ve ışık yoğunluğudur. Üç veri tipi için de eksik veri durumu oluşması durumu senaryolaştırılmış ve önerilen dört model ile ayrı ayrı eksik veri tamamlama süreçleri yapılmıştır. Test sonuçları önerilen modellerin başarısını göstermektedir. ANFIS temelli modeller (GP-ANFIS, SCL-ANFIS ve FCM ANFIS) kısa çalışma zamanında kabul edilebilir sonuçlar verirken, DL temelli yaklaşımın yüksek çalışma zamanına sahip olmasına rağmen en düşük hata oranına sahip olduğu gözlemlenmiştir.

4. ÖNCELİKLENDİRME İLE FARKLI QOS SEVİYELERİNİN SAĞLANMASI

IoT uygulamaları tarafından talep edilen verinin toplandığı nesneden kullanılacağı konuma etkin bir şekilde ulaşması uygulamanın QoS seviyesine direkt etki eder. Farklı uygulamaların farklı gereksinimleri olabilir. Örnek vermek gerekirse, hava kirliliğini izlemek ve olası riskli durumları hava kirliliğine karşı hassas olan solunum yolu hastalıklarına sahip bireylere iletme olan A isimli bir uygulamayı ele alalım. Bu uygulamada nesnelerden toplanan hava kirliliği verisinin veri analizinin yapıldığı merkeze ulaşması sürecinin oldukça kısa olması gereklidir. Veri ne kadar kısa sürede iletilip işlenirse, hassas bireyler o kadar kısa sürede uyarılabilir. B isimli ikinci bir uygulama aynı ağı paylaşsın, hava durumunu anlık takip edip ileride kullanılmak üzere bir veri merkezine göndersin. Örnek olarak verilen senaryo incelendiğinde A uygulamasının zaman-kritik bir uygulama olduğu, bu uygulama için toplanan verinin gecikme minimize edilecek şekilde iletilmesi gerektiği görülmektedir. Uygulamanın etkin bir şekilde çalışması verinin minimal sürede konumuna ulaştırılmasına bağlıdır. B uygulaması ise gözlemlenen veriyi ileriki bir zamanda kullanılmak üzere yedeklemektedir. Dolayısıyla verinin erişim zamanı veya iletim sırasında kaybolması kritik değildir. A ve B uygulamalarının aynı ağı paylaştığı göz önüne alınırsa, bu uygulamalara ait veri paketlerinin ortak kaynakları kullanması dolayısıyla paylaşması beklenir. Bu durum ağ kaynakları kısıtlı olmadığı durumda bir problem yaratmaz. Ancak ağ kaynakları üzerindeki iş yükü yüksekse paket iletim süreleri yükselir. A ve B uygulamasına ait verilere eşit bir şekilde muamele edildiği bir durumda iki uygulamaya ait veriler benzer QoS seviyesi düşüşlerine maruz kalacaktır. Bu düşüş daha düşük QoS gereksinimine sahip B uygulaması için kabul edilebilir bir durum olsa dahi, A uygulaması için kritiktir. Örnekte verilen senaryo benzeri kaynak kısıtı olan ve uygulamaların QoS seviyeleri arasında fark olan durumlarda, uygulamalara ait verilere farklı muamele edilmesi gereklidir. Verilen örnekte A uygulamasına ait verilere daha fazla kaynak harcanması problemi çözecektir.

Bu bölümde anlatılan çalışmada, ağ kaynaklarının QoS seviyelerine göre IoT uygulamaları arasında paylaştırılmasına dair bir araştırma yapılmıştır. Araştırma hava kalitesi öngörü senaryosuna uygulanmıştır. Kullanılan veri Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı'nın Hava Kalitesi İzleme Ağı'nın [230] verileridir. Portal üzerinden alınan veri, ağdaki veri toplama istasyonlarının konumlarına uygun bir şekilde oluşturulan bir IoT ağda olması beklenecek bir veri trafiğine çevrilmiştir ve simüle edilmiştir.

4.1. Önerilen Yaklaşım

Önerilen yaklaşım FC ve SDN teknolojilerini kullanmaktadır. Bu bölümde bu teknolojilerden, bu çalışma kapsamında nasıl beraber kullanıldıklarından ve önerilen yaklaşımdan bahsedilecektir.

FC kavramı bulut sunucuların sağladığı işlevlerin (hesaplama, veri saklama ve iletişim kurma) bulut katmanı ile nesne katmanı arasına yerleştirilen, çok sayıda cihazdan oluşan ve sis katmanı olarak adlandırılan bir ara katmana taşınmasını önerir. Tamamlayıcı bir teknoloji olan FC uygulamaları ve servisleri son kullanıcıya ve nesne katmanına yaklaştırarak düşük gecikmeye ihtiyaç duyan ve konum-farkında uygulamaların QoS seviyelerini artırır [231, 232]. Bu yeteneğinden ötürü FC sanal gerçeklik, multimedya uygulamaları ve kritik veri akışına sahip uygulamalar için oldukça uygundur [80, 233]. Veri akışlarının sis sunucular üzerinden iletilmesi birçok avantaj sağlamaktadır. Mevzu bahis avantajlar [19, 234] aşağıda sıralanmıştır:

- FC oldukça heterojen yapıda olan IoT verisinde farklı karakterdeki veriler için farklı akış protokollerinin uygulanmasını sağlar.
- Sis sunucular üzerinde tanımlanacak veri ön işleme ve veri birleştirme servisleri ağ üzerindeki trafiği azaltır.
- Sis sunucularda veri içeriği, tipi ve QoS isterleri incelererek, QoS seviyesini yükseltecek müdahaleler yapılabilir.
- Veri işleme süreçleri sis sunuculara taşınarak daha hızlı sonuç alınması sağlanabilir.

FC kavramı ve sis sunucuları önceki bölümde bahsedildiği üzere birçok fayda sağlamaktadır. Ancak FC kavramının uygulanması ağdaki karmaşıklığı oldukça yükseltmektedir ve yönetilebilirlik konusunda ek ihtiyaçlar doğurmaktadır. Sis sunucular veri işleme işlemlerini gerçekleştirebilir, veri akışlarına dair kararlar verebilir ancak bu veri akış kararlarının uygulanmasına müdahale edemez. Sis sunucularda alınan kararların ağ katmanında iletme uygulanabilmesi için ek bir teknoloji gereklidir [19]. Bu çalışma kapsamında FC kavramına tamamlayıcı olarak da tanımlanabilecek bir teknoloji olan SDN kullanılmıştır. SDN sis sunucuları tarafından karar verilen veri önceliklerinin ağ katmanında uygulanmasını sağlayabilir. Ayrıca, sis sunucularda, nesnelerde veya ağ geçitlerinde olabilecek donanımsal

hatalarda ağ akış rotalarını değiştirip ağın işlevine devam etmesini sağlar. Genel olarak ağın yönetilebilirliğini artırır, ağ optimize eder ve hata kontrolü sağlar [235, 236].

Bu çalışma kapsamında, farklı QoS seviyelerine sahip paketlerin önceliklendirme QoS seviyelerine uygun yönlendirilmesi problemine odaklanılmıştır. Bu amaçla FC ve SDN teknolojileri birlikte kullanılmıştır. Önerilen yöntemin test edilmesi amacıyla hava kalitesi öngörü senaryosu oluşturulmuştur.

4.2. Veri Analizi ve Senaryo

Önerilen yöntemin test edilmesi için hava kalitesi öngörüsüne dair bir senaryo oluşturulmuştur. Senaryonun olabildiğince gerçek dünya uygulamalarına uygun olması istendiğinden gerçek dünya verisi kullanılmıştır.

4.2.1. Veri kaynağı

Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı'nın Hava Kalitesi İzleme Ağı [230] ülkeye yayılmış 339 hava kalitesi izleme istasyonuna sahiptir. Bu istasyonları en çok barındıran il ise Türkiye'nin en yüksek nüfusa sahip ili olan İstanbul'dur. Bu sebeple İstanbul ili senaryo konumu olarak seçilmiştir. Senaryo zamanı ise İstanbul'un tüm 2019 verilerini içermektedir. İstanbul ilinde bulunan istasyonlardan en yüksek veri doluluğuna sahip 29 istasyon seçilmiştir. Bakanlığın portalı kullanılarak seçilen istasyonlardan çeşitli kirleticilere dair veriler indirilmiş ve kullanılmak üzere bir veri seti oluşturulmuştur. Kirleticilerden aşağıda verilenler seçilmiştir:

- Nitrojen Dioksit (NO₂)
- Sülfür Dioksit (SO₂)
- Ozon (O₃)
- Karbon Monoksit (CO)
- 10 mikrometre çaplı parça madde (PM₁₀)

Bu kirleticilerin seçiminde Birleşik Krallık [237], Amerika Birleşik Devletleri (ABD) [238] ve Avrupa Birliği'nin hava kalitesi endeksleri [239] kullanılmıştır. PM₁₀, O₃ ve NO₂

kirleticileri bahsedilen üç endekste de kullanılmaktadır. SO₂ hem ABD'nin hem de Birleşik Krallık'ın endeksinde kullanılmaktayken CO sadece ABD'nin endeksinde kullanılmaktadır.

Veri ön işleme aşamasında bazı istasyonlarda veri doluluğunun spesifik kirleticilerde düşük olduğu gözlemlenmiştir. Bu sebeple doluluk oranı %90'ın üstünde olan kirletici verileri kullanılmıştır. Çizelge 4.1'de seçilen istasyonlar ve her istasyonda takip edilen kirletici türleri verilmiştir. Çizelgede görülen işaretlemelerden “X” o verinin ilgili istasyonda ölçüldüğünü ve çalışma kapsamında kullanıldığını, “D” ise verinin ölçüldüğünü ama doluluğu az olduğu için kullanılmadığını ifade etmektedir.



Çizelge 4.1. Test senaryosunda kullanılan istasyonlar

İstasyon No	İstasyon ID	İstasyon İsmi	İzlenen Kirleticiler				
			CO	NO ₂	O ₃	PM10	SO ₂
1	153787	Silivri MHMT		X	X	X	X
2	153791	Esenyurt MHMT		X	X	X	D
3	153799	Avcılar	X	X		X	X
4	153792	Basakşehir MHMT	X	X	X	X	X
5	153779	Yenibosna	X			X	X
6	153795	Şirinevler MHMT	X	X		X	X
7	153773	Esenler	D	X		X	X
8	153788	Sultangazi MHMT		X	X	X	X
9	153802	Kumköy	X	X	X	X	
10	153774	Üsküdar				X	X
11	153801	Selimiye	X	X	X	X	
12	153771	Kadıköy		X		X	X
13	153783	Göztepe			X	X	
14	153796	Üsküdar MHMT	X	X		X	
15	153800	Çatladıkapı	X	X	X	X	
16	153772	Aksaray	X	X		X	X
17	153778	Alibeyköy		X	X	X	X
18	153789	Kağıthane MHMT		X	X		
19	153782	Kağıthane	X			X	X
20	153794	Mecidiyeköy MHMT	X	X		X	
21	153780	Beşiktaş	X	X	X	X	X
22	153785	Maslak		X	X	X	
23	153776	Sarıyer	D				
24	153784	Kandilli				X	
25	153797	Kandilli MHMT	X	X		X	
26	153793	Ümraniye MHMT	X	X		X	X
27	153777	Ümraniye	X	X		X	X
28	153775	Kartal				X	X
29	153790	Sultanbeyli MHMT		X	X	X	X

4.2.2. IoT temelli mimari

İstanbul'daki istasyonların portaldan alınan görüntüsü Şekil 4.1'de verilmiştir.

ağ ile toplanan verilerin kullanılarak her t saatinde bir sonraki saate (t+1) dair hava kalitesini öngörmektir. IoT temelli mimari nesneler, sis sunucuları ve bir bulut sunucudan oluşur. En merkezi konumdaki istasyon bulut sunucusu olarak kararlaştırılmıştır ve Şekil 4.1’de sarı ile renklendirilmiştir. Kendi bölgelerinde merkez olan istasyonlar ise sis sunucu olarak seçilmiştir ve kırmızı renkli olarak gösterilmiştir. Kalan istasyonlar nesnedir ve beyaz renkte işaretlenmişlerdir. Seçilen nesneler ve görevleri Çizelge 4.2’de verilmiştir.

Çizelge 4.2. IoT temelli mimaride istasyonların görevleri

Rol	Seçilen İstasyon Numaraları
Nesne	1,3,4,5,6,8,9,10,12,13,14,15,16,17,18,19,20,22,24,25,26,28,29
Sis Sunucu	2,7,11,23,27
Bulut Sunucu	21

IoT tabanlı mimaride ağ elemanlarının görevleri aşağıda verildiği gibidir.

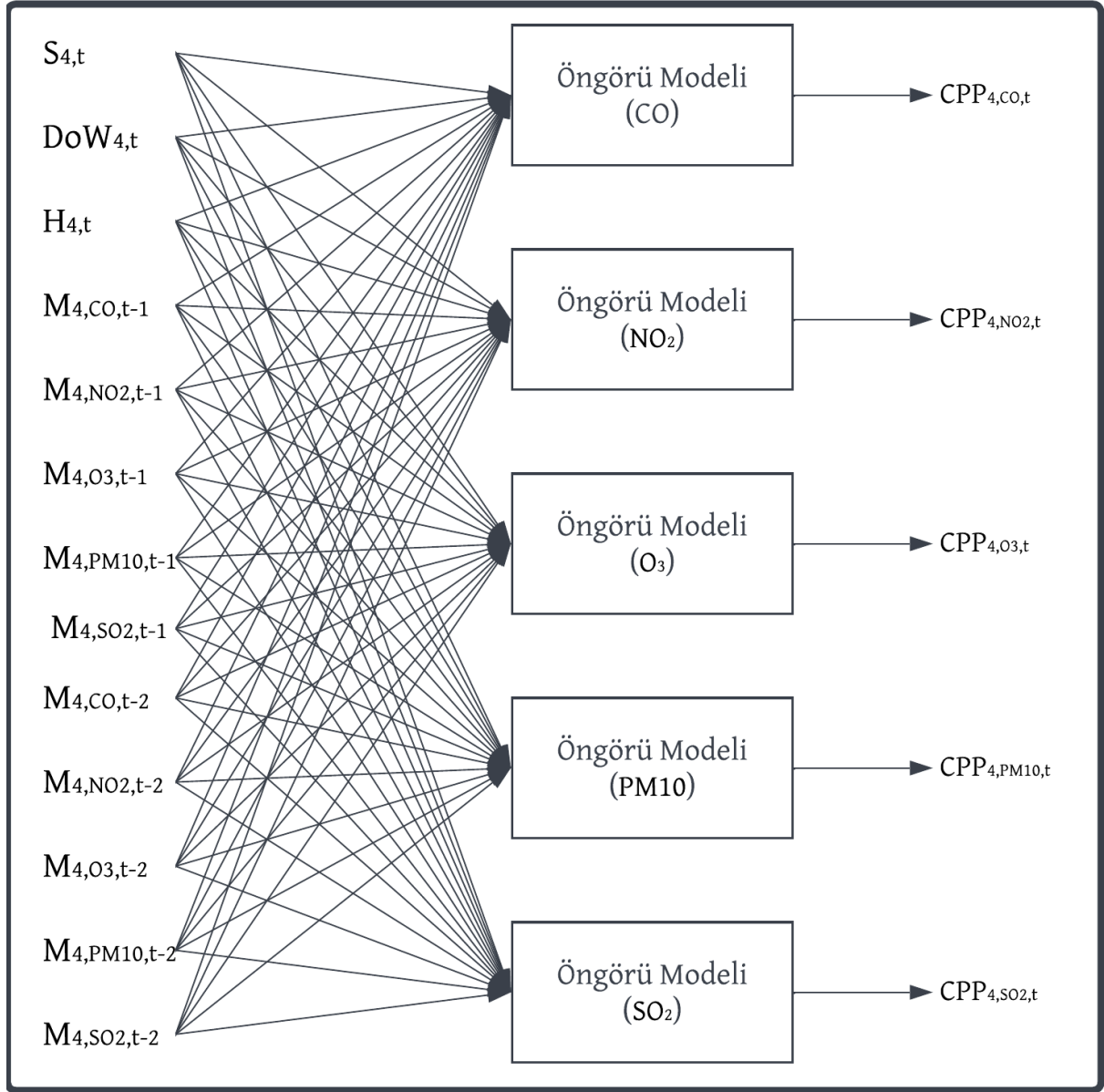
- Nesneler kendi alanlarında kirletici ölçümlerini yapar. Topladığı verileri sis sunucularına iletir. Kendi topladıkları verileri kullanarak kirleticiler arası öngörü (Cross-pollutant Prediction / CPP) karakterinde öngörü yapar. Öngörü sonuçlarını kirletici seviyesine göre önceliklendirerek bulut sunucuya gönderir. Toplam 23 tane nesne bulunmaktadır.
- IoT tabanlı mimaride iş yükünün çoğu sis katmanı üzerindedir. Bu katman bulut sunucu ile nesneleri bağlar. Veri akışları nesnelerden bulut sunucuya giderken sis sunucular üzerinden geçer. Sis sunucular kendi alanlarında kirletici ölçümleri yapar. Topladıkları veri ile CPP karakterinde öngörü yapar, sonuçları kirletici ölçümüne göre önceliklendirerek bulut sunucuya gönderir. Ayrıca, sis sunucular kendi sorumluluklarındaki kirleticinin mekânsal öngörü (Spatial Prediction / SP) karakteristiğindeki öngörüsünü yapar, sonuçları kirletici ölçümüne göre önceliklendirerek bulut sunucuya gönderir. Toplamda beş istasyon sis sunucu olarak tanımlanmıştır.
- Bulut sunucu ağ akışlarının ulaştığı son duraktır. Bulut sunucuda toplanan veriler analiz edilir. Yüksek hava kirliliği ön görülürse, hava kirliliğine karşı hassas olanlara olası sağlık problemlerinin önüne geçmek adına alarm gönderilir. Bulut sunucunun bu çalışma kapsamındaki görevi kendi bölgesindeki kirletici seviyelerini ölçüp ilgili sis düğümüne gönderme ve CPP karakterinde öngörü yapmaktır.

4.2.3. Öngörü modelleri

IoT temelli mimaride iki çeşit öngörü mekanizması tanımlanmıştır. Bu mekanizmalar SP ve CPP öngörü mekanizmalarıdır. CPP mekanizması nesnelerde, yerel olarak gözlenen son 2 saatin kirletici konsantrasyonlarını, mevsimi, haftanın gününü ve günün saatini kullanarak bir sonraki saate dair kirletici konsantrasyonlarını öngörür. Her kirletici için öngörü ayrı ayrı yapılır. Toplamda beş çeşit kirletici ölçen bir nesnede, her saat başı her kirletici için bir olmak üzere toplamda beş öngörü modeli çalıştırılır. Örnek olarak 4 numaralı nesnenin t'inci diliminde yaptığı öngörüler aşağıda sıralanmıştır. $CPP_{A,B,C}$ A nesnesinin B kirleticisine dair C'inci zaman dilimine dair yaptığı öngörü, $M_{D,E,F}$ D nesnesinin E kirleticisine dair F'inci zaman diliminde yaptığı ölçüm, $S_{G,H}$, $DoW_{G,H}$ ve $H_{G,H}$ sırasıyla G nesnesinin H'inci zaman dilimindeki mevsim, haftanın günü ve günün saati bilgisini, ÖNGÖR ise öngörü modelini ifade etmektedir.

- $CPP_{4,CO,t} = \text{ÖNGÖR}(S_{4,t}, DoW_{4,t}, H_{4,t}, M_{4,CO,t-1}, M_{4,NO_2,t-1}, M_{4,O_3,t-1}, M_{4,PM_{10},t-1}, M_{4,SO_2,t-1}, M_{4,CO,t-2}, M_{4,NO_2,t-2}, M_{4,O_3,t-2}, M_{4,PM_{10},t-2}, M_{4,SO_2,t-2})$
- $CPP_{4,NO_2,t} = \text{ÖNGÖR}(S_{4,t}, DoW_{4,t}, H_{4,t}, M_{4,CO,t-1}, M_{4,NO_2,t-1}, M_{4,O_3,t-1}, M_{4,PM_{10},t-1}, M_{4,SO_2,t-1}, M_{4,CO,t-2}, M_{4,NO_2,t-2}, M_{4,O_3,t-2}, M_{4,PM_{10},t-2}, M_{4,SO_2,t-2})$
- $CPP_{4,O_3,t} = \text{ÖNGÖR}(S_{4,t}, DoW_{4,t}, H_{4,t}, M_{4,CO,t-1}, M_{4,NO_2,t-1}, M_{4,O_3,t-1}, M_{4,PM_{10},t-1}, M_{4,SO_2,t-1}, M_{4,CO,t-2}, M_{4,NO_2,t-2}, M_{4,O_3,t-2}, M_{4,PM_{10},t-2}, M_{4,SO_2,t-2})$
- $CPP_{4,PM_{10},t} = \text{ÖNGÖR}(S_{4,t}, DoW_{4,t}, H_{4,t}, M_{4,CO,t-1}, M_{4,NO_2,t-1}, M_{4,O_3,t-1}, M_{4,PM_{10},t-1}, M_{4,SO_2,t-1}, M_{4,CO,t-2}, M_{4,NO_2,t-2}, M_{4,O_3,t-2}, M_{4,PM_{10},t-2}, M_{4,SO_2,t-2})$
- $CPP_{4,SO_2,t} = \text{ÖNGÖR}(S_{4,t}, DoW_{4,t}, H_{4,t}, M_{4,CO,t-1}, M_{4,NO_2,t-1}, M_{4,O_3,t-1}, M_{4,PM_{10},t-1}, M_{4,SO_2,t-1}, M_{4,CO,t-2}, M_{4,NO_2,t-2}, M_{4,O_3,t-2}, M_{4,PM_{10},t-2}, M_{4,SO_2,t-2})$

Yukarıda verilen öngörü fonksiyonları daha anlaşılabilir olması için aşağıda Şekil 4.3'de görsel olarak da verilmiştir.



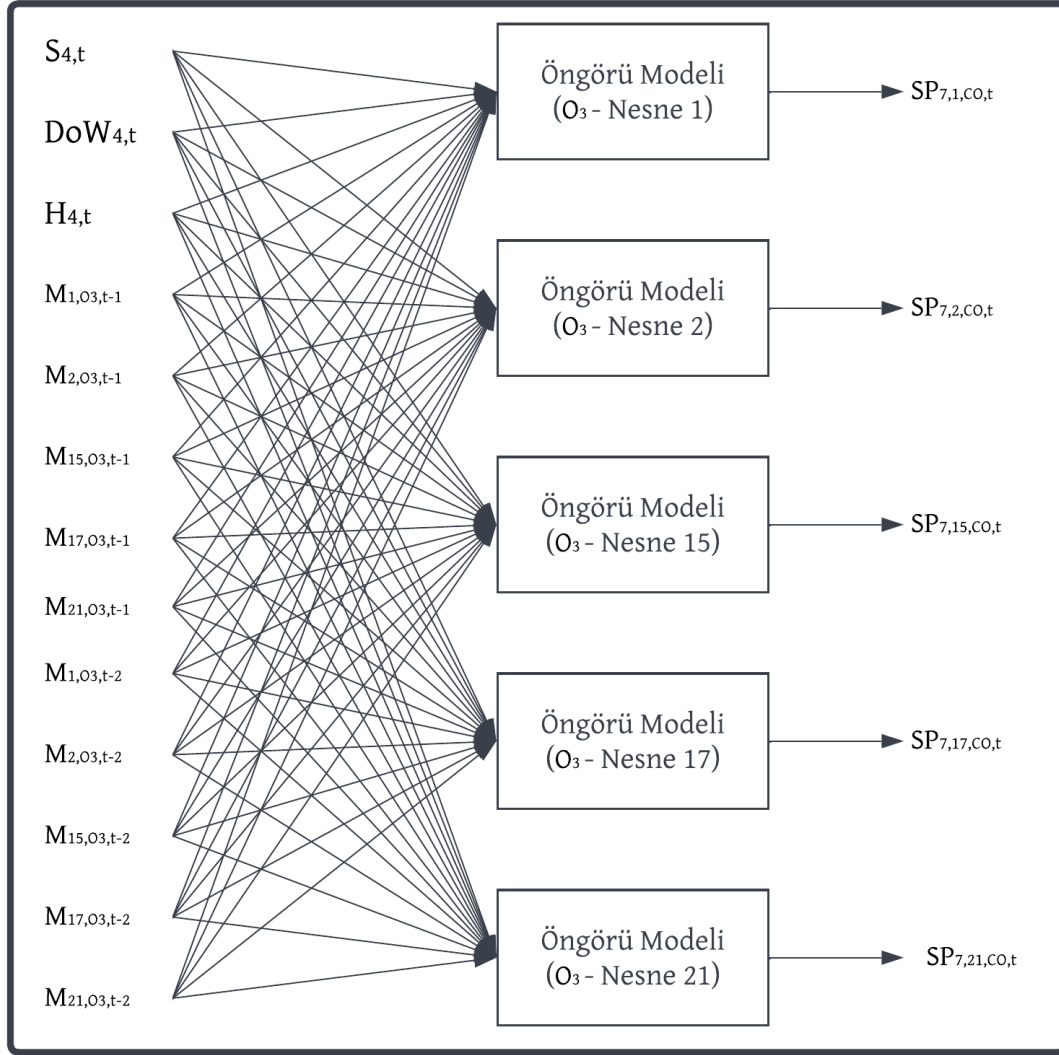
Şekil 4.3. Dördüncü istasyonda yapılan CPP öngörü modellerinin giriş ve çıkışları

SP ise aynı kirleticinin farklı nesnelerden alınan ölçümlerini kullanarak kirleticilerin konsantrasyonunu öngörür. Bu çalışma kapsamında CPP’de yapıldığı gibi zamansal ek özellikler kullanılarak (mevsim, haftanın günü, saat) doğruluk oranı arttırılmıştır. SP modelleri sis sunuculara konumlandırılmıştır. SP, CPP ile paralel bir şekilde saatlik çalışır ve son iki saatin ölçümlerini kullanır. IoT tabanlı mimaride her sis sunucusu bir kirleticinin SP ile öngörüsünü yapmaktan sorumludur. Sis sunucu, sorumluluğu kendisinde olan kirleticinin ölçüldüğü tüm nesneler için ayrı birer model çalıştırır. Elde ettiği öngörülerini bulut sunucuya iletir. Daha açıklayıcı olmak adına O₃ kirleticisinin SP öngörüsünden sorumlu olan sunucuda (7 numaralı istasyon) O₃ kirleticisini ölçen 5 nesne (1, 2, 15, 17 ve 21 numaralı

istasyonlar) için yapılan öngörüler aşağıda verilmiştir. $SP_{A,B,C,D}$ A sunucusunun B nesnesinin C kirleticisine dair D'inci zaman dilimine dair yaptığı öngörü, $ME_{E,F,G}$ E nesnesinin F kirleticisine dair G'inci zaman diliminde yaptığı ölçüm, $S_{H,j}$, $DoW_{H,j}$ ve $H_{H,j}$ sırasıyla H nesnesinin J'inci zaman dilimindeki mevsim, haftanın günü ve günün saati bilgisini, ÖNGÖR ise öngörü modelini ifade etmektedir.

- $SP_{7,1,CO,t} = \text{ÖNGÖR}(S_{1,t}, DoW_{1,t}, H_{1,t}, M_{1,O3,t-1}, M_{2,O3,t-1}, M_{15,O3,t-1}, M_{17,O3,t-1}, M_{21,O3,t-1}, M_{1,O3,t-2}, M_{2,O3,t-2}, M_{15,O3,t-2}, M_{13,O3,t-2}, M_{21,O3,t-2})$
- $SP_{7,2,CO,t} = \text{ÖNGÖR}(S_{1,t}, DoW_{1,t}, H_{1,t}, M_{1,O3,t-1}, M_{2,O3,t-1}, M_{15,O3,t-1}, M_{17,O3,t-1}, M_{21,O3,t-1}, M_{1,O3,t-2}, M_{2,O3,t-2}, M_{15,O3,t-2}, M_{13,O3,t-2}, M_{21,O3,t-2})$
- $SP_{7,15,CO,t} = \text{ÖNGÖR}(S_{1,t}, DoW_{1,t}, H_{1,t}, M_{1,O3,t-1}, M_{2,O3,t-1}, M_{15,O3,t-1}, M_{17,O3,t-1}, M_{21,O3,t-1}, M_{1,O3,t-2}, M_{2,O3,t-2}, M_{15,O3,t-2}, M_{13,O3,t-2}, M_{21,O3,t-2})$
- $SP_{7,13,CO,t} = \text{ÖNGÖR}(S_{1,t}, DoW_{1,t}, H_{1,t}, M_{1,O3,t-1}, M_{2,O3,t-1}, M_{15,O3,t-1}, M_{17,O3,t-1}, M_{21,O3,t-1}, M_{1,O3,t-2}, M_{2,O3,t-2}, M_{15,O3,t-2}, M_{13,O3,t-2}, M_{21,O3,t-2})$
- $SP_{7,21,CO,t} = \text{ÖNGÖR}(S_{1,t}, DoW_{1,t}, H_{1,t}, M_{1,O3,t-1}, M_{2,O3,t-1}, M_{15,O3,t-1}, M_{17,O3,t-1}, M_{21,O3,t-1}, M_{1,O3,t-2}, M_{2,O3,t-2}, M_{15,O3,t-2}, M_{13,O3,t-2}, M_{21,O3,t-2})$

Yukarıda verilen öngörü fonksiyonları daha anlaşılabilir olması için aşağıda Şekil 4.4'de görsel olarak da verilmiştir.



Şekil 4.4. Yedinci istasyonda yapılan SP öngörü modellerinin giriş ve çıkışları

4.2.4. Veri işlemleri

Her kirlotici için özel olarak belirlenen bir aralığının dışında bulunan ve yanlış ölçüm olma ihtimali yüksek olan veriler olağan dışıdır. Veride bulunan olağan dışı değerler NaN ile değiştirilmiş ve eksik veriler ile birlikte KNNImpute [196] algoritması ile doldurulmuştur. Olağan dışı değerlerin tespiti için kullanılan aralıklar Çizelge 4.3’de verilmiştir.

Çizelge 4.3. Kullanılan olağan dışı değer aralıkları

Kirletici	Değer Alt Sınır	Değer Üst Sınır
CO	0.0001	5000
NO2	0.0001	300
O3	0.0001	250
PM10	0.0001	750
SO2	0.0001	200

Veri işleminin ardından IoT temelli mimaride her saat başı yapılması tasarlanan öngörüler yapılmıştır. SP ve CPP öngörülerinde SVR algoritması kullanılmıştır. Öngörüler MatLab ile yapılmıştır ve 5-katlı çapraz doğrulama veri seçiminin etkisini azaltmak için uygulanmıştır. CPP karakteristiğindeki öngörü modellerinin [0,1] aralığına normalize edilmiş hata payları Çizelge 4.4’de verilmiştir.

Çizelge 4.4. CPP öngörü modellerinin hata oranları

İstasyon	Kirleticiler					Hata Oranları	
	CO	NO ₂	O ₃	PM10	SO ₂	MSE	RMSE
1		X	X	X	X	0,035	0,041
2		X	X	X		0,034	0,039
3	X	X		X	X	0,040	0,036
4	X	X		X	X	0,050	0,044
5	X			X	X	0,033	0,029
6	X	X		X	X	0,044	0,038
7		X		X	X	0,021	0,033
8		X		X	X	0,049	0,039
9	X	X		X		0,055	0,053
10				X	X	0,042	0,028
11	X	X		X		0,035	0,034
12		X		X	X	0,036	0,029
13				X		0,048	0,028
14	X	X		X		0,044	0,041
15	X	X	X	X		0,038	0,053
16	X	X		X	X	0,036	0,030
17		X	X	X	X	0,033	0,037
18		X				0,052	0,038
19	X			X	X	0,029	0,033
20	X	X		X		0,044	0,039
21	X	X	X	X	X	0,041	0,045
22		X		X		0,048	0,039
24				X		0,083	0,040
25	X	X		X		0,046	0,042
26	X	X		X	X	0,054	0,047
27	X			X	X	0,047	0,039
28				X	X	0,041	0,033
29		X		X	X	0,053	0,042
Ortalama						0,043	0,038

SP karakteristiğindeki öngörü modellerinin [0,1] aralığına normalize edilmiş hata payları

Çizelge 4.5’de verilmiştir.

Çizelge 4.5. SP öngörü modellerinin hata oranları

Kirletici	Hata Oranları	
	MSE	RMSE
CO	0,038	0,041
NO ₂	0,039	0,045
O ₃	0,022	0,058
PM10	0,042	0,030
SO ₂	0,035	0,029

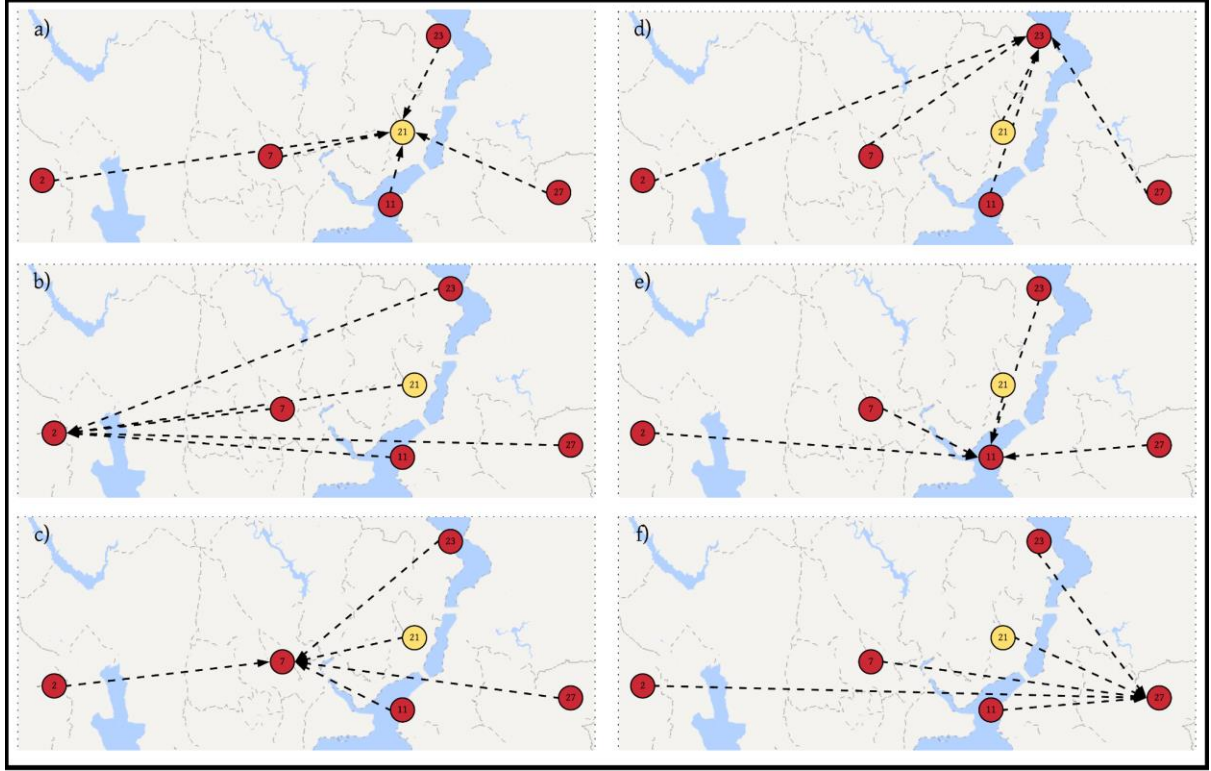
4.2.5. Veri akışları

Kullanılan veri ve tasarlanan IoT temelli mimari göz önüne alındığında aşağıda verilen üç çeşit veri akışının ağ üzerinde oluşması beklenmektedir.

- Tip 1 Veri Akışı (T1) algılanan kirletici seviyelerinin SP öngörüsü yapılması amacıyla ilgili sis düğümüne gönderilmesi sürecinde oluşan veri akışıdır. T1 akışı sis düğümleri arasındadır.
- Tip 2 Veri Akışı (T2) nesnelerde yapılan CPP karakterindeki öngörü sonuçlarının, sis düğümleri aracılığı ile bulut sunucusuna gönderilmesi sürecinde oluşan veri akışıdır. Sis düğümler ile bulut sunucu arasındadır.
- Tip 3 Veri Akışı (T3) sis düğümlerinde yapılacak olan SP karakterindeki öngörü sonuçlarının bulut sunucusuna gönderilmesi sürecinde oluşan veri akışıdır. Sis düğümleri ile bulut sunucu arasındadır.

T1, T2 ve T3 veri akışlarının toplamından oluşan ve ağ üzerinde oluşması beklenen ağ paketleri MatLab aracıyla oluşturulmuştur. Çalışma kapsamında, sis sunucular ve bulut sunucudan oluşan üst ağ katmanında oluşacak ağ trafiği kullanılmıştır. Bunun sebebi, nesneler ile sis sunucuların direkt bağlı olduğunun kabul edilmesidir. Her nesne kendinden sorumlu sis sunucusuna tek bir dedike bağlantı hattı üzerinden bağlanır. Dolayısıyla, nesne ile sis sunucu arası bağlantılarda optimize edilecek durum bulunmamaktadır. Ağın üst katmanında ise, farklı kaynaklardan gelen farklı önceliklere sahip paketler bulunmaktadır.

Bu sebeple ağın üst katmanında optimize edilebilecek bir menzil bulunmaktadır. T1, T2 ve T3 tipindeki veri akışlarının yönleri Şekil 4.5’de verilmiştir.



Şekil 4.5. Veri akışları a) T2, T3 b) T1(PM10), c) T1(O3), d) T1(NO₂), f) T1(SO₂)

4.2.6. Veri paketlerinin önceliklendirilmesi

Oluşturulan ağ akışı kirletici seviyelerine göre önceliklendirilmiştir. Daha yüksek hava kirliliğini öngören paketlere daha yüksek öncelik verilmiştir. T1 akışındaki paketlerin hepsi en düşük seviye önceliğe (Seviye 1) sahiptir. T2 ve T3 akışındaki veriler ise Çizelge 4.6’da verilen değerlere göre önceliklendirilmiştir.

Çizelge 4.6. Kirleticiler önceliklendirme sınırları

Kirleticiler	Seviye 1		Seviye 2		Seviye 3		Seviye 4		Seviye 5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
CO	0	4999	5000	9999	10000	14999	15000	19999	20000	~
NO ₂	0	49	50	99	100	149	150	199	200	~
O ₃	0	59	60	119	120	179	180	239	240	~
PM10	0	24	25	49	50	74	75	99	100	~
SO ₂	0	174	175	349	350	524	525	699	700	~

Toplam ağ trafiğini oluşturan paketlerin ise önceliklerine göre dağılımı analiz edilmiştir. Bütün paketlerin ve bulut sunucudaki paketlerin önceliklerine göre dağılımı Çizelge 4.7’de, sis sunucularınkiler ise Çizelge 4.8’de verilmiştir.

Çizelge 4.7. Bulut sunucudaki paketlerin öncelik seviyelerine göre dağılımı

Sunucu	Kirleticiler	Öncelik Seviyesi				
		1	2	3	4	5
Bulut Sunucu	CO	66 631	0	0	1	0
	NO ₂	50 800	14 333	1 316	38	1
	O ₃	28 816	4 032	13	0	3
	PM10	56 502	13 535	6 876	2 575	2 725
	SO ₂	33 009	0	0	0	3
	Paket Sayısı	235 758	31 900	8 205	2 614	2 732
	Yüzde	83,84	11,34	2,92	0,93	0,97
Toplam	Paket Sayısı	1.833 185	339 920	83 248	23 598	20 730
	Yüzde	80	15	4	1	1

Çizelge 4.8. Sis sunuculardaki paketlerin öncelik seviyelerine göre dağılımı

Sunucu	Kirleticiler	Öncelik Seviyesi				
		1	2	3	4	5
Sis Sunucu 1	CO	165 878	0	0	5	0
	NO2	43 550	17 884	4 676	406	31
	O3	31 570	876	0	0	18
	PM10	55 918	19 289	4 677	1 340	841
	SO2	49 497	0	0	0	1
	Paket Sayısı	346 413	38 049	9 353	1 751	891
	Yüzde	87,38	9,60	2,36	0,44	0,22
Sis Sunucu 2	CO	32 546	0	0	1	0
	NO2	159 883	61 251	9 837	802	173
	O3	29 459	2 486	8	0	9
	PM10	36 275	9 541	1 904	482	220
	SO2	0	0	0	0	0
	Paket Sayısı	258 163	73 278	11 749	1 285	402
	Yüzde	74,86	21,25	3,41	0,37	0,12
Sis Sunucu 3	CO	16 680	0	0	0	0
	NO2	52 497	11 339	2 131	245	27
	O3	118 354	24 951	580	5	763
	PM10	37 968	14 235	7 351	3 088	2 757
	SO2	65 402	0	0	0	0
	Paket Sayısı	290 901	50 525	10 062	3 338	3 547
	Yüzde	81,17	14,10	2,81	0,93	0,99
Sis Sunucu 4	CO	49 393	0	0	1	0
	NO2	62 232	3 630	103	0	6
	O3	39 598	8 671	361	2	104
	PM10	124 205	105 986	37 252	12 644	11 197
	SO2	65 271	1	0	0	2
	Paket Sayısı	340 699	118 288	37 716	12 647	11 309
	Yüzde	65,44	22,72	7,24	2,43	2,17
Sis Sunucu 5	CO	24 646	0	0	0	0
	NO2	42 384	6 558	552	15	66
	O3	35 663	4 717	175	1	54
	PM10	56 049	16 605	5 436	1 947	1 720
	SO2	202 509	0	0	0	9
	Paket Sayısı	361 251	27 880	6 163	1 963	1 849
	Yüzde	90,52	6,99	1,54	0,49	0,46

Çizelge 4.7 ve Çizelge 4.8’de görüldüğü üzere kritik sınırı en çok aşan kirleticilerin PM10, O3 ve NO2 olduğu görülmektedir. Bunlar arasında PM10 en kritik kirleticidir. Bu çıkarımın yapılabilmesinin arkasında kritik eşiği aşan paketlerin büyük çoğunluğunun PM10’a ait olması gelmektedir. CO ve SO2 kirleticilerinin kritik sınırı aşan çok az sayıda öngörüsü bulunmaktadır. NO2 ve O3 kirleticilerine dair örüntüler düzenli değildir ve sis düğümleri arasında değişmektedir. Bir numaralı sis sunucusunda (istasyon 11) öngörülen hava kirliliğinde NO2 ve O3 etkisi oldukça azdır. Az sayıda orta seviyeli kirliliğe katkıları görülmektedir, yüksek seviyede ise oldukça azdır. İki numaralı sis sunucusunda (istasyon 23) NO2 önemli bir kirleticidir, O3 kirleticisinin ise oldukça az etkisi görülmektedir. Üç numaralı sis sunucusunda (istasyon 7) NO2 ve O3 kirleticilerinin etkisi düşük hava kirliliği ile orta seviyeli hava kirliliği arasında değişmektedir. Yüksek seviyeli kirlilikte ise O3 kirleticisinin büyük oranda etkili olduğu görülmektedir. Dört numaralı sis sunucusunda (istasyon 2) NO2 ve O3 kirleticilerinin hava kirliliğine oldukça sınırlı etkisi olduğu görülmektedir. Bu sunucuda esas kirleticinin PM10 olduğu görülmektedir. Benzer şekilde beş ve altı numaralı sis sunucularında (istasyonlar 27 ve 21) NO2 ve O3 kirleticilerinin orta ve yüksek hava kirliliğine katkısının oldukça kısıtlı olduğu görülmektedir.

Yapılan analiz İstanbul ilinde hava kirliliğine sebep olan esas kirleticinin PM10 olduğunu göstermektedir. PM10’un arkasından O3 ve NO2 kirleticileri gelmektedir. CO ve SO2 kirleticilerinin ise hava kirliliğine oldukça kısıtlı etki yaptığı görülmüştür.

Önerilen yaklaşımda belirli kirleticilere dair yapılan öngörülerden yüksek hava kirliliğine sahip olan paketlere öncelik verilecektir. Bu önceliklendirme yaklaşımında kirletici türüne dikkat edilmemektedir, yüksek seviyede hava kirliliği öngören paketlere öncelik verilmektedir. Bu sebeple, kritik olacak verinin büyük oranda PM10 paketlerinden oluşacağı beklenmektedir. Öncelikli paketler arasında düşük bir oranda NO2 ve O3 paketi olması, SO2 ve CO paketlerininse oldukça nadir olması beklenmektedir.

4.4. Test Sonuçları

Çalışma kapsamında SDN konusunda SDN temelli içerik tabanlı yönlendirme (Content based forwarding via SDN / CBF-SDN) ve SDN temelli önceliklendirme etkinleştirilmiş içerik tabanlı yönlendirme (Priority Enabled Content Based Forwarding via SDN / PCBF-SDN) iletişim modelleri geliştirilmiştir. Bunlardan CBF-SDN, FC kavramını kullanarak

paket iletim zamanını düşürmektedir. PCBF-SDN ise CBF-SDN modeline önceliklendirme yeteneği ekleyerek kritik verilerin hedef konumlarına daha hızlı varmasını sağlar. Modeller ile detaylı bilgi için yayınlanan makaleye bakılabilir [19].

PCBF-SDN modelinde paket öncelikleri veri paketlerinde bulunan Type-of-service (ToS) biti ile belirtilmiştir. Bu yöntem seçiminden dolayı paketlerdeki beş seviyeli önceliklendirme sadeleştirilmiştir. Çizelge 4.9’da verilen değerler üzerinden veri paketleri öncelikli ve öncelikli olmayan olmak üzere ikiye ayrılmıştır.

Çizelge 4.9. Kirleticilerin önceliklendirme şartları

Kirletici	Önceliklendirme Şartı
CO	> 15000
NO ₂	> 100
O ₃	> 120
PM10	> 25
SO ₂	> 175

Oluşturulan senaryo GNS3’te tasarlanmış Docker üzerinde 70 switch, 36 sanal makina kullanılarak oluşturulmuş bir ağ topolojisi üzerinde test edilmiştir. Simülasyon ortamına ve kullanılan SDN teknolojilerine dair detaylı bilgi için yayınlanan makaleye bakılabilir[19]. Simülasyon sonuçları Çizelge 4.10’da verilmiştir.

Çizelge 4.10. Test sonuçları

Model		Ortalama Gecikme (ms)
Konvansiyonel İletişim		1,10
CBF-SDN		1,02
PCBF-SDN	Öncelikli Veri	0,80
	Önceliksiz Veri	1,30
	Tüm Veri	1,23

Çizelgede görüldüğü üzere önerilen ilk model CBF-SDN konvansiyonel IoT iletişim yöntemlerine göre daha düşük gecikme ile veri transferi yapmaktadır. Bu durum FC kavramının uygulanmasının ağ iletişimi üzerinde etkisini ortaya koymaktadır. PCBF-SDN ise ortalama gecikmeyi yükseltse de, öncelikli paketlerin hedefe erişim zamanını oldukça düşürmüştür. Öncelikli verinin daha düşük gecikme ile hedefe varması bu çalışmanın ana konusuydu ve test sonuçları bu amacın yerine getirildiğini göstermektedir.

4.5. Değerlendirme

Bu bölümde anlatılan çalışma [19] kapsamında IoT’de farklı QoS seviyelerine sahip uygulamaların QoS-farkında olarak kaynakların etkin bir şekilde paylaşılmasına dair bir öneri yapılmıştır. Ağ gecikmesi odaktaki QoS isteri olarak seçilmiştir. Önerilen yöntem PCBF-SDN, FC ve SDN kavramlarının yardımı ile veri paketlerini önceliklendirerek kritik paketlerin hedefe daha düşük ağ gecikmesi ile ulaşmasını sağlamayı amaçlamaktadır. Önerilen yöntemin test edilmesi bir hava kirliliği öngörü senaryosu oluşturulmuştur. Senaryo Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı’nın Hava Kalitesi İzleme Ağına dahil İstanbul ilindeki 29 hava kalitesi takip istasyonunun verileri ve konumları kullanılarak oluşturulmuştur. Senaryoda son iki saatin verileri kullanılarak gelecek saatin hava kirliliği ön görülmeye çalışır. Yüksek hava kirliliğini öngören paketler ise yüksek öncelikli paketler olarak etiketlenmiştir. Senaryo gerçekçi bir IoT ağ topolojisine oturtulmuş ve simüle edilmiştir. Simülasyon sonuçları PCBF-SDN iletişim modelinin başarısını göstermektedir. PCBF-SDN iletişim modeli öncelikli paketlerin ağ gecikmesini %27 oranında düşürmüştür.

5. IOT UYGULAMALARININ QOS KISITLARI ALTINDA ADIL VE ENERJİ ETKİN KOMPOZİSYONU

IoT birbirine bağlı, coğrafi olarak dağılmış milyarlarca nesneden oluşan ve kullanıcılara yüksek çeşitlilikte uygulamalar sunan bir yapıdır [1, 240]. Günümüzde IoT nesnelerinin sayısının 30 milyarı aştığı ve bu sayının hızlı bir şekilde yükseldiği bildirilmektedir. IoT görece yeni bir kavram olmasına rağmen hali hazırda endüstriyel üretim süreçlerini ve insan hayatını devrimsel bir ölçekte değiştirmiştir [241-243]. IoT uygulamaları bulut sunucuların yardımıyla sağlık, akıllı evler ve şehirler, lojistik, sosyal uygulamalar ve benzer birçok alanda kullanılmaktadır [1, 244].

IoT kavramında her nesne bir veya birkaç işlev sunar. Bu işlevler SOA uyarınca servisler olarak enkapsüle edilir ve IoT servisi olarak adlandırılır. IoT ağı oluşturan nesneler genellikle kısıtlı enerji, bellek ve hesaplama kapasitesine sahiptir. Bu sebeple IoT servisleri genellikle oldukça basittir [244]. Ancak, IoT ağı kullanıcıları IoT servislerinin sunduğu işlevlerden çok daha karmaşık işlevleri talep edebilirler. Basit IoT işlevleri birleştirilerek karmaşık işlevler elde edilebilir [245, 246]. Birleştirme işlemi çok sayıda basit IoT servisini, talep edilen işlev akışına uygun bir şekilde bir IoT uygulamasına çevirme sürecidir [242, 247]. Bu işleme literatürde servis birleştirme adı verilir. SC sonucu elde edilen IoT uygulamaları basit servislerin sunamayacağı işlevleri sunabilir [248]. SC sürecinde göz önüne alınması gereken kriterler bulunmaktadır. Bunlar QoS, enerjinin etkin kullanımı ve IoT servisleri arası adil iş yükü dağılımının sağlanmasıdır.

QoS, IoT kavramının en çekirdek bileşenlerinden biridir [249]. IoT’de her servis ve uygulama QoS istekleri ile karakterize edilir [250]. QoS metriği bir servisin/uygulamanın servis kalitesi gereksinimlerini gösterir. Örnek vermek gerekirse, bir video uygulamasının kalitesi video kalitesi ile ölçülür, bu sebeple kritik QoS metriği video kalitesidir. Bir başka örnek olarak çevre izleme uygulamalarını ele alalım. Bu uygulamalarda kritik QoS metriği veri bütünlüğü ve hata oranıdır. IoT uygulamalarının QoS metrikleri oldukça çeşitlidir. Uygulamaların QoS isteklerinin sağlanması SC probleminin en önemli gereksinimlerinden biridir. QoS-farkında SC yaklaşımları, basit servisleri birleştirirken talep edilen işleve ek olarak uygulamanın talep ettiği QoS isteklerini de göz önüne alır [251]. Talep edilen QoS seviyesi, servis sağlayıcı tarafından Servis Seviyesi Anlaşmaları (Service Level Agreement / SLA) ile garanti edilir. QoS-farkında SC sürecinde, esas amaç birleştirilen uygulamanın SLA kontratına uygun olmasının sağlanmasıdır.

Nesneler boyut olarak küçüktür ve düşük enerji profillerine sahiptir. Ancak, toplam sayıları on milyarları bulan nesnelerin çalışmak için harcadıkları enerji ve nesneleri birbirine bağlayan ağ altyapısının harcadığı enerji miktarı oldukça yüksektir [243, 252]. 2020 yılı itibari ile üretilen karbon ayak izinin %6 ila %8'inin bilgi teknolojileri ve altyapıları tarafından üretileceği öngörülmektedir [253]. Bu oranın, IoT karakterindeki ağların çok daha yaygın bir hale gelmesi ile birlikte yükselmesi beklenmektedir. Bu artışın önüne geçmek adına GIoT kavramı IoT ağların daha sürdürülebilir ve çevre dostu bir hale getirilmesi adına öne sürülmüştür [252, 253]. GIoT'un önerdiği yaklaşımlardan biri nesnelerin enerji tüketiminin düşürülmesidir [252]. Ek olarak, sürdürülebilir IoT ağların oluşturulabilmesi için IoT ağın tüm operasyonlarının da enerji tüketimini göz önünde tutarak yapılması beklenmektedir. Yakın gelecekte, enerji etkinliği göz önünde olmadan yapılan işlemlerin IoT dünyasından tamamen dışlanması beklenmektedir. Sektörde yapılan uygulamalar ve literatürde yapılan çalışmalar bunu açıkça göstermektedir. Enerji etkinliği geleceğin teknoloji dünyasının bir zorunluluğu haline gelecektir. Bu sebeple, IoT uygulamalarının SC süreçlerinde enerji tüketimi sürecin içine katılmalıdır.

SC sürecinde göz önünde tutulması gereken bir diğer kriter IoT servisleri arasında iş yükünün adil bir şekilde dağıtılmasıdır [254, 255]. IoT servislerinin dengeli ve adil bir şekilde dağıtılması daha düşük hata oranlarına neden olur ve sağlanan QoS seviyelerini yükseltir [179, 256]. Dengesiz iş yükü dağılımları ağdaki bir grup IoT servisinin art arda kullanılmasına, kalan servislerin çoğunlukla boş durmasına sebep olur. Bu durumda, art arda kullanılan IoT servislerinde donanımsal hatalar ve QoS seviyelerinde düşüş gözlenir, çok kullanılan IoT servislerine ulaşan ağ bağlantılarında tıkanmalar meydana gelir [256]. Adil iş yükü dağıtımının etki ettiği bir diğer husus ise enerji tüketimine dairdir. IoT cihazlarının önemli bir bölümü dedike bir enerji hattına sahip değildir ve enerjisini bataryadan sağlar [254]. Aynı servisin art arda kullanılması servisi sağlayan nesnenin bataryasının erken tükenmesine, dolayısıyla kısa süre içerisinde ağ fonksiyonlarında kayba sebep olur [254]. Bu sebeplerden ötürü, SC sürecinde iş yükünün IoT servisler arasında adil dağılımını sağlayan mekanizmalara ihtiyaç vardır.

Günümüzde IoT uygulamalarının SC süreçleri bulut bilişim desteği ile yapılmaktadır. Ancak bulut tabanlı altyapı çoğu IoT uygulamasına uygun değildir [257, 258]. Cisco tarafından önerilen FC, bulut sunucuların sağladığı hesaplama ve veri saklama yeteneklerini ağın uç ve son kullanıcıya kısmına yaklaştırır [249]. Son kullanıcıya sis sunucularda yapılan işlemler

daha düşük gecikme ve bant genişliği harcamasını sağlar [79, 80], ağ kullanım verimini arttırır. Ağ katmanına sağladığı avantajlara ek olarak FC, ağların ölçeklenebilirlik ve heterojen yapısını bulut bilişime kıyasla daha yüksek seviyede destekler. Bahsedilen avantajlardan yola çıkarak FC temelli yaklaşımların özellikle ağ gecikmesi konusunda hassas olan uygulamalar için çok daha uygun olduğu söylenebilir [29, 258-260].

Bu bölümde anlatılan çalışma [17] kapsamında enerji tüketim ve adil iş yükü dağıtımının sağlanması için FC kavramından yararlanılmıştır. Bu çalışmada QoS, enerji tüketimi ve IoT servisleri arasında adil iş yükü dağılımı objektiflerine sahip IoT servislerinin çok hedefli kompozisyonu incelenmiştir. Önerilen yaklaşımda en temel hedef uygulamalar birleştirilirken SLA kontratlarına uygun hareket edilmesidir, dolayısıyla QoS hedefidir. İkinci hedef enerji tüketiminin GIoT kavramı ile paralel bir şekilde minimize edilmesidir. Adil iş yükü dağılımına dair olan son objektif ise, aynı servislerin art arda kullanımının minimize edilmesini sağlamayı amaçlar. IoT’de SC problemi NP-hard bir problem olarak tanımlanmaktadır [17]. Bu sebepten ötürü, önerilen yaklaşım üç objektifi sağlayarak uygulamaları oluşturmak için optimizasyon yaklaşımını kullanır. Problem bir optimizasyon problemi olarak formüle edilmiştir ve NSGA-II algoritması ile çözülmüştür. Önerilen yaklaşımın etkinliğinin gösterilmesi için detaylı testler yapılmıştır. Test sonuçları toplam enerji tüketiminde %10’luk bir düşüşü, aynı servislerin art arda kullanımının büyük oranda engellendiğini ve bu enerji etkinliği ile adil iş yükü objektifleri sağlanırken QoS seviyesinin yüksek seviyede düşmediği görülmüştür. Literatüre yapılan katkı aşağıda özetlenmiştir.

- IoT uygulamalarının SC ile elde edilmesi için FC temelli yeni bir yaklaşım önerilmiştir.
- IoT servisleri arasında adil iş yükü dağılımı problemi incelenmiş ve optimizasyon sürecinin SC sürecinin bir hedefi olarak tanımlanmıştır. Bildiğimiz kadarıyla bu çalışma, IoT servisleri arası adil iş yükü dağılımını bir hedef olarak kullanan ilk çalışmadır.
- Bildiğimiz kadarıyla, önerilen yaklaşım enerji etkinliği, adil iş yükü dağılımı ve QoS hedeflerini IoT SC problemlerine birlikte uygulayan ilk çalışmadır.
- Literatürdeki çoğu çalışmanın aksine, toplam enerji tüketimi IoT’de SC probleminin bir objektifi olarak tanımlanmıştır.

5.1. İlgili Çalışmalar

SC süreci kullanıcılar tarafından talep edilen işlevlerin, basit işlevler sağlayan çok sayıda servisin birleştirilmesi ile elde edilmesine dayanır. Bu alanda yapılan öncül çalışmalar tamamen bu odağa sahiptir, servislerin birleştirilmesini sağlayan yapıların ve metodolojilerin geliştirilmesini amaçlar. Bu sebeple, öncül çalışmalarda QoS veya enerji etkinliği gibi ek hedefler bulunmamaktadır. Çalışmalarında Akasiadis ve arkadaşları [261] akıllı ortamlar için bir SC şeması önermişlerdir. Görseller üzerinde yapılan arka plandan kurtulma, obje tespiti ve obje takibi gibi operasyonlar basit servis olarak tanımlanmıştır. Sıralanan basit servisler birleştirilerek, bir odadaki insanları sayan bir uygulama elde edilmiştir.

Çalışmalarında Yachir ve arkadaşları [262] FASEM (Framework for Ambient Services and Events Monitoring) yapısını önermişlerdir. QoS faktörleri olarak uygunluk (availability), tepki zamanı (response time) ve tepki olasılığı (response probability) seçilmiş ve uygulamaya ait tek bir QoS seviyesini gösterecek şekilde birleştirilmiştir. İki fazdan oluşan yapıda, uygulama oluşturmak için kullanılacak IoT servislerinin seçimi için yerel bir servis seçim süreci yapılır. Servis seçiminde QoS seviyesi göz önüne alınır. Ardından global fazda seçilen servisler kullanılarak uygulama oluşturulur.

Çalışmalarında Adewuyi ve arkadaşları [263] SC sürecinde güven sağlanması problemine eğilmişlerdir. Önceki çalışmalarında [264] önerdikleri CTRUST modelini daha yüksek başarımla ile güven seviyesi tahmin edebilecek şekilde geliştirmişlerdir. Önerilen SC-TRUST modeli oluşturulan uygulamanın güven seviyesini, uygulamayı oluşturmak için kullanılan servislerin güven seviyelerini birleştirerek elde etmektedir.

Günümüze yaklaştıkça SC yaklaşımlarında QoS kriterine ek olarak yeni kriterlerin de sürece katıldığı görülmüştür. Böyle durumlarda problem karmaşıklığı artmaktadır, dolayısıyla optimizasyon algoritmaları kullanılmaktadır. Çalışmalarında Khanouche ve arkadaşları [248] IoT’de SC sürecinde QoS kısıtları altında enerji tüketiminin optimize edilmesini amaçlamışlardır. Bu amaçla problem çok hedefli optimizasyon problemi olarak formüle edilmiştir. Önerilen yaklaşım olan enerji-merkezli ve QoS-farkında servis seçim algoritması (Energy-Centered and QoS-Aware Services Selection Algorithm / EQSA) iki fazdan oluşmaktadır. İlk fazda servislerin ön seçimi yapılarak optimizasyon probleminin çözüm uzayı küçültülür. Servisler, sağladıkları QoS seviyesinin kullanıcı tarafından talep edilen

QoS seviyesine uygunluğuna göre etiketlenir. Başarılı servisler sonraki faza gönderilir. İkinci fazda servisler birleştirme sıralama (Merge Sort) algoritması ile sıralanır. Bu sıralamada hem QoS seviyeleri hem de enerji profilleri göz önüne alınır. İki kritere göre sıralama yapılabilmesi için Pareto-hakimiyet (Pareto Dominance) sıralama esası kullanılır.

Benzer bir yaklaşım Chai ve arkadaşlarının [265] çalışmasında kullanılmıştır. Çalışmada kullanıcı tarafından tanımlanan QoS kısıtları altında enerji tüketimi optimizasyonu amaçlanmıştır. Bu amaçla hiyerarşik optimizasyon kullanılmıştır. Önerilen yaklaşım olan hızlı enerji-merkezli ve QoS-farkında servis birleştirme (Fast Energy-Centered and QoS-Aware Service Composition / FSCA-EQ) yaklaşımı öncelikle QoS seviyesine göre servis seçimi yapar. Akabinde pareto hakimiyet kullanarak enerji tüketimini minimize etmeye çalışır.

Graf temelli SC yaklaşımları literatürde bulunmaktadır. Çalışmalarında Alsaryrah ve arkadaşları [266] QoS ve enerji perspektifinden probleme yaklaşmışlardır. Objektifler enerji etkinliği ve bileşik QoS değerinin sağlanmasıdır. Çok sayıda QoS isterinin değeri basit ağırlıklı toplama (Simple Additive Weighting / SAW) yaklaşımı ile tekilleştirilmiştir. Problem çift hedefli optimizasyon problemi olarak formüle edilmiş ve nabız algoritması (Pulse Algorithm / PA) isimli en kısa yol algoritması ile çözülmüştür. Önerilen yaklaşım hem çok objektifli bir optimizasyon probleminin SAW ve graf algoritmasının beraber kullanımı ile çözülmesini önerir. Bir sonraki çalışmalarında Alsaryrah ve arkadaşları [267] üç fazlı melez bir yaklaşım önermişlerdir. İlk fazda global QoS isterleri lokal QoS isterlerine kırılır. Ardından lokal QoS isterlerine göre çözüm uzayı küçültülür. Son fazda pareto hakimiyet kullanılarak pareto çözümler oluşturulur. Çözüm uzayının küçültülmesi için aday servislere üç farklı filtre uygulanır. Filtrelerden geçemeyen servisler ve bu servisleri kullanan çözümler çözüm uzayından çıkarılır. Filtreler fonksiyon filtresi, QoS filtresi ve enerji tüketim filtresidir. Çözümlerin üretilmesi için en kısa yol algoritması kullanılır. Üretilen veri üzerinde testler yapılmıştır. Test sonuçları önerilen yaklaşımın yakın-optimal QoS seviyeleri elde ederken enerji tüketimini düşürdüğünü göstermektedir.

Sezgisel yaklaşımlar optimizasyon problemlerinde çözüm uzayını etkin bir şekilde arama yeteneğine sahiptir. Kurdi ve arkadaşları çalışmalarında [268] çoklu bulut ortamları (Multi-Cloud Environment / MCE) için bir SC yaklaşımı önermişlerdir. Önerilen yaklaşım uygulama birleştirme zamanı ve uygulama minimalitesini optimize etmek için guguk kuşu

arama algoritması (cuckoo-search algorithm / CSA) kullanır. Önerilen yöntemi değerlendirmek için iki set test yapılmıştır. Test sonuçlarında iki hedefinde de efektif bir şekilde optimize edildiği görülmüştür.

Çalışmalarında Sun ve arkadaşları [269] akıllı ortamlar için üç katmanlı bir SC yapısı önermişlerdir. Önerilen yapının test edilmesi için ACO, GA ve PSO algoritmaları kullanılmıştır. Bir sonraki çalışmalarında [270] IoT servislerinin enerji etkin birleştirilmesi için Gri Kurt Optimizasyonu (Grey Wolf Optimization / GWO) ve PSO algoritmalarının kullanımını önermişlerdir. Çalışma eş-zamanlı servislerin birleştirilmesi problemine özelleşmektedir. Test sonuçları önerilen yaklaşımın enerji tüketimini zaman ve mekan kısıtları altında etkin bir şekilde azalttığını göstermektedir.

Zhou ve arkadaşlarının çalışmalarında [271], WSN'ler için yakın-optimal SC problemine eğilmişlerdir. Çalışmada GA ve PSO algoritmaları, zaman ve mekan kısıtları altında enerji tüketiminin minimizasyonu için kullanılmıştır.

Khanouche ve arkadaşları [250], iki aşamalı bir SC yaklaşımı önermişlerdir. Önerilen yaklaşımda aday servisler QoS seviyelerine göre k-means ile kümelenir. Ardından en yüksek QoS seviyelerine sahip küme uygulamaların oluşturulması için bir sonraki faza gönderilir. karıştırılmış kurbağa zıplama algoritması (Shuffled Frog Leaping Algorithm / SFLA) ile yakın-optimal uygulama birleştirme şemaları elde edilir. Test sonuçları kümeleme adımının etkin bir şekilde arama uzayını küçülttüğünü göstermektedir. Bir sonraki çalışmalarında [244] IoT'de SC problemi için öğretme-öğrenme temelli optimizasyon (Teaching Learning based Optimization / TLBO) algoritmasının kullanımını önermişlerdir.

Badawy ve arkadaşları çalışmalarında [102] geri adımlı arama optimizasyonu algoritması (Backtracking Search Optimization Algorithm / BSOA) temelli SC yaklaşımını önermektedir. BSOA'nın genetik tekniği ile global QoS isterleri lokal isterlere bölünür, servis seçimleri için lineer arama kullanılır.

Li ve arkadaşlarının çalışmalarında [272] geniş IoT ağlar için GA temelli bir SC yaklaşımı önerilmiştir. Ağdan toplanan ampirik bilgi GA'in yakınsama yeteneğini geliştirmek için kullanılmıştır.

Büyük çaplı IoT ağlara odaklanan bir başka çalışma [273] Esmailyfard ve Naderi tarafından yapılmıştır. Çalışmada karmaşık olay servislerinin birleştirilmesi için dağıtık bir yaklaşım önerilmektedir. Önerilen yaklaşım ölçeklenebilirliği arttırmak için FC, adaptasyon yeteneğini arttırmak için iki sezgisel algoritmanın kullanımı önerir.

Çalışmalarında Razian ve arkadaşları [274] bulut temelli IoT (CloudIoT) ortamında SC için bir yaklaşım önerirler. Önerilen ölçeklenebilir anomali-farkında yaklaşım (Scalable Anomaly-Aware Approach / SAIoT) daha iyi QoS sağlanması için basit servislerin QoS seviyelerini öngörür. Bu amaçla log dosyaları kullanılır. Ancak çalışmada belirtildiği üzere log dosyalarındaki anomaliler öngörü işleminin doğruluğuna zarar verebilir. Bu durumun önüne geçmek için izolasyon ormanı (Isolation Forest / IS) algoritmasını log dosyalarındaki anomalileri tespit etmek için kullanır. SAIoT birleştirme şemalarını elde etmek için ACO kullanır.

Sefati ve arkadaşlarının çalışmasında [275] IoT'de SC problemi için HMM ve ACO kullanımı önerilmiştir. HMM ile QoS seviyeleri öngörülür. Ardından öngörülen QoS seviyesi ve ACO algoritması birleştirme şemaları elde etmek için kullanılır.

IIoT ortamına odaklanan Sourı ve arkadaşlarının çalışmasında [276], endüstriyel IoT uygulamalarının birleştirilmesi için resmi onaylama (formal verification) temelli SC yaklaşımı önerilmiştir. Birleştirilen uygulamaların QoS seviyelerinin geliştirilmesi için balina optimizasyonu algoritması (Whale Optimization Algorithm / WOA) kullanılmıştır.

NN temelli yaklaşımlar web servislerinin birleştirilmesi için yıllardır kullanılmaktadır. Benzer yaklaşımlar IoT servislerinin birleştirilmesi için de bulunmaktadır. Çalışmalarında Gharineiat ve arkadaşları [277] DRL kullanımını önermişlerdir. Önerilen yaklaşım kararlı bir şekilde hareket eden IoT servislerini de göz önüne alarak yazarın önceki çalışmalarını [278, 279] geliştirmektedir. Ayrıca, büyük veri teknoloji kullanılarak aday servisler filtrelenmiş ve çözüm uzayı küçültülmüştür.

İlgili çalışmalar Çizelge 5.1'de listelenmiştir. Görüldüğü üzere IoT'de SC problemi genellikle iki objektife odaklanmaktadır, enerji tüketimi ve QoS. Diğer objektifler literatürde olmasına rağmen kullanımları yaygın değildir. Çözüm yöntemi açısından bakılırsa, IoT'de

SC probleminin çözümü için en sık kullanılan yöntem türünün sezgisel optimizasyon algoritmaları olduğu görülmektedir.

Çizelge 5.1. SC alanındaki çalışmalar

Çalışma	Yıl	Objektifler				Yaklaşım/ Algoritma	Platform
		İşlev	QoS	Enerji	Ek Objektifler		
[261]	2015	✓				Yapı Önerisi	IoT
[262]	2016	✓				Yapı Önerisi	IoT
[248]	2016	✓	✓	✓		Ön seçim, Pareto hakimiyet	IoT
[266]	2018	✓	✓	✓		PA	IoT
[268]	2018	✓	✓		Birleştirme Basitliği	CSA	MCE
[272]	2018	✓	✓			GA	IoT
[269]	2018	✓	✓	✓		GA, PSO, ACO	IoT
[267]	2019	✓	✓	✓		En kısa yol yaklaşımı	IoT
[102]	2019	✓	✓			BSOA	IoT
[250]	2019	✓	✓			K-means, SFLA	IoT
[270]	2019	✓	✓	✓		GWO, PSO	IoT
[271]	2019	✓	✓	✓		GA, PSO	WSN
[244]	2020	✓	✓			TLBO	IoT
[273]	2021	✓				Sezgisel algoritmalar	IoT
[274]	2021	✓	✓		Ölçeklenebilirlik	IS, ACO	CloudIoT
[263]	2021	✓			Güven	Model Önerisi	IoT
[275]	2021	✓	✓			HMM, ACO	IoT
[276]	2021	✓	✓			WOA	IIoT
[265]	2021	✓	✓	✓		Ön seçim, Pareto hakimiyet	IoT
[277]	2021	✓	✓			DRL	IoT
Önerilen yaklaşım		✓	✓	✓	Adil İş Yükü Dağılımı	NSGA-II	IoT

5.2. Konsept Tasarımı

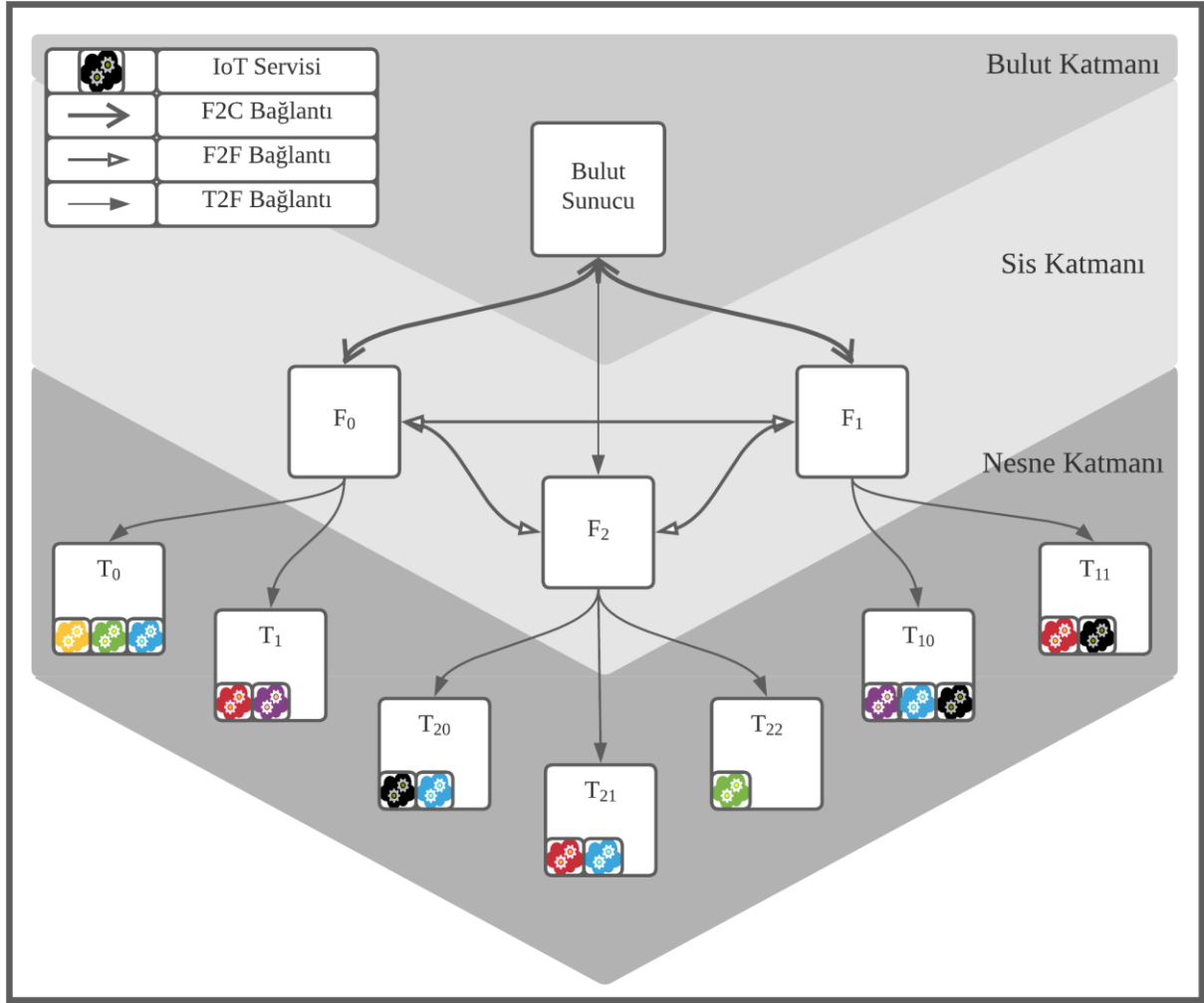
Bu bölümde anlatılan çalışma kapsamında IoT servisleri için FC temelli bir birleştirme yaklaşımı önerilmektedir. Bu bölümde kabuller ve tasarlanan ve yaklaşımın değerlendirilmesi için kullanılan IoT ağ modeli kısaca anlatılacaktır. Kullanılan kısaltmalar Çizelge 5.2’de verilmiştir.

Çizelge 5.2. Notasyonlar

Kısaltma	Açıklama
N	Ağ
C	Bulut Sunucu
F	Sis Sunucu
T	Nesne
S_T	T tarafından sunulan IoT servislerinin listesi
$S_{T,n}$	T tarafından sunulan n’inci IoT servisi
EP_s	s’nin enerji profili
EP_T	T’nin enerji profili
$QoSV_s$	s’nin QoS vektörü
$QoSA_{s,n}$	$QoSV_s$ ’nin n’inci metriği
c	Servis sınıfı
C_s	s’nin servis sınıfı
App	Uygulama
WF_{APP}	APP’in birleştirilmesi için gerekli taslak
SLA_{APP}	APP’in SLA kontratı
$SSLA_{APP}$	APP’in Alt SLA kontratı

5.2.1. Ağ modeli

Sis temelli ağ [17]'de bahsedildiği üzere üç katmanlı ağaç-graf şeklinde tasarlanmıştır. Ağın en üst katmanı bulut katmanıdır. Bulut katmanında grafin kökü olan bulut sunucu bulunur. Ağ (N) fn sis sunucu sayısı olmak üzere $N = \{C, \{F_1, F_2, \dots F_{fn}\}\}$ şeklinde tanımlanmıştır. Ağın ikinci katmanı sis sunuculardan oluşan sis katmanıdır. FC kavramına göre [280], sis sunucular veri saklama ve hesaplama yeteneklerini ağın uç kısmına getirir [257, 260, 281-284]. Bu çalışma kapsamında sis sunucuları SC işleminin bulut sunucu yerine sis sunucularda yapılmasını önermektedir. Sis katmanı bulut katmanı ile nesneler katmanını birbirine bağlar. Her sis sunucusuna çok sayıda nesne bağlanır. Bir sis sunucusu (F), tn F_{id} sunucusuna bağlanan nesne sayısı olmak üzere $F = \{T_1, T_2, \dots T_{tn}\}$ olarak tanımlanır. Nesneler ağın en alt katmanında yer alır ve üçüncü katmanı oluşturur. Her nesne çeşitli sayıda ve işlev sahip IoT servisi barındırır ve ağın geri kalanına bağlı olduğu sis sunucu ile bağlanır. Nesnelerin enerji profilleri ve barındırdıkları servisleri vardır. Bir nesne (T), $T_{id} = \{S_{Tid}, EP_{Tid}\}$ şeklinde tanımlanır. S_{Tid} T_{id} 'nin barındırdığı servislerin listesidir ve sn T_{id} 'nin barındırdığı servislerin sayısı olmak üzere $S_{Tid} = \{S_{Tid,1}, S_{Tid,2}, \dots S_{Tid,sn}\}$ şeklinde tanımlanır. Bir IoT servisi (s), $s_{id} = \{c_{sid}, EP_{sid}\}$ şeklinde tanımlanır. s_{id} 'nin enerji profili olan EP_{sid} tekil bir değerdir ve s_{id} kullanıldığında harcayacağı enerjiyi belirtir. Tasarlanan ağ yapısına dair örnek görsel Şekil 5.1'de verilmiştir.



Şekil 5.1. Tasarlanan ağ yapısına dair örnek görsel

Şekildeki ağ bir bulut sunucu, üç sis sunucu ve yedi nesneden oluşmaktadır. Nesneler farklı sayılarda IoT servisi barındırmaktadır. Resimde farklı işlevler sağlayan servisler farklı renklerde verilmiştir. Yapılan ağ modellemesinde ve literatürde, bir servisin sağladığı işlev sınıf olarak enkapsüle edilir. Dolayısıyla, resimdeki renk servislerin sınıfını ifade etmektedir. Resimdeki F2C bağlantı sis sunucular ile bulut sunucu arasındaki bağlantıyı, F2F bağlantı sis sunucuları arasındaki bağlantıyı ve T2F bağlantı ise nesneler ile sis sunucuları arasındaki bağlantıyı temsil etmektedir.

5.2.2. Uygulama modellemesi

Bir uygulama (APP_{id}) $APP_{id} = \{WF_{APP_{id}}, SLA_{APP_{id}}\}$ şeklinde tanımlanır. $WF_{APP_{id}}$ APP_{id} uygulaması birleştirilirken kullanılacak taslaktır, kullanılacak basit servislerin sınıflarını içerir. APP_{id} 'nin taslağı $WF_{APP_{id}}$, cn APP_{id} 'yi elde etmek için gereken basit servis sayısı

olmak üzere $WF_{APPid} = \{c_1, c_2, \dots, c_{cn}\}$ şeklinde tanımlanır. SLA_{APPid} ise uygulamanın uyması gereken QoS isterlerinin garanti altına alındığı kontrattır. IoT’de SC probleminin odak noktası uygulamayı oluşturacak servislerin uygulama taslağına belirtilen servis sınıfına sahip servislerden, oluşturulan uygulamanın QoS seviyesi SLA kontratına uygun olma şartını sağlayacak şekilde seçilmesidir.

5.2.3. QoS modellemesi

QoS metriği veya QoS isteri terimleri servis kalitesine dair uygulamaların talep ettiği özellikler için kullanılmaktadır. Literatürde ve sektörde Çizelge 2.2, Çizelge 2.3 ve Çizelge 2.4’de verdiğimiz üzere çok sayıda QoS metriği bulunmaktadır [16, 285]. Kritik QoS metriği uygulamanın türüne göre değişmektedir. Örneğin, bir video uygulaması için bant genişliği kritik QoS metriğidir. Bir çevre izleme uygulaması için ise ölçümlerin doğruluğu (accuracy) kritik QoS metriğidir. QoS metriklerinde çeşitlilik söz konudur. Çeşitliliğe ek olarak istenilen QoS metriklerin sayıları da değişkendir. Bir uygulama birden çok QoS metriği talep edebilir. Bu durumu daha kolay programlanabilir hale getirmek adına bu çalışma kapsamında QoS vektörlerinden yararlanılmıştır. QoS vektörleri çok sayıda farklı QoS metriğini tek bir veri yapısında tutarak önerilen yaklaşıma genelleyici bir arayüz sunar. Bir servisin (s_{id}) QoS vektörü ($QoSV_{sid}$), qn toplam QoS metriği sayısı olmak üzere $QoSV_{sid} = \{QoSA_{sid,1}, QoSA_{sid,2}, \dots, QoSA_{sid,qn}\}$ olarak tanımlanır.

QoS ister sayılarında genelleyicilik QoS vektörlerinin kullanımı ile sağlanmıştır. Benzer bir yaklaşım QoS isterleri için de gereklidir. IoT uygulamaları oldukça çeşitlidir ve bu çeşitlenme devam etmektedir. Özellikle uygulama katmanında tanımlanan QoS isterlerine gün geçtikçe yenileri eklenmektedir. Bu değişim ile başa çıkabilmek için QoS isterlerinin gruplanması adına genelleyici bir QoS modellemesi kullanılması gereklidir. Bu amaçla bu bölümde spesifik QoS isterleri tanımlamak ve kullanmaktansa, genelleyici bir QoS modellemesi kullanmayı tercih ettik. Bu çalışmada QoS isterleri [248, 266] ile paralel bir şekilde SC perspektiften sınıflandırılmıştır. Bunlar etki ve birleştirme operatörüdür.

Etki perspektifinden QoS metrikleri ikiye bölünmüştür, pozitif ve negatif QoS metrikleri.

- Pozitif QoS metriklerinin yüksek olması servisin kalitesinin de yüksek olduğunu gösterir. Dolayısıyla, pozitif QoS isterleri SC sürecinde maksimize edilmeye çalışılır. Bant genişliği ve güvenilirlik pozitif QoS isterlerine örnektir.

- Negatif QoS metriklerinin yüksek olması servis kalitesinin düşük olduğunu gösterir. Dolayısıyla, negatif QoS isterleri SC sürecinde minimize edilmeye çalışır. Hata oranı, ağ gecikmesi ve tepki süresi negatif QoS isterlerine örnektir.

QoS isterlerini gruplandıran ikinci perspektif birleştirme operatörüdür. Birleştirme operatörleri QoS isterlerini birleştirmek veya parçalamak için kullanılır. Literatürde kullanım alanı bir uygulamanın QoS seviyesini, uygulamayı oluşturan servislerin QoS seviyelerini kullanarak hesaplamaktır. Ancak yapılan çalışmada QoS vektörlerinin parçalanması için kullanılmıştır. Uygulamanın sahip olması beklenen QoS isterleri parçalanarak servis seçiminde kullanılmıştır. Birleştirme operatörüne göre QoS isterleri dört gruba ayrılır.

- Çarpma operatörü olasılıksal QoS isterlerinin birleştirilmesi için kullanılır. Bu metriklere örnek olarak uygunluk, hata oranı ve güvenilirlik verilebilir.
- Toplama operatörü eklemesi QoS isterlerinin birleştirilmesi için kullanılır. Servis zamanı, ağ gecikmesi ve tepki süresi bu gruba örnektir.
- Minimum operatörü bir QoS isterine ait çok sayıda değerden en küçüğünün önemli olduğu zamanlarda kullanılır. Bant genişliği buna örnektir.
- Maksimum operatörü bir QoS isterine ait çok sayıda değerden en büyüğünün önemli olduğu zamanlarda kullanılır.

İki perspektiften yapılan kategorizasyon sonucu toplam sekiz QoS ister sınıfı elde edilmiştir. Elde edilen sınıflar literatüre [248, 266] benzerdir ve SC perspektifinden olası tüm QoS metriklerini kapsaması beklenmektedir. Burada belirtilmelidir ki, SC süreci ağı uygulama katmanında gerçekleşmektedir. Bu sebeple kapsanan uygulama katmanında tanımlanabilecek tüm QoS metriklerinin kapsanacağıdır. Ancak ağ katmanına yapılacak bir uygulamada, ağ katmanının spesifik QoS isterlerinin desteklenmesi adına eklemeler yapılması gerekebilir.

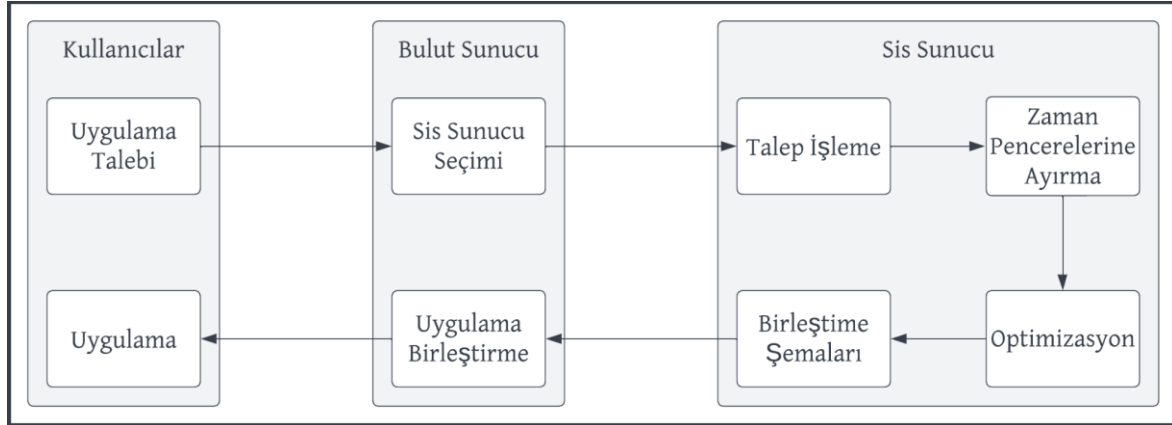
Önerilen yaklaşımın test edilmesi için her kategorideki QoS isterlerinden birer tane oluşturulmuştur. Ağdaki her servis bu sekiz kategoride oluşturulan QoS isterlerini belirli seviyelerde sağlar. Uygulamalar ise, test senaryosunda belirtilen sayıda QoS isterini SLA kontratı kapsamında belirli seviyelerde talep eder. Çizelge 5.3’de tanımlanan QoS isterleri verilmiştir.

Çizelge 5.3. Geliştirilen QoS modellemesinde tanımlanan QoS ister tipleri

QoS İsteri	Etki	Birleştirme Operatörü
Type 1 (T1)	Pozitif	Çarpma
Type 2 (T2)	Negatif	Çarpma
Type 3 (T3)	Pozitif	Toplama
Type 4 (T4)	Negatif	Toplama
Type 5 (T5)	Pozitif	Minimum
Type 6 (T6)	Negatif	Minimum
Type 7 (T7)	Pozitif	Maksimum
Type 8 (T8)	Negatif	Maksimum

5.3. Önerilen Yaklaşım

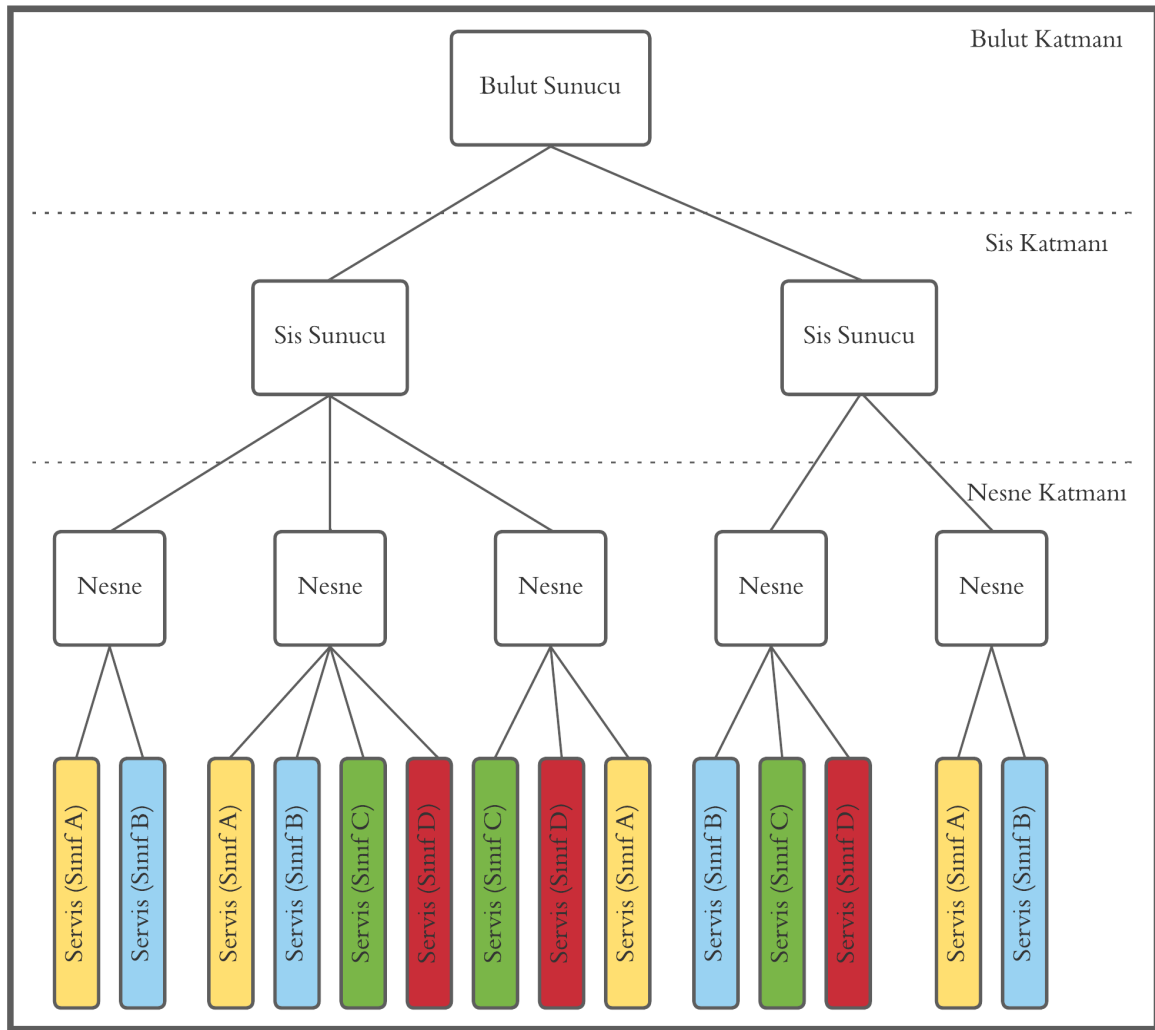
Önerilen yaklaşım kullanıcılardan uygulama taleplerini toplar, uygulamaları QoS gereksinimlerine uygun olarak, ağıın enerji tüketimi ve iş yükünün IoT servisleri arasında dengeli dağıtılmasını gözeterek uygulamaları oluşturur ve son adımda kullanıcılara sunar. Bu yaklaşım uyarınca bir bulut sunucu ve çok sayıda sis sunucu kullanılır. Bulut sunucu uygulama taleplerini toplamaktan ve sis düğümlerine dağıtmaktan sorumludur. Bir uygulama bulut sunucuya geldiğinde talebi seçtiği bir sis sunucuya iletir. Uygulama talebini alan sis sunucu öncelikle talebi işler, servis taleplerini zaman düzlemine haritalar ve zaman pencerelerine böler. Her zaman penceresi kendi içerisinde optimize edilir. Ardından birleştirme şemaları bulut sunucuya gönderilir. Kompozisyon şemalarına göre uygulamalar birleştirilir ve son kullanıcıya sunulur. Anlatılan iş akışı Şekil 5.2’de verilmiştir. Uygulama taleplerinin sis sunuculara dağıtılmasını eşitlikçi bir yaklaşım izlenir.



Şekil 5.2. Önerilen yaklaşım

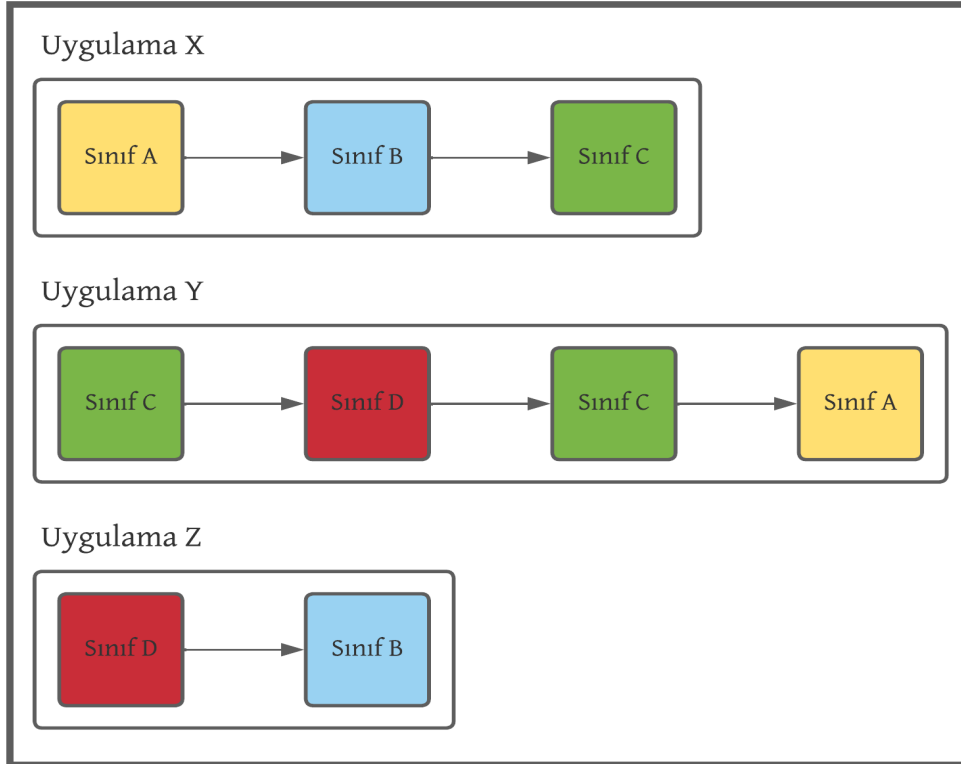
5.3.1. Servis birleştirme süreci

Bu alt bölümde servis birleştirme süreci bir senaryo üzerinden görsellerle desteklenerek anlatılacaktır. Bir bulut sunucu, iki sis sunucu, beş nesneden oluşan bir ağ Şekil 5.3'te verilmiştir. Ağdaki nesnelere değişken sayıda ve sınıfa ait servisler bağlıdır. Toplam olarak dört çeşit servis sınıfı tanımlanmıştır. Resimde daha anlaşılabilir olması adına servis sınıfları renklerle kodlanmıştır.



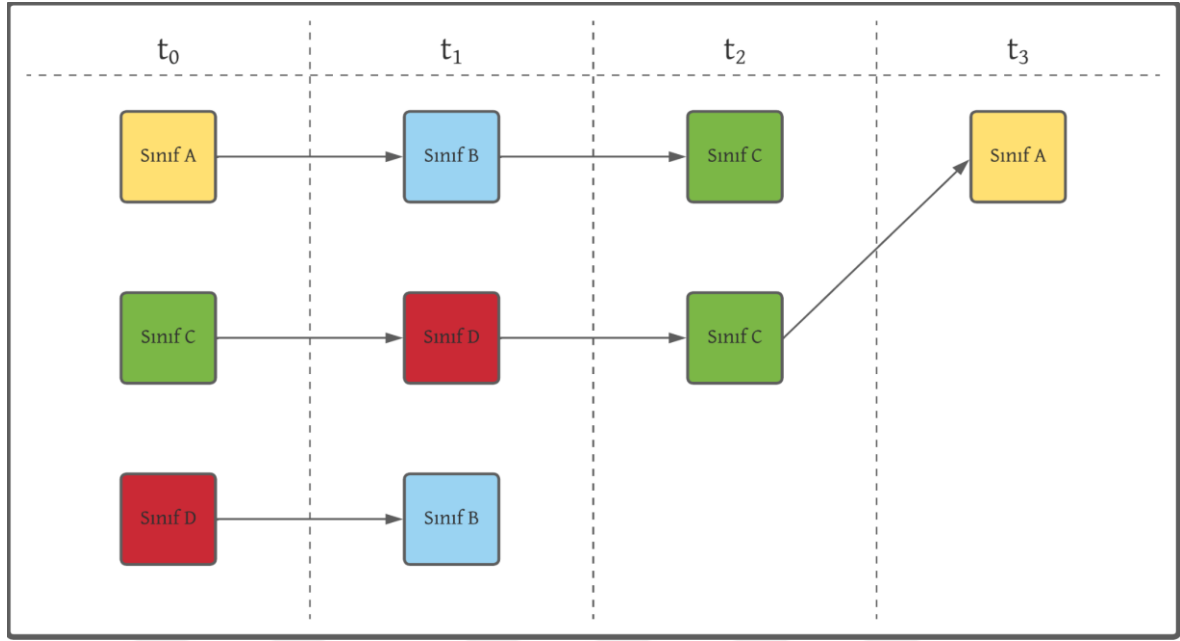
Şekil 5.3. SC süreci görsel örnek I

Bu ağdan üç adet uygulama talep edildiğini kabul edelim. Bu uygulamaların taslakları (WF) Şekil 5.4’te verilmiştir. Resimde verilen servis sınıflarına sahip servisler ağdan seçilip uygulamayı birleştirmek için kullanıldığı takdirde uygulamanın talep ettiği işlev sağlanacaktır.



Şekil 5.4. SC süreci görsel örnek II

Uygulamalarda kullanılan servislerin hepsi birer zaman dilimi almaktadır ve art arda sıralanmıştır. Üç uygulamanın da t_0 zaman diliminde çalışmaya başlaması beklendiği takdirde uygulama talepleri Şekil 5.5'te verildiği gibi zaman pencerelerine bölünür. Resimde görüldüğü üzere t_0 zaman dilimi için A,C,D sınıflarından 3 servise, t_1 zaman dilimi için B sınıfından iki D sınıfından bir servise, t_2 zaman dilimi için C sınıfından iki servise ve t_3 zaman dilimi için A sınıfından bir servise ihtiyaç vardır.



Şekil 5.5. SC süreci görsel örnek III

Zaman dilimlerinde talep edilen servislerin ağdan servislerle eşleştirilmesi her zaman dilimi için ayrı ayrı optimize edilir. Bu süreçte, uygulamanın QoS isterleri, ağın enerji tüketimi ve IoT servisleri arasında adil iş yükü dağılımı göz önüne alınır. Örnek bir eşleştirme Şekil 5.6'da verilmiştir.

Ağıdaki Servisler	Talep Edilen Servisler									
	Servis (Sınıf A)	t_0 Servis (Sınıf C)	Servis (Sınıf D)	Servis (Sınıf B)	t_1 Servis (Sınıf D)	Servis (Sınıf B)	t_2 Servis (Sınıf C)	Servis (Sınıf C)	t_3 Servis (Sınıf A)	
Servis (Sınıf A)										
Servis (Sınıf B)										
Servis (Sınıf A)	X									
Servis (Sınıf B)				X						
Servis (Sınıf C)							X			
Servis (Sınıf D)										
Servis (Sınıf C)		X						X		
Servis (Sınıf D)					X					
Servis (Sınıf A)									X	
Servis (Sınıf B)						X				
Servis (Sınıf C)										
Servis (Sınıf D)			X							
Servis (Sınıf A)										
Servis (Sınıf B)										

Şekil 5.6. SC süreci görsel örnek IV

Önerilen yaklaşımda sis sunucular uygulama taleplerini işler, optimize birleştirme şemaları elde eder ve şemaları bulut sunucuya iletir. Bir grup uygulama talebi geldiğinde yapılan ilk iş taleplerin işlenmesidir. Bu sürecin amacı, her zaman penceresinde ihtiyaç duyulan IoT servis sayılarını ve sınıflarını ortaya çıkarmaktır. İhtiyaç duyulan servisler zaman düzlemine yayıldıktan sonra talep edilen servisler sis düğümüne bağlı servislerle eşleştirilir. Bu eşleştirme sonucu birleştirme şemaları elde edilir. Ardından birleştirme şemaları bulut sunucuya gönderilir ve uygulamalar bulut sunucu tarafından şemaya göre birleştirilir. Bu eşleştirme rastgele bir şekilde yapılırsa talep edilen işlevi yerine getirir. Ancak QoS gereksinimlerini ve tanımlanan diğer objektifleri yerine getiremez. Bu sebeple, birleştirme şemalarının optimize edilmesi gereklidir. Bu çalışma kapsamında servis eşleştirmelerinin ve dolayısıyla birleştirme şemalarının elde edilmesinde çok hedefli bir optimizasyon algoritması olan NSGA-II kullanılmıştır [286].

5.3.2. Çok objektifli optimizasyon

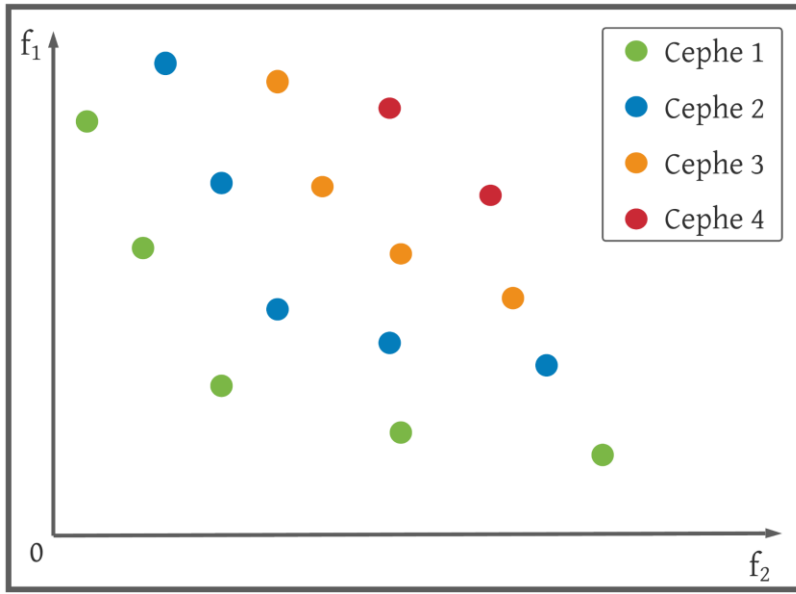
Çok hedefli optimizasyon problemlerinin çözümünde iki yaklaşım kullanılır. Bunlardan ilki çok sayıda hedefin SAW tekniği ile toplanarak tek bir değer haline getirilmesi ve optimizasyon algoritmasının tek bir uygunluk fonksiyonunu optimize etmeye çalışmasıdır. Bu yaklaşımda çok hedefli optimizasyon problemleri tek hedefli optimizasyon algoritmaları ile çözülebilir. Uygulaması oldukça basit olan bu yaklaşımın bir takım dezavantajları bulunmaktadır. Öncelikle, objektiflerin ağırlıklandırılması süreci algoritmanın başarısını çok etkiler. Ek olarak, kullanıcı taleplerinde değişim olduğunda hedefler yeniden ağırlıklandırılmalı ve algoritma yeniden çalıştırılmalıdır. İkinci yaklaşım çok hedefli optimizasyon yaklaşımıdır. Çok objektifli optimizasyon algoritmalarında bir çözümü değerlendirmek için objektif sayısı kadar uygunluk fonksiyonu bulunur. Toplam objektif sayısı n , f_x x numaralı objektife göre puanlama yapan uygunluk fonksiyonu ve S çözüm uzayı olmak üzere birçok objektifli optimizasyon problemi (F) aşağıda verildiği gibi ifade edilebilir:

$$\text{Minimize } F(s) = (f_1(s), f_1(s) \dots f_n(s)), s \in S \quad (5.1)$$

Çok hedefli optimizasyon algoritmaları çözümleri çok sayıda uygunluk fonksiyonu ile puanlandırır. Bu her çözümün çok sayıda skoru olmasına neden olur. En iyi çözümlerin tespit edilebilmesi için çok sayıda skora sahip çözümlerin karşılaştırılabilmesi ve sıralanması gereklidir. Çok skorlu çözümlerin sıralanması için optimizasyon algoritmaları çoğunlukla pareto hakimiyet kavramını kullanır. Pareto hakimiyete göre, bir çözüm (s_1) bir diğer çözümü (s_2) iki şartı sağlıyorsa domine eder.

- s_1 s_2 'den en az bir hedefte daha iyidir.
- s_1 s_2 'den hiçbir objektifte daha kötü değildir.

Puanlanmış bir sete pareto hakimiyet uygulanırken tüm çözümler birbiri ile karşılaştırılır. Ardından çözümler domine etme ve edilme seviyelerine göre kümelenir. Bu kümelere Pareto Cephe denir. Pareto hakimiyet uygulamasından sonra çözümler pareto cephelere bölünmüş şekilde elde edilir. İki objektifli bir problemin çözümlerinin pareto cephelere bölünmüş hali Şekil 5.7'de verilmiştir.



Şekil 5.7. Pareto cepheler

En iyi cephe optimizasyon algoritmasının çıktısı olarak kullanılır. Tek bir en iyi çözüm veren tek hedefli optimizasyon algoritmalarının aksine çok objektifli optimizasyon algoritmalarının sonucunda çok sayıda çözüm içeren bir pareto cephe elde edilir. Kullanıcılar, tercihlerine göre pareto cepheden istedikleri çözümü seçer ve kullanır.

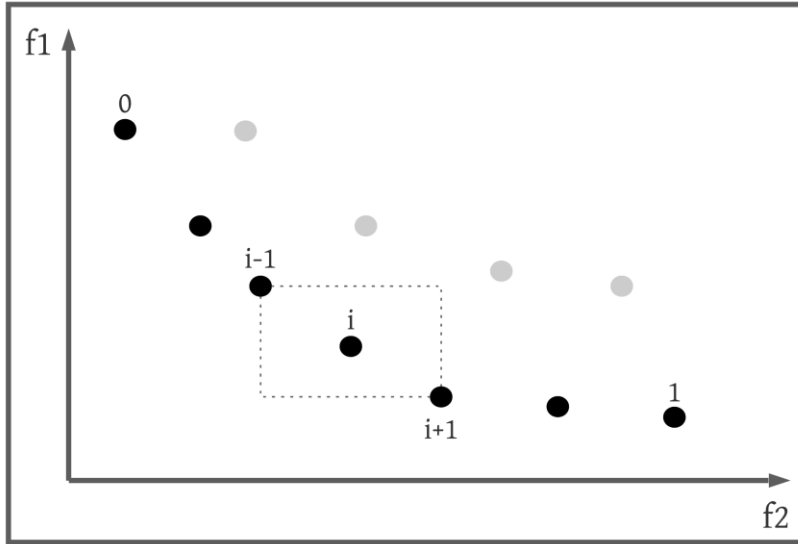
5.3.3. NSGA-II

Deb ve arkadaşları tarafından sunulan NSGA-II [286] çok hedefli optimizasyon problemleri için kullanılan evrimsel temelli bir algoritmadır. NSGA-II iki ana prensibe dayanır. Bunlar popülasyonun hızlı sıralanması ve çeşitliliğin korunmasıdır [287]. Bu prensiplerden ilki için hızlı domine edilmeyen sıralama algoritması (fast non-dominated sorting algorithm / FNDSA), ikincisi için ise kalabalık mesafesi (crowding distance) kullanılmıştır.

FNDSA, pareto hakimiyet yaklaşımı ile çözümleri sıralayan bir algoritmadır. İki aşamadan oluşur. İlk aşamada tüm çözümler birbiri ile karşılaştırılır. Ardından her çözüm için çözümün kaç tane diğer çözümü domine ettiği ve kaç tane çözüm tarafından domine edildiğini hesaplanır. Sonra algoritmanın ikinci aşamasına geçilir. Bu aşamada iteratif bir yaklaşımla çözüm kümesi pareto cephelere ayrılır [288]. Öncelikle, hiç bir başka çözüm tarafından domine edilmeyen çözümler ilk pareto cephede toplanır. Ardından bu çözümler popülasyondan çıkarılır. Ardından popülasyonda kalan çözümlerden hiçbir popülasyon üyesi çözüm tarafından domine edilmeyen çözümler yeni bir pareto cephe olarak tanımlanır

ve çözüm popülasyonundan çıkarılır. Yeni oluşturulan cephe iki numaralı cephedir. Bu şekilde popülasyonda çözüm kalmayana kadar bu döngü devam eder.

Pareto cepheler elde edildikten sonra cephe içlerinde sıralama için kalabalık mesafeleri hesaplanır. Kalabalık mesafeleri çözümleri en yakın diğer çözümlere olan mesafelerine göre en ıssız konumlardaki çözümlere yüksek puan vererek sıralar. Öncelikle sınır çözümlere sonsuz mesafe atanır. Diğer çözümler iki yanlarındaki çözümlere uzaklıklarına göre puanlandırılır. Bu aşamada objektif skorlarının istatistiki karakterleri sonuca etki etmesin diye, skorlar normalize edilip kullanılır. Örnek bir hesaplama i numaralı çözüm için Şekil 5.8’de verilmiştir.



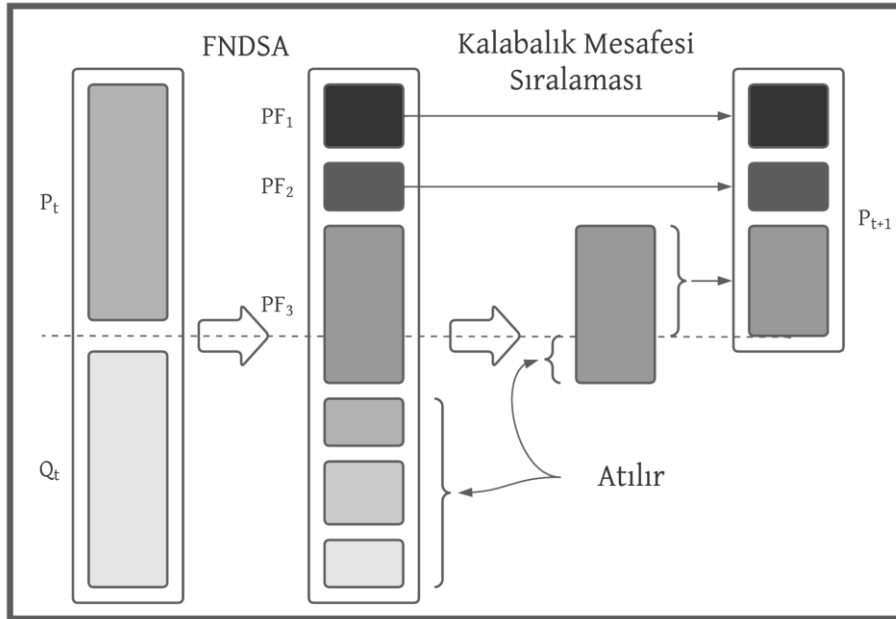
Şekil 5.8. Kalabalıklık mesafesi hesaplanması

NSGA-II çok objektifli problem için kullanılan en popüler evrimsel algoritmalarından biridir. Bunun sebebi basitliği ve etkinliğidir [289]. Doğanın evrim mekaniklerini kullanan NSGA-II rastgele çözümlerden oluşan bir popülasyon oluşturur. Ardından genetik operatörlerin yardımı ile popülasyonu daha iyi çözümlerden oluşan bir hale getirir. NSGA-II algoritmasının adımları aşağıda verilmiştir.

1. Rastgele çözümlerden oluşan bir popülasyon oluştur.
2. Objektiflere dair uygunluk fonksiyonlarını kullanarak popülasyondaki çözümleri puanlandır.
3. FNDSA ile popülasyonu sırala.
4. Çözümler için kalabalık mesafesini ölç ve cepheleri kalabalık mesafesine göre sırala.

5. Popülasyonu ikiye böl ve daha iyi olan kısmı yeni çözümler üretmek için kullanılacak ebeveyn popülasyon olarak etiketle. Kalan çözümleri at.
6. Yeni bireyler üretmek için ebeveyn popülasyonundan turnuva seçimi ile ebeveyn seç.
7. Çaprazlama operasyonu ile yeni bireylerden oluşan bir popülasyon oluştur.
8. Ebeveyn ve yeni birey popülasyonunu birleştir.
9. Objektiflere dair uygunluk fonksiyonlarını kullanarak yeni popülasyondaki çözümleri puanlandır.
10. FNDSA ile yeni popülasyonu sırala.
11. Çözümler için kalabalık mesafesini ölç ve cepheleri kalabalık mesafesine göre sırala.
12. Algoritmanın sona erme kriterini kontrol et. Eğer karşılanmadıysa Adım 5'e ilerle. Aksi takdirde Adım 13'e ilerle.
13. En iyi pareto cepheyi döndür.

NSGA-II algoritmasının yeni popülasyon üretme prosedürü Şekil 5.9'da verilmiştir.



Şekil 5.9. NSGA-II yeni popülasyon üretme prosedürü

Şekilde $t+1$ 'inci neslin ebeveyn popülasyonu (P_{t+1}), t 'inci neslin ebeveyn (P_t) ve yeni birey (Q_t) popülasyonlarından elde edilmesi gösterilmektedir. P_t ve Q_t birleştirilir ve tek bir popülasyon haline getirilir. Ardından popülasyon FNDSA ile cephelere bölünür, cephe içlerinde kalabalık mesafesine göre sıralama yapılır. Ardından popülasyonun yüksek başarılı

bireylerden oluşan yarısı bir sonraki neslin ($t+1$) ebeveyn popülasyonu (P_{t+1}) olarak etiketlenir.

5.3.4. Problemin optimizasyon formülasyonu

Bir eşleştirme, uygulama tarafından talep eden bir servis ile ağda bulunan bir IoT servisin eşleştirir. Bu eşleştirmede esas, talep edilen servis sınıfı ile IoT servis sınıfının aynı olmasıdır. Eşleştirmeler sis düğümleri tarafından yapılır. Bir eşleştirme (M), c talep edilen servis sınıfı ve s c sınıfına dahil ağdaki bir IoT servisi olmak üzere $M = \{c, s\}$ şeklinde tanımlanır. Bir çözüm (S) ise, bir zaman dilimindeki tüm eşleşmeler olarak toplam talep edilen servis sayısı ns olmak üzere $S = \{M_0, M_1, \dots, M_{ns}\}$ şeklinde tanımlanır. NSGA-II algoritması çözümleri 3 objektife göre optimize eder. Bunlar QoS, enerji etkinliği ve adil iş yükü dağılımıdır.

İlk objektif QoS hedef (f_{QoS})'dir. f_{QoS} objektifinin hedefi uyulmayan SLA kontratı sayılarını minimize etmektir. Bu amaçla, uygulamanın sahip olması gereken SLA kontratı IoT servis seçiminde kullanılacak alt-SLA (SSLA) kontratlarına kırılır. Bir SLA kontratı (SLA) $Cond_{id,x}$ SLA_{id}'nin x 'inci isteri ve cn toplam QoS isteri sayısı olmak üzere $SLA_{id} = \{Cond_{id,1}, Cond_{id,2}, \dots, Cond_{id,cn}\}$ olarak tanımlanabilir. Bir ister olan $Cond_{id}$ ise $Cond_{id} = \{QoSA, val\}$ şeklinde tanımlanabilir. Tanımdaki QoSA isterin konusu olan QoS metriğini, val ise QoS isterinin hangi değerde sağlanması gerektiğini ifade eder. Birleştirme operatörlerine göre SLA kontratları alt servis sayısı kadar SSLA kontratlarına kırılır. SSLA, SLA ile aynı yapıdadır atası olan SLA ile aynı sayıda istere, aynı QoS metriklerine sahiptir. Farkı isterlerin (Cond) içindeki val değerleridir. SLA'deki val değerleri birleştirme operatörleri ile SSLA'lerin val değerlerine çevrilir. Bu çevrim, eğer servis seçiminde SSLA'lere uygun olunursa uygulamanın QoS seviyesinin de SLA'e uygun olacağını garanti edecek şekilde deterministik karakterdedir. SSLA kontratlarının elde edilmesinden sonra, çözümler (S) eşleşmelerin (M) SSLA kontratlarına uygun olup olmadığını kontrol eder. Uygun olmayan SSLA sayısı çözümün QoS skoru (f_{QoS}) olarak kullanılır.

İkinci hedef enerji tüketimi skorudur (f_E). f_E bir çözümde kullanılan tüm IoT servislerinin kullanım enerjilerinin toplamı olarak tanımlanmıştır.

Üçüncü hedef iş yükü dağılım objektifidir (f_F). Bu çalışma kapsamında servislerin art arda kullanılma sayıları f_F olarak kullanılmıştır. Objektifin amacı iş yükünü aynı sınıftaki IoT

servisleri arasında etkin bir şekilde dağıtmaktır. Servislerin son n zaman dilimindeki kullanımları kayan pencere yaklaşımı ile saklanmıştır.

NSGA-II algoritması mevzu bahis üç hedefi (f_{QoS} , f_E , f_F) minimize etmeye çalışır.

5.3.5. Genetik operatörler

NSGA-II algoritması üç genetik operasyon kullanır. Bunlar aşağıda verilmiştir.

- Seçim operasyonu gelecek nesli üretmek için kullanılacak ebeveyn çözümlerin seçim işlemidir. Bu operasyon için turnuva seçimi (tournament selection) yaklaşımı[286]'de belirtildiği gibi kullanılmıştır.
- Mutasyon operasyonu popülasyonda tekillleşmenin önüne geçmek için kullanılır. Ebeveyn seçiminden sonra her ebeveyne mutasyon süreci yürütülür. $[0, 1]$ aralığında bir değer rastgele seçilir. Eğer değer NSGA-II algoritmasının bir parametresi olan mutasyon şansından düşükse ebeveyn çözüme mutasyon uygulanır. Aksi takdirde uygulanmaz. Mutasyon uygulanması, çözüm içerisindeki bir seçili IoT servisinin aynı sınıftan ve uygun olan başka bir IoT servisi ile değiştirilmesi olarak tanımlanmıştır.
- Çaprazlama operasyonu seçilen ebeveynlerden yeni birey üretme sürecidir. Ebeveynlerin çözüm (S) listeleri rastgele bir yerden kesilir ve çapraz şekilde birbirine eklenir. Bu sayede iki ebeveynden iki yeni birey elde edilir.

5.3.6. Değerlendirme

Özellikle çok objektifli optimizasyon problemlerinde çözüm uzayı oldukça geniştir. Böyle durumlar hızlı bir şekilde çözüme yakınsama ve çeşitlilik koruma yetenekleri ile NSGA-II kullanımı oldukça uygundur. Ek olarak, NSGA-II algoritması çok hedefi aynı anda optimize edebilme yeteneğine sahiptir. Bu sebeplerle, IoT'de SC problemi için NSGA-II temelli optimizasyon modelleri kullanılmıştır.

5.4. Test Sonuçları

Önerilen yaklaşımın etkinliğinin gösterilmesi adına simülasyonlar yapılmıştır. IoT'de SC problemi için kullanılabilecek simülatör/emülatör eksikliğinden dolayı basit bir simülatör Python 3.8. ile kodlanmıştır. Simülatör ayrıca bir ağ yapısı ve uygulama talepleri akışını

belirtilen parametrelere göre ve Bölüm 5.3’de verilen ağ modeline sadık kalacak şekilde üretir. Üretilmiş veri kullanılmasının sebebi bu alanda kullanılabilecek IoT veri setinin bulunmamasıdır. Simülasyon parametreleri Çizelge 5.4’de verilmiştir. Nesnelerin enerji profilleri [10, 34] çalışmalarına dayanarak yapılmıştır.

Çizelge 5.4. Simülasyon parametreleri

Parametre	Değer
Sis sunucu sayısı	5 Adet
Sis sunucu başına nesne sayısı	6 Adet
Nesne başına servis sayısı	3 Adet
Servis sınıfı sayısı	4 Adet
Uygulama uzunlukları	[2, 3] Adet
Her uygulama için kritik QoS metriği sayısı	[1, 2, 3] Adet
IoT servis enerji tüketimi	[100:10000] mAs (miliamper/saat)
Simülasyon süresi	50 Zaman Dilimi
Uygulama talep üretim süresi	40 Zaman Dilimi
Her zaman diliminde üretilen uygulama talep sayısı	[9, 10, 11] Adet
Olasılıksal QoS metriği değer aralığı	[0.9:1.0] Kayan Noktalı Değer
Olasılıksal olmayan QoS metriği değer aralığı	[1:10] Tamsayı Değer

Simülasyon senaryolarında üç çeşit SLA kontratı üretilmektedir.

- Yüksek SLA, yüksek servis kalitesi bekleyen uygulamaların kontratıdır. QoS isterinin olası minimum maksimum değer menziline %75’inden daha iyi değer talep eder.
- Orta SLA, ortalama servis kalitesi bekleyen uygulamaların kontratıdır. QoS isterinin olası minimum maksimum değer menziline %50’sinden daha iyi değer talep eder.
- Düşük SLA, düşük servis kalitesi bekleyen uygulamaların kontratıdır. QoS isterinin olası minimum maksimum değer menziline %25’inden daha iyi değer talep eder.

Uygulamaların üretilmesi aşamasında yüksek, ortalama ve düşük SLA kontratları eşit oranda üretilir. Geliştirilen simülatör verilen parametrelere göre bir test senaryosu oluşturur ve bu

senaryoyu dört optimizasyon modeli kullanarak çözer. Geliştirilen optimizasyon modelleri aşağıda verilmiştir.

- GA (Genetik Algoritma temelli Model), ilk modeldir. Sadece f_{QoS} objektifini optimize eder. GA'nın kullanılmasının sebebi konvansiyonel sadece QoS objektifine sahip sezgisel yaklaşımları temsil etmesidir.
- NSGA-II QE f_{QoS} ve f_E objektiflerine sahip modeldir. Gereği sağlanan SLA kontratlarını maksimize ederken enerji tüketimini minimize etmeye çalışır. Bu model adil iş yükü objektifini yok sayar literatürdeki güncel QoS ve enerji farkında yaklaşımları temsil eder.
- NSGA-II QF f_{QoS} ve f_F objektiflerine sahip modeldir. Gereği sağlanan SLA kontratlarını maksimize ederken aynı IoT servislerin art arda kullanımlarını minimize etmeye çalışır.
- NSGA-II QEF üç objektife (f_{QoS} , f_E , f_F) de sahip modeldir. Gereği sağlanan SLA kontratlarını maksimize ederken enerji tüketimini ve aynı IoT servislerin ardarda kullanımlarını minimize etmeye çalışır.

Bu çalışma kapsamında önerilen yaklaşım NSGA-II QEF modelinin IoT'de SC süreçlerinde QoS kısıtları altında enerji etkin ve adil birleştirme şemaları elde edilmesi için önerilmiştir. Diğer 3 model (GA, NSGA-II QE ve NSGA-II QF) NSGA-II QEF modelinin etkinliğinin gösterilmesi için kullanılmıştır. Ek olarak, NSGA-II QE ve NSGA-II QF modelleri enerji tüketimi ile adil iş yükü arasındaki takası ortaya çıkarmak için kullanılmıştır.

Optimizasyon modellerinin parametreleri ayrı ayrı optimize edilmiştir. Optimize edilen parametreler genetik tabanlı evrimsel algoritmaların çoğunda ortak olan popülasyon büyüklüğü, mutasyon şansı ve nesil sayısıdır. Her model ayrı ayrı ızgara arama (grid search) metodu ile optimize edilmiştir. Popülasyon sayısı için $\langle 50, 75, 100, 150, 200 \rangle$, mutasyon şansı için $\langle 0.1, 0.2, 0.3 \rangle$ değerleri test edilmiştir. Her parametre kombinasyonu rastgeleliğin etkisini azaltmak adına 15 kere koşturulmuş ve 15 koşturmanın ortalaması en iyi parametre kombinasyonunun elde edilmesi için kullanılmıştır. Elde edilen parametre kombinasyonları Çizelge 5.5'de verilmiştir.

Çizelge 5.5. Optimal parametre konfigürasyonları

Model	Nesil Sayısı	Popülasyon Büyüklüğü	Mutasyon Şansı
GA	200	50	0.3
NSGA-II QE	150	50	0.1
NSGA-II QF	150	50	0.2
NSGA-II QEF	150	75	0.2

Önerilen yaklaşım çift 2.20 GHz frekansa sahip Intel Xeon Silver 4114 işlemciye ve 64 GB belleğe sahip Linux Mint 19.04 işletim sistemi kullanan bir sunucuda test edilmiştir. Rastgeleliğin etkisinin azaltılması adına on farklı test senaryosu oluşturulmuş ve her senaryo beş kere çözülmüştür. Sonuç olarak verilen değerler 50 koşturmanın ortalamasıdır. Test sonuçları Çizelge 5.6’da verilmiştir.

Çizelge 5.6. Test sonuçları

Model	Sağlanan SLA kontratlar sayıları	Enerji tüketimi (mAs)	Ardarda kullanım sayıları			
			n=2	n=3	n=4	n=5
Rastgele	109,08	3 663 461,64	112,22	16,08	2,28	0,46
GA	148,36	3 691 745,16	118,92	19,06	3,48	0,8
NSGA-II QE	146,54	2 338 050,12	208,84	72,5	30,08	13,18
NSGA-II QE	144,28	3 684 572,78	155,7	7,58	0,44	0,06
NSGA-II QE	142,62	2 949 526,90	151,32	18,14	1,98	0,12

5.4.1. Sonuçların QoS perspektifinden değerlendirilmesi

Test senaryosunda ortalama uygulama sayısı 300’dir. Çizelge 5.6’da her optimizasyon modeli için QoS seviyesi sağlanabilen uygulama sayısı verilmiştir. Görüldüğü üzere dört model de QoS objektifinde benzer performans göstermiş ve %50 başarı oranına yaklaşık değerler almışlardır. QoS objektifinin daha iyi incelenebilmesi adına farklı sayıda kritik QoS metriğine sahip senaryolar ile testler yapılmıştır. Bu testler kapsamında sadece kritik QoS

sayısı değiştirilmiş, diğer test parametreleri Çizelge 5.4'deki ile aynı bırakılmıştır. Test sonuçları Çizelge 5.7'de verilmiştir.

Çizelge 5.7. Farklı kritik QoS metrik sayılarında elde edilen sonuçlar

Kritik QoS Metrik Sayısı	Modeller				Uygulama Sayısı
	GA	NSGA-II QE	NSGA-II QF	NSGA-II QEF	
[1,1]	214	209	207	207	300
[1,2]	176	174	174	170	300
[1,3]	155	154	151	151	300
[1,4]	132	129	129	126	300
[1,5]	126	125	117	124	300
[1,6]	110	109	107	106	300
[1,7]	97	97	95	94	300

Çizelgede görüldüğü gibi, kritik QoS metrik sayısı arttıkça talep edilen QoS seviyesinin sağlanması istikrarlı bir şekilde zorlaşmaktadır. Çizelge 5.6'ya benzer bir şekil aynı kritik QoS sayılarında modeller yakın değerlerde kümelenmektedir. Bu durum önerilen çok objektifli yaklaşımın QoS kısıtları altında minimal bir QoS seviyesi azalması ile ek objektifleri optimize edebildiğini göstermektedir.

5.4.2. Sonuçların enerji etkinliği ve adil iş yükü dağılımı perspektifinden değerlendirilmesi

Beklenildiği üzere NSGA-II QE en düşük enerji tüketimine sahip modeldir. NSGA-II QF ve GA modellerinde enerji etkinliğine dair bir objektif bulunmadığından dolayı enerji tüketimleri yüksektir. NSGA-II QEF modeli ise arada durmaktadır. Çizelge 5.6'dan görüldüğü üzere NSGA-II QEF modeli GA ve NSGA-II QF modellerine kıyasla %20 daha düşük enerji tüketimine sahiptir ancak NSGA-II QE modeline kıyasla %26 daha fazla enerji tüketir.

Adil iş yükü dağılımı objektifinin değerlendirmesi IoT servislerinin art arda kullanılması üzerinden yapılmıştır. Art arda [2, 3, 4, 5] kere kullanılan IoT servislerinin sayıları Çizelge

5.6'da verilmiştir. Görüldüğü üzere NSGA-II QE adil iş yükünü bozmaktadır ve IoT servislerinin art arda kullanılmasına sebep olmaktadır. Bu durumun arkasındaki sebep enerji tüketimi ve adil iş yükü dağılımı objektifleri arasındaki takastır. Enerji tüketimi objektifi ağı düşük enerji profiline sahip IoT servisleri tekrar tekrar kullanmaya iter. Bu durum enerji tüketimini düşürse de, iş yükündeki adil dağılımı bozar. Mevzu bahis takas Çizelge 5.6'da açıkça görülebilmektedir. Bu takası daha iyi irdelemek adına GA ve NSGA-II QE modellerinin sonuçlarına bakalım. NSGA-II QE modeli GA modeline benzerdir, tek farkı enerji tüketimi hedefine de sahip olmasıdır. Ayrıca iki modelde de tanımlanmış bir adil iş yükü dağılımı objektifi yoktur. Bu sebeplerden dolayı bu iki modelin sonuçları incelenirse, enerji tüketimi objektifinin etkisi çok daha açık bir şekilde görünecektir. GA modeline kıyasla NSGA-II QE modeli art arda [2, 3, 4, 5] kere kullanılan IoT servis sayılarını [%76, %280, %764, %1547] oranında arttırmaktadır. Bu sonuçlar adil iş yükü dağılımının IoT SC sürecine dahil edilmesi gereğini ortaya koymaktadır.

Çizelge 5.6'da görüldüğü üzere NSGA-II QEF modeli QoS, enerji tüketimi ve adil iş yükü objektiflerini aynı anda ve başarılı bir şekilde optimize edebilmiştir. Adil iş yükü dağılımı ve enerji tüketimi objektifleri arasındaki takasa rağmen NSGA-II QEF modeli art arda kullanımları GA ve NSGA-II QE modellerine göre etkin bir şekilde azaltabilmiştir. Model [3, 4, 5] kerelik art arda kullanımları GA modeline kıyasla [%5, %43, %85] oranlarında azaltmıştır. NSGA-II QE modeline kıyasla ise, NSGA-II QEF modeli [2, 3, 4, 5] kere art arda kullanımları [%28, %75, %93, %99] oranlarında azaltmıştır. Bu duruma tek istisna NSGA-II QEF ve GA modelleri iki art arda kullanım sayısında karşılaştırıldığında ortaya çıkmaktadır. Benzer bir durum NSGA-II QF ile GA karşılaştırıldığında, aynı şekilde iki kere art arda kullanımda görülmektedir. Bu fenomenin arkasındaki sebep net olarak bilinmemektedir. Ancak burada göz önünde tutulmalıdır ki, adil iş yükü açısından bakıldığında art arda kullanım sayısı arttıkça durumun vehameti de artmaktadır. Art arda 5 kullanım, 2 kullanıma göre çok daha kötü bir durumdur. NSGA-II QEF modeli, yüksek art arda kullanım sayılarını etkin bir şekilde engellemektedir.

5.4.3. Çalışma zamanları karşılaştırılması

Modellerin çalışma zamanlarına dair karşılaştırma Çizelge 5.8'de verilmiştir.

Çizelge 5.8. Modellerin çalışma zamanları

Model	Çalışma Zamanları (Saniye)	
	Zaman Penceresi	Toplam
GA	0,29	14,78
NSGA-II QE	1,35	67,68
NSGA-II QF	1,65	82,64
NSGA-II QEF	2,99	149,85

Çizelgede görüldüğü üzere en düşük çalışma zamanına sahip model GA modelidir. Tek objektifli bir optimizasyon algoritması olduğu için GA modelinin NSGA-II temelli modellere göre çok daha düşük hesaplama karmaşıklığı vardır. NSGA-II temelli modeller arasında NSGA-II QE modeli en düşük çalışma zamanına sahiptir. NSGA-II QE modelinin çalışma zamanı NSGA-II QF modeline göre %18.19 daha düşüktür. Bu durum bize adil iş yükü objektifinin hesaplama karmaşıklığını göstermektedir. Bahsedildiği üzere sis düğümleri kendisine bağlı her IoT servisinin kullanım sayılarını bir kayan pencere yaklaşımı ile tutar ve adil iş yükü objektifini bu veri üzerinden hesaplanır. Bu ekstra hesaplama yükü görüldüğü üzere adil iş yükü objektifini kullanan modellerin çalışma zamanını arttırmaktadır. NSGA-II QEF modelinin çalışma zamanı ise diğer modellere kıyasla oldukça yüksektir. Bunun arkasındaki sebebin enerji tüketimi ile adil iş yükü dağılımı objektifleri arasındaki takas sebebiyle modelin optimal çözüme daha yavaş yaklaşması olduğu düşünülmektedir. Daha geniş bir çözüm uzayı keşfedilmesi için daha yüksek popülasyon ve nesil sayıları gerekli olur. Bu da çalışma zamanını yükseltir. Çizelge 5.5’de görüldüğü üzere, NSGA-II QEF modeli NSGA-II QE ve NSGA-II QF modellerine kıyasla %50 daha büyük bir popülasyon büyüklüğü ile çalışmaktadır.

Önerilen yaklaşım her zaman diliminde bir birleştirme şeması üretir. Test sonuçları her üç saniyede bir birleştirme şeması üretebilmiştir. Burada gözden kaçırılmamalıdır ki, test senaryoları tek iplik (thread) şeklinde çalıştırılmış ve testler 40 ipliğe sahip bir bilgisayarda 30 senaryo aynı anda çalışacak şekilde yapılmıştır. Önerilen yaklaşımın paralel şekilde çalışacak şekilde güncellenmesi ve bu problemin probleme dedike sunucularda çalıştırılması bu optimizasyon sürelerinin çok daha düşürecektir. Buna rağmen, üç saniyede bir optimize

birleştirme şemaları elde edilmesi üç objektifli ve iki objektifi arasında net bir takas olan NP-hard bir problem için kabul edilebilirdir.

5.4.4. Değişen talep sayılarının etkileri

Değişen uygulama talep hızlarının önerilen yaklaşıma etkilerinin incelenmesi için ek testler yapılmıştır. Testler için Çizelge 5.4’de verilen parametreler kullanılmıştır, sadece uygulama talep hızları değiştirilmiştir. Kullanılan talep hızları aşağıda verilmiştir.

- Yavaş uygulama talep hızında amaç ağ kapasitesinin %15’ini dolduracak hızda uygulama talebi üretmektir. Her zaman diliminde beş uygulama talebi üretilir.
- Normal uygulama talep hızında amaç ağ kapasitesinin %30’unu dolduracak hızda uygulama talebi üretmektir. Her zaman diliminde on uygulama talebi üretilir.
- Hızlı uygulama talep hızında amaç ağ kapasitesinin %50’sini dolduracak hızda uygulama talebi üretmektir. Her zaman diliminde on beş uygulama talebi üretilir.

Yavaş, normal ve hızlı uygulama talep üretimlerine dair yapılan testlerin sonuçları Çizelge 5.9’da verilmiştir.

Çizelge 5.9. Farklı uygulama talep hızlarına dair test sonuçları

Uygulama talep hızı	Model	SLA kontrat sayıları	Enerji tüketimi (mAs)	Ardarda n kere kullanılan servis sayıları			
				n=2	n=3	n=4	n=5
Yavaş	Random	78,92	2 527 591,50	30,25	1,25	0.00	0.00
	GA	105,17	2 526 486,42	34,58	1,25	0.00	0.00
	NSGA-II QE	106,00	1 542 366,00	71,67	7,92	0.00	0.00
	NSGA-II QF	103,58	2 527 390,75	47,67	0,33	0.00	0.00
	NSGA-II QEF	102,42	1 890 128,00	48,33	2,25	0.00	0.00
Normal	Random	144,08	5 095 867,42	208,50	34,75	5,50	0,92
	GA	195,08	5 145 070,67	234,25	50,75	13,42	3,33
	NSGA-II QE	193,58	3 542 310,08	357,58	142,83	63,33	29,58
	NSGA-II QF	193,75	5 128 975,75	283,67	24,92	1,75	0,08
	NSGA-II QEF	190,58	4 331 356,50	274,25	46,75	6,17	0,67
Hızlı	Random	211,75	7 411 059,00	536,33	191,92	69,25	24,25
	GA	297,33	7 441 403,50	552,33	208,83	81,75	33,75
	NSGA-II QE	293,33	5 328 689,08	771,33	481,33	329,58	237,58
	NSGA-II QF	290,42	7 428 174,75	643,92	179,33	25,75	3,67
	NSGA-II QEF	290,67	6 477 142,75	631,50	224,33	70,25	21,58

Çizelgeden görüldüğü üzere NSGA-II QEF enerji tüketimi ve adil iş yükü objektiflerini simultane şekilde optimize edebilmektedir. Sağlanan SLA kontratlarının sayıları aynı hız

içerisindeki farklı optimizasyon modellerinde oldukça yakındır. NSGA-II QEF hiçbir hızda önemli bir QoS kaybı göstermemektedir. Enerji tüketimi perspektifinden NSGA-II QEF modeli GA modeline kıyasla yavaş, normal ve hızlı uygulama talep hızlarında sırasıyla %25, %16 ve %13 daha az enerji tüketimine sebep olmaktadır. Enerji tüketiminde gözlenen düşüş uygulama talep hızı arttıkça azalmasına rağmen, her hızda bulunmaktadır. NSGA-II QEF modeli yavaş, normal ve hızlı uygulama talep hızlarında NSGA-II QE modeline kıyasla sırasıyla %22, %22 ve %21 daha yüksek enerji tüketimine, NSGA-II QF modeline kıyasla sırasıyla %25, %15 ve %13 daha düşük enerji tüketimine sahiptir. Adil iş yükü dağılımı perspektifinden, NSGA-II QEF modeli her uygulama talep hızında art arda kullanımları etkin bir şekilde azaltmaktadır.

NSGA-II QEF ile NSGA-II QE modelleri arasında %21 - %22 bandında NSGA-II QE lehine enerji tüketim farkı görülmektedir. Art arda kullanımlar her uygulama talep hızında NSGA-II QEF tarafından etkin bir şekilde azaltılmıştır. NSGA-II QEF bu sonuçları elde ederken QoS seviyelerinde yüksek bir düşüş gözlemlenmemiştir. Bu sonuçlar uyarınca NSGA-II QEF modelinin ve önerilen yaklaşımın farklı uygulama üretim hızlarında etkin bir şekilde çalışmaktadır.

5.4.4. Değişen ağ büyüklüklerinin etkileri

Önerilen yaklaşım ve NSGA-II QEF modeli farklı ağ büyüklüklerinde test edilmiştir. Çok küçük, küçük, ortalama ve büyük ağ büyüklükleri elde etmek için sis sunucusu başına düşen nesne sayıları ve nesne başına düşen IoT servis sayıları değiştirilmiştir. Değişen ağ büyüklüklerinde aynı ağ doluluğunu sağlamak için uygulama talep üretim hızları ağın %25'ini dolduracak seviyelere çekilmiştir. Diğer parametreler Çizelge 5.4'deki ile aynıdır. Tanımlanan ağ büyüklükleri aşağıdaki gibidir.

- Çok küçük ağ büyüklüğünde her sis sunucusu başına dört nesne, her nesne başına üç IoT servisi vardır. Her zaman diliminde beş yeni uygulama talebi gönderilir.
- Küçük ağ büyüklüğünde her sis sunucusu başına altı nesne, her nesne başına üç IoT servisi vardır. Her zaman diliminde yedi buçuk yeni uygulama talebi gönderilir.
- Ortalama ağ büyüklüğünde her sis sunucusu başına sekiz nesne, her nesne başına üç IoT servisi vardır. Her zaman diliminde on yeni uygulama talebi gönderilir.

- Büyük ağ büyüklüğünde her sis sunucusu başına on nesne, her nesne başına üç IoT servisi vardır. Her zaman diliminde on iki buçuk yeni uygulama talebi gönderilir.

Farklı ağ büyüklüklerine dair yapılan testlerin sonuçları Çizelge 5.10’da verilmiştir.

Çizelge 5.10. Farklı ağ büyüklüklerine dair test sonuçları

Uygulama talep hızı	Model	SLA kontrat sayıları	Enerji tüketimi (mAs)	Art arda n kere kullanılan servis sayıları			
				n=2	n=3	n=4	n=5
Çok küçük	Random	73,50	2 576 990,83	46,25	1,83	0,00	0,00
	GA	93,67	2 594 060,33	51,08	3,25	0,00	0,00
	NSGA-II QE	92,17	1 824 069,17	84,50	9,33	0,00	0,00
	NSGA-II QF	94,92	2 574 654,92	70,67	1,67	0,00	0,00
	NSGA-II QEF	88,92	2 164 259,25	64,33	2,92	0,00	0,00
Küçük	Random	103,83	3 709 424,25	101,83	12,08	1,58	0,00
	GA	145,58	3 695 780,50	109,25	16,58	1,83	0,25
	NSGA-II QE	141,58	2 357 086,33	201,17	69,92	30,17	13,75
	NSGA-II QF	144,08	3 686 965,00	147,08	6,33	0,08	0,00
	NSGA-II QEF	140,00	2 950 194,42	144,92	18,42	2,75	0,67
Orta	Random	152,75	5 372 684,17	178,08	28,75	4,17	0,58
	GA	222,67	5 380 074,08	200,33	42,17	10,83	2,75
	NSGA-II QE	215,00	3 187 812,67	356,00	166,08	79,33	39,33
	NSGA-II QF	217,50	5 400 360,17	256,92	13,58	0,58	0,00
	NSGA-II QEF	211,25	4 245 817,00	256,83	38,67	3,75	0,25
Büyük	Random	194,67	6 398 126,33	217,00	33,83	6,17	1,25
	GA	273,50	6 275 881,83	240,92	48,42	9,08	2,00
	NSGA-II QE	270,42	3 430 878,67	498,00	257,75	150,25	93,83
	NSGA-II QF	266,50	6 359 749,67	324,58	13,17	0,58	0,00
	NSGA-II QEF	266,58	4 797 350,83	346,17	55,83	9,00	2,00

Çizelgede görüldüğü üzere NSGA-II QEF enerji tüketimini ve art arda kullanım sayılarını GA modeline göre her ağ büyüklüğünde düşürebilmektedir. Ek olarak, NSGA-II QEF enerji

tüketimi ve adil iş yükü objektifleri arasında dengeyi her ağ büyüklüğünde kurabilmekte, iki objektifi simultane şekilde optimize edebilmektedir. Ancak NSGA-II QE ve NSGA-II QEF modelleri karşılaştırıldığında enerji tüketimine dair NSGA-II QEF modelinin büyük ağ büyüklüğüne adapte olma eksikliği görülmektedir. NSGA-II QEF çok küçük, küçük, ortalama ve büyük ağ büyüklüklerinde NSGA-II QE modeline kıyasla sırasıyla %18, %25, %33 ve %39 daha yüksek enerji tüketimine sebep olmaktadır. Adil iş yükü dağılımı objektifinde ise her ağ büyüklüğünde NSGA-II QEF başarılıdır.

5.4.5. Test sonuçlarının toplu değerlendirmesi

Test sonuçları önerilen yaklaşımın ve NSGA-II QEF modelinin IoT’de SC problemini QoS kısıtları altında, enerji etkin ve adil bir şekilde çözebildiği gözlemlenmiştir. NSGA-II QEF, sadece QoS objektifine sahip GA modeline kıyasla enerji tüketimini ve IoT servislerinin art arda kullanımlarını QoS seviyesinde yüksek bir düşüş olmadan sağlayabilmiştir. Ek olarak, literatürde kullanılan QoS ve enerji tüketimi farkında yaklaşımların enerji profili düşük olan IoT servislerinde adaletsiz iş yüklerine ve yüksek art arda kullanılma sayılarına sebep olduğu gösterilmiş, enerji tüketimi ve adil iş yükü dağılımı objektifleri arasındaki takas ortaya çıkarılmıştır. Görülmüştür ki NSGA-II QE modeli NSGA-II QEF modelinden daha az enerji tüketimine sahiptir. Benzer bir şekilde NSGA-II QF modelinin NSGA-II QEF modeline kıyasla daha adil iş yükü dağılımı yaptığı gözlemlenmiştir. Ancak test sonuçları göstermiştir ki NSGA-II QEF birbirine karşı çalışan enerji tüketimi ve adil iş dağılımı objektiflerini simultane bir şekilde optimize ederek iki objektifi de kabul edilebilir bir oranda sağlayabilmektedir.

5.5. Bölüm Değerlendirmesi

Bu bölümde anlatılan çalışma kapsamında FC temelli IoT ağlarda SC probleminin enerji etkin ve adil bir şekilde çözümüne dair bir yaklaşım ve model önerilmiştir. Önerilen yaklaşımda QoS, enerji etkinliği ve adil iş yükü dağılımı objektifleri olarak belirlenmiştir. SC problemi bir çok objektifli optimizasyon problemi olarak ele alınmış ve NSGA-II algoritması ile çözülmüştür. IoT alanının durmadan evrimleşen ve heterojen yapısını desteklemek adına genelleyici bir QoS modellemesi geliştirilmiş ve yaklaşımda kullanılmıştır. Geleneksel olarak bulut sunucularda gerçekleştirilen SC süreci sis sunucularına taşınmıştır, bu sayede daha ölçeklenebilir bir SC süreci elde edilmiştir. Sis

sunucuları kendilerine gelen uygulama taleplerini servis taleplerine kırar ve zaman pencerelerine böler. Ardından her zaman penceresi kendi içerisinde optimize edilir. Optimizasyon sürecinde NSGA-II algoritması kullanılır ve QoS, enerji etkinliği, adil iş yükü dağılımı objektifleri kullanılır.

Test sonuçları enerji etkinliği ile adil iş yükü dağılımı objektifleri arasında bir takas olduğunu göstermiştir. Enerji etkinliği objektifi ağı daha düşük enerji profiline sahip IoT servislerini yüksek sıklıkta kullanmaya itmektedir. Aynı IoT servislerinin sık kullanılması ağdaki adil iş yükü objektifine zarar vermektedir. QoS ve enerji etkinliği objektiflerine sahip olan NSGA-II QE modeli, QoS ve adil iş yükü dağılımı objektifine sahip NSGA-II QF modeline kıyasla [2, 3, 4, 5] kere art arda çalışan servis sayılarını sırasıyla [%38, %956, %6836, %21966] oranlarında arttırmaktadır. Adil iş yükü objektifi de ağdaki yüksek enerji profiline sahip servisleri diğer servisler ile aynı sıklıkta kullandırmaya çalıştığı için enerji tüketimini yükseltmektedir. NSGA-II QF modeli NSGA-II QE modeline kıyasla %57 oranında daha yüksek enerji tüketimine sebep olmaktadır.

Önerilen NSGA-II QEF modeli aralarında takas bulunan enerji etkinliği ve adil iş yükü dağılımı objektiflerini QoS seviyesinde yüksek kayba sebep olmadan simultane şekilde optimize edebilmektedir. Test sonuçlarına göre NSGA-II QEF modeli sadece QoS objektifi olan modellere kıyasla enerji tüketimini %20 oranında, art arda [3, 4, 5] kere kullanılan servis sayılarını [%5, %43, %85] oranlarında azaltmaktadır. NSGA-II QEF modeli NSGA-II QE modeline kıyasla [2, 3, 4, 5] kere kullanılan servis sayılarını %28, %75, %93 ve %99 oranlarında azaltırken, NSGA-II QF modeline göre enerji tüketimini %20 oranında azaltmaktadır.

IoT gün geçtikçe insan hayatının içine daha fazla nüfuz etmektedir. Dolayısıyla IoT ağların boyutları artacaktır. Bu artışla beraber ölçeklenebilirlik problemleri artmaktadır. Bu sebepten ötürü gelecekte IoT'de SC probleminin ölçeklenebilirlik problemi üzerine gidilecektir. Bu amaçla öncelikle önerilen modelin daha düşük çalışma zamanına sahip olması adına çalışmalar yapılacaktır.



6. SONUÇ VE ÖNERİLER

IoT geleceğin baskın internet vizyonu olarak kabul edilmektedir. Günümüzde en az bir tarafı insan olan iletişim miktarı M2M iletişim miktarı ile yakın miktardadır. Ancak bu denge M2M lehine bozulmaktadır. Hayatın her alanına nüfuz eden IoT uygulamaları insan hayatını devrimsel ölçekte değiştirmektedir. Şüphe yoktur ki bu değişim hızlanarak devam edecektir. IoT ağların sayıları ve büyüklükleri gün geçtikçe artmakta, hayatın çoğu alanında insan faktörü gereksiz bir hale gelmektedir. Akıllı tarım, kişiselleştirilmiş sağlık hizmetleri, Endüstri 4.0 temelli üretim ortamları, akıllı evler ve şehirler IoT sayesinde mümkün olan ve çoğu işlemin M2M iletişime dayalı ve otomatik bir şekilde yapıldığı alanlardır. Ancak, IoT'nin öngörülen yaygınlığa ve başarıya ulaşması için çözülmesi gereken zorluklar vardır. Bunlardan biri ve bu tez kapsamında odaklanılanı QoS sağlanmasıdır. SOA uyarınca servisler olarak sağlanan IoT işlevlerinin belirli kalite seviyelerinde sağlanması bir gerekliliktir. Bahsedilen kalite kriterleri QoS olarak özetlenir ve IoT kavramının en temel yapı taşlarından biridir. Bu sebeple IoT'de QoS sağlanması aktif bir alan problemidir. Bu tez kapsamında IoT'de QoS sağlanmasına dair çalışmalar yapılmıştır.

Öncelikle, IoT'de QoS sağlanmasına dair literatür taraması yapılmıştır. Literatür taramasında QoS odaklı IoT mimarileri, yapıları, iletişim protokolleri incelenmiş ve sunulmuştur. IoT'de kullanılan iletişim protokolleri QoS perspektifinden değerlendirilmiştir. Literatürde karşılaşılan QoS istekleri IoT ağın hangi katmanında sağlanabileceklerine göre kategorize edilmiştir ve sunulmuştur. IoT'de QoS seviyelerinin yükseltilmesini sağlayan teknikler taranmış ve incelenmiştir. Literatür taraması sonucunda görülmüştür ki IoT'de QoS sağlanması için çok katmanlı bir yaklaşım yapılması gereklidir. Her katmanın kendine has QoS istekleri olmasına rağmen bir IoT uygulamasına QoS sağlanması için her katmanda QoS-farkında bir yaklaşım uygulanması gereklidir. Kategorize edilen QoS istekleri kullanılarak her katmanda sıkça talep edilen isteklere dair üç problem belirlenmiştir.

Nesne katmanında veri bütünlüğü isteklerine odaklanılmıştır. Veriye dayanan IoT uygulamalarında veri bütünlüğü oldukça önemli bir QoS isteğidir. Bu isteğin tatmin edilebilmesine yönelik olarak ANFIS ve DL temelli özgün eksik veri tamamlama modelleri önerilmiştir. Gerçek veri üzerinde uygulanan modeller yüksek başarıda veri tahmin edebilmektedir. Test sonuçlarının ortaya çıkardığı üzere ANFIS temelli yöntemler çok daha hızlı çalışırken, DL temelli yöntemler daha yavaş çalışmasına rağmen özellikle tahmini zor

olan veri tiplerinde çok daha başarılı sonuçlar vermektedir. ANFIS ve DL temelli yöntemler SVR ve GKR regresyon modelleri ile karşılaştırılmış ve sonuçlar verilmiştir.

Çözülen ikinci problem ağ katmanında gecikmelerin minimize edilmesine dairdir. IoT uygulamalarının QoS seviyeleri farklıdır. Bazı uygulamalar çok da düşük gecikmeler talep ederken, bazı uygulamalar için gecikme kritik bir QoS isteri değildir. Bu gibi durumlarda ağ kaynaklarının yüksek QoS seviyeli uygulamalar için daha öncelikli kullanılması için önceliklendirme temelli özgün bir yaklaşım önerilmiştir. Temelinde yaklaşım, yüksek QoS seviyesine sahip paketlerin önceliklendirilerek gönderilmesi fikrine dayanır. Ağın ölçeklenebilirliğinin artırılması için FC, paket önceliklendirme için SDN teknolojileri kullanılmıştır. Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı'nın Hava Kalitesi İzleme Ağı kullanılarak bir FC temelli bir IoT ağ yapısı oluşturulmuştur. Ağ yapısı otomatik bir şekilde bir sonraki saate dair hava kalitesini öngörür ve bulut sunucuya iletir. Yüksek hava kirliliği öngören paketler önceliklendirilerek gönderilir. Veri akışı mevzu bahis ağdan toplanan gerçek veri kullanılarak elde edilmiştir. Test sonuçları, önerilen model olan PCBF-SDN modelinin öncelikli paketlerde gecikmeyi konvansiyonel IoT ağ yapısına göre %27 oranında azalttığını göstermiştir.

Basit IoT servislerinin birleştirilip uygulamalar haline getirilmesi problemine IoT'de SC problemi denir. SC problemi IoT ağın uygulama/servis katmanına ait bir problemdir. Günümüzde IoT'de SC problemine odaklanan çalışmalar genellikle QoS-farkında ve enerji etkindir. Ancak, bu çalışmalar IoT servisleri arasındaki iş yükünün adil paylaşılmasına dair bir hedefe sahip değildir. Adil iş yükü dağılımı aynı servislerin art arda kullanılmasından doğacak birçok problemi engeller ve ağ verimliliğini artırır. Ayrıca, literatür taramasının ortaya çıkardığına göre iş yükü dağılımı IoT'de QoS seviyelerinin artırılması için kullanılan tekniklerden biridir. Bu sebeplerden yola çıkarak üçüncü problem olarak IoT'de SC problemine QoS, enerji etkinliği ve IoT servisleri arası adil iş yükü hedefleri ile yaklaşmıştır. Bahsedilen üç hedefe sahip bir optimizasyon problemi formüle edilmiş ve problem NSGA-II algoritması temelli özgün bir optimizasyon modeli ile çözülmüştür. Test sonuçları enerji etkinliği ve adil iş yükü hedefleri arasında bir takas olduğunu ortaya koymuştur. Enerji etkinliği hedefi düşük enerji profiline sahip servisleri art arda kullanılmasına sebep olurken, adil iş yükü hedefi enerji profilinden bağımsız bir şekilde iş yükünün IoT servisleri arasında dağılmasına sebep olmaktadır. Bu takas yüzünden iki hedef birbirine zıt şekilde çalışmaktadır. Önerilen üç hedefli yaklaşım QoS seviyelerine ciddi zarar

vermeden enerji etkinliđi ve adil iş yükü hedeflerinin etkin bir şekilde ortada buluşturabilmektedir.

Bu tez kapsamında IoT’de QoS sağlanması problemine odaklanılmıştır. Öncelikle bir literatür taraması yapılmıştır. Ardından, IoT ağıın farklı katmanlarında farklı QoS isterlerine dair iyileştirme çalışmaları yapılmıştır. Odaklanılan üç problem başarı ile çözülmüştür ve özgün araştırma makaleleri olarak yayınlanmıştır. Yapılan literatür taraması ve çalışmalar ortaya çıkarmıştır ki, IoT’de QoS desteğinin tam anlamıyla sağlanması için yapılması gerekenler vardır. Bu gerekliliklerden gözümüze çarpanlar gelecekte yapacak araştırmacılara yol göstermesi amacıyla aşağıda verilmiştir.

- Literatürdeki çalışmaların çoğu spesifik QoS isterlerine yöneliktir. Bu durum, seçimden ziyade QoS isterlerini genelleyici bir şekilde gruplayacak modellemelerin bulunmamasından dolayıdır. SC problemi ve benzer birkaç problemde tarafımızdan yapılan çalışma ile benzer şekilde QoS isterlerini kategorize eden modellemeler bulunmasına rağmen, QoS isterlerini problemden bağımsız şekilde genelleyebilen bir model bulunmamaktadır. IoT’de QoS desteğinin tam anlamıyla sağlanabilmesi için tam anlamıyla genelleyici bir modelleme geliştirilmesi gereklidir.
- Bir IoT uygulaması sağlanırken IoT ağıın her katmanı kullanılır. Dolayısıyla IoT uygulamasına QoS sağlanırken her katmanda QoS sağlanmalıdır. QoS sağlanırken katmanlar arası bir devamlılık sunulması literatürde açık olan bir konudur.

KAYNAKLAR

1. Whitmore, A., Agarwal, A., and Da Xu, L. (2014). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*. 17(2):261-274.
2. Li, S., Xu, L.D., and Zhao, S. (2014). The internet of things: a survey. *Information Systems Frontiers*. 17(2):243-259.
3. Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*. 54(15):2787-2805.
4. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*. 17(4):2347-2376.
5. Waleed, J., Abduldaïm, A.M., Hasan, T.M., and Mohaisin, Q.S. (2018) *Smart home as a new trend, a simplicity led to revolution*. in 2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES).
6. Ismagilova, E., Hughes, L., Dwivedi, Y.K., and Raman, K.R. (2019). Smart cities: Advances in research—An information systems perspective. *International Journal of Information Management*. 47:88-100.
7. Anagnostopoulos, T., et al. (2021). Challenges and Solutions of Surveillance Systems in IoT-Enabled Smart Campus: A Survey. *IEEE Access*. 9:131926-131954.
8. Gharaibeh, A., et al. (2017). Smart Cities: A Survey on Data Management, Security, and Enabling Technologies. *IEEE Communications Surveys & Tutorials*. 19(4):2456-2501.
9. Srivastava, A. and Das, D.K. (2022). A Comprehensive Review on the Application of Internet of Thing (IoT) in Smart Agriculture. *Wireless Personal Communications*. 122(2):1807-1837.
10. Lova Raju, K. and Vijayaraghavan, V. (2020). IoT Technologies in Agricultural Environment: A Survey. *Wireless Personal Communications*. 113(4):2415-2446.
11. Friha, O., Ferrag, M.A., Shu, L., Maglaras, L., and Wang, X. (2021). Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. *IEEE/CAA Journal of Automatica Sinica*. 8(4):718-752.
12. Farooq, M.S., Riaz, S., Abid, A., Abid, K., and Naeem, M.A. (2019). A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access*. 7:156237-156271.
13. Xu, H., Yu, W., Griffith, D., and Golmie, N. (2018). A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. *IEEE Access*. 6:78238-78259.
14. Xu, L.D., He, W., and Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*. 10(4):2233-2243.

15. Guzel, M. and Ozdemir, S. (2019) *A New CEP-based Air Quality Prediction Framework for Fog based IoT*. in 2019 International Symposium on Networks, Computers and Communications (ISNCC). İstanbul, Türkiye.
16. Dilek, S., Irgan, K., Guzel, M., Ozdemir, S., Baydere, S., and Charnsripinyo, C. (2022). QoS-aware IoT networks and protocols: A comprehensive survey. *International Journal of Communication Systems*. 35(10):e5156.
17. Guzel, M. and Ozdemir, S. (2022). Fair and energy-aware IoT service composition under QoS constraints. *The Journal of Supercomputing*. 78:13427–13454.
18. Guzel, M., Kok, I., Akay, D., and Ozdemir, S. (2020). ANFIS and Deep Learning based missing sensor data prediction in IoT. *Concurrency and Computation: Practice and Experience*. 32(2):e5400.
19. İnağ, Y., Güzel, M., Okay, F., Demirci, M., and Özdemir, M. (2022). Priority Enabled Content Based Forwarding in Fog Computing via SDN. *Turkish Journal of Electrical Engineering and Computer Sciences*. 30(4):1439-1459.
20. Koshizuka, N. and Sakamura, K. (2010). Ubiquitous ID: Standards for Ubiquitous Computing and the Internet of Things. *IEEE Pervasive Computing*. 9(4):98-101.
21. Ding, J., Nemati, M., Ranaweera, C., and Choi, J. (2020). IoT Connectivity Technologies and Applications: A Survey. *IEEE Access*. 8:67646-67673.
22. Chettri, L. and Bera, R. (2020). A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. *IEEE Internet of Things Journal*. 7(1):16-32.
23. Nguyen, D.C., et al. (2022). 6G Internet of Things: A Comprehensive Survey. *IEEE Internet of Things Journal*. 9(1):359-383.
24. Guo, F., Yu, F.R., Zhang, H., Li, X., Ji, H., and Leung, V.C.M. (2021). Enabling Massive IoT Toward 6G: A Comprehensive Survey. *IEEE Internet of Things Journal*. 8(15):11891-11915.
25. Javed, F., Afzal, M.K., Sharif, M., and Kim, B.-S. (2018). Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review. *IEEE Communications Surveys & Tutorials*. 20(3):2062-2100.
26. Hahm, O., Baccelli, E., Petersen, H., and Tsiftes, N. (2016). Operating Systems for Low-End Devices in the Internet of Things: A Survey. *IEEE Internet of Things Journal*. 3(5):720-734.
27. Silva, M., Cerdeira, D., Pinto, S., and Gomes, T. (2019). Operating Systems for Internet of Things Low-End Devices: Analysis and Benchmarking. *IEEE Internet of Things Journal*. 6(6):10375-10383.
28. Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*. 10(7):1497-1516.

29. Bellavista, P., Berrocal, J., Corradi, A., Das, S.K., Foschini, L., and Zanni, A. (2019). A survey on fog computing for the Internet of Things. *Pervasive and Mobile Computing*. 52:71-99.
30. Popli, S., Jha, R.K., and Jain, S. (2022). Green IoT: A Short Survey on Technical Evolution & Techniques. *Wireless Personal Communications*. 123(1):525-553.
31. Alsamhi, S.H., Ma, O., Ansari, M.S., and Meng, Q. (2019). Greening internet of things for greener and smarter cities: a survey and future prospects. *Telecommunication Systems*. 72(4):609-632.
32. Malik, A. and Kushwah, R. (2022). A Survey on Next Generation IoT Networks from Green IoT Perspective. *International Journal of Wireless Information Networks*. 29(1):36-57.
33. Yu, Y., Bu, X., Yang, K., and Han, Z. (2018) *Green Fog Computing Resource Allocation Using Joint Benders Decomposition, Dinkelbach Algorithm, and Modified Distributed Inner Convex Approximation*. in 2018 IEEE International Conference on Communications (ICC). Kansas City, MO, USA.
34. Çorak, B., Yıldırım Okay, F., Güzel, M., Şahin, M., and Özdemir, S. (2018) *Comparative Analysis of IoT Communication Protocols*. in International Symposium on Networks, Computers and Communications (ISNCC). Rome, Italy.
35. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*. 4(5):1125-1142.
36. Naik, N. (2017) *Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP*. in 2017 IEEE International Systems Engineering Symposium (ISSE). Vienna, Austria.
37. Naik, N. and Jenkins, P. (2016) *Web protocols and challenges of Web latency in the Web of Things*. in 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN). Vienna, Austria.
38. Lohiya, R. and Thakkar, A. (2021). Application Domains, Evaluation Data Sets, and Research Challenges of IoT: A Systematic Review. *IEEE Internet of Things Journal*. 8(11):8774-8798.
39. Kumar, S., Tiwari, P., and Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*. 6:111.
40. Moustaka, V., Vakali, A., and Anthopoulos, L.G. (2019). A Systematic Review for Smart City Data Analytics. *ACM Computing Surveys*. 51(5):1-41.
41. Anagnostopoulos, T., et al. (2017). Challenges and Opportunities of Waste Management in IoT-Enabled Smart Cities: A Survey. *IEEE Transactions on Sustainable Computing*. 2(3):275-289.

42. Cirillo, F., Gomez, D., Diez, L., Elicegui Maestro, I., Gilbert, T.B.J., and Akhavan, R. (2020). Smart City IoT Services Creation Through Large-Scale Collaboration. *IEEE Internet of Things Journal*. 7(6):5267-5275.
43. Ke, R., Zhuang, Y., Pu, Z., and Wang, Y. (2021). A Smart, Efficient, and Reliable Parking Surveillance System With Edge Artificial Intelligence on IoT Devices. *IEEE Transactions on Intelligent Transportation Systems*. 22(8):4962-4974.
44. Ketu, S. and Mishra, P.K. (2022). A Contemporary Survey on IoT Based Smart Cities: Architecture, Applications, and Open Issues. *Wireless Personal Communications*. 125:1-49.
45. Mahmood, I. and Zubairi, J.A. (2019). Efficient Waste Transportation and Recycling: Enabling technologies for smart cities using the Internet of Things. *IEEE Electrification Magazine*. 7(3):33-43.
46. Ejaz, W., Naeem, M., Shahid, A., Anpalagan, A., and Jo, M. (2017). Efficient Energy Management for the Internet of Things in Smart Cities. *IEEE Communications Magazine*. 55(1):84-91.
47. Talebkhah, M., Sali, A., Marjani, M., Gordan, M., Hashim, S.J., and Rokhani, F.Z. (2021). IoT and Big Data Applications in Smart Cities: Recent Advances, Challenges, and Critical Issues. *IEEE Access*. 9:55465-55484.
48. Kirmat, A., Krejcar, O., Kertesz, A., and Tasgetiren, M.F. (2020). Future Trends and Current State of Smart City Concepts: A Survey. *IEEE Access*. 8:86448-86467.
49. Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet of Things Journal*. 4(1):269-283.
50. Elci, E., Yalcin, N., and Unaldi, S. (2021) *Internet of Things based Smart Home System Design for Individuals with Disabilities*. in 2021 6th International Conference on Computer Science and Engineering (UBMK). Ankara, Türkiye.
51. Sooraj, S.K., Sundaravel, E., Shreesh, B., and Sireesha, K. (2020) *IoT Smart Home Assistant for Physically Challenged and Elderly People*. in 2020 International Conference on Smart Electronics and Communication (ICOSEC). Trichy, India.
52. Mao, W., Zhao, Z., Chang, Z., Min, G., and Gao, W. (2021). Energy-Efficient Industrial Internet of Things: Overview and Open Issues. *IEEE Transactions on Industrial Informatics*. 17(11):7225-7237.
53. Ahmed, I., Jeon, G., and Piccialli, F. (2022). From Artificial Intelligence to eXplainable Artificial Intelligence in Industry 4.0: A survey on What, How, and Where. *IEEE Transactions on Industrial Informatics*. 18(8):5031 - 5042.
54. Bhuiyan, M.N., Rahman, M.M., Billah, M.M., and Saha, D. (2021). Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. *IEEE Internet of Things Journal*. 8(13):10474-10498.

55. Taimoor, N. and Rehman, S. (2022). Reliable and Resilient AI and IoT-Based Personalised Healthcare Services: A Survey. *IEEE Access*. 10:535-563.
56. Dhanvijay, M.M. and Patil, S.C. (2019). Internet of Things: A survey of enabling technologies in healthcare and its applications. *Computer Networks*. 153:113-131.
57. Qadri, Y.A., Nauman, A., Zikria, Y.B., Vasilakos, A.V., and Kim, S.W. (2020). The Future of Healthcare Internet of Things: A Survey of Emerging Technologies. *IEEE Communications Surveys & Tutorials*. 22(2):1121-1167.
58. Alshehri, F. and Muhammad, G. (2021). A Comprehensive Survey of the Internet of Things (IoT) and AI-Based Smart Healthcare. *IEEE Access*. 9:3660-3678.
59. Aceto, G., Persico, V., and Pescapé, A. (2020). Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0. *Journal of Industrial Information Integration*. 18:100129.
60. Riazul Islam, S.M., Kwak, D., Humaun Kabir, M., Hossain, M., and Kwak, K.-S. (2015). The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access*. 3:678-708.
61. Tran-Dang, H., Krommenacker, N., Charpentier, P., and Kim, D.-S. (2020). The Internet of Things for Logistics: Perspectives, Application Review, and Challenges. *IETE Technical Review*. 39(1):93-121.
62. Humayun, M., Jhanjhi, N., Hamid, B., and Ahmed, G. (2020). Emerging Smart Logistics and Transportation Using IoT and Blockchain. *IEEE Internet of Things Magazine*. 3(2):58-62.
63. Song, Y., Yu, F.R., Zhou, L., Yang, X., and He, Z. (2021). Applications of the Internet of Things (IoT) in Smart Logistics: A Comprehensive Survey. *IEEE Internet of Things Journal*. 8(6):4250-4274.
64. Hopkins, J. and Hawking, P. (2018). Big Data Analytics and IoT in logistics: a case study. *The International Journal of Logistics Management*. 29(2):575-591.
65. Ray, P.P., Mukherjee, M., and Shu, L. (2017). Internet of Things for Disaster Management: State-of-the-Art and Prospects. *IEEE Access*. 5:18818-18835.
66. Chamola, V., Hassija, V., Gupta, S., Goyal, A., Guizani, M., and Sikdar, B. (2021). Disaster and Pandemic Management Using Machine Learning: A Survey. *IEEE Internet of Things Journal*. 8(21):16047-16071.
67. Mei, G., Xu, N., Qin, J., Wang, B., and Qi, P. (2020). A Survey of Internet of Things (IoT) for Geohazard Prevention: Applications, Technologies, and Challenges. *IEEE Internet of Things Journal*. 7(5):4371-4386.
68. Neumann, G.B., Almeida, V.P.D., and Endler, M. (2018) *Smart Forests: fire detection service*. in 2018 IEEE Symposium on Computers and Communications (ISCC). Natal, Brazil.

69. Ishitha, S., Nagaraju, S., Mohan, H.A., Harshitha, M., Gowda, G.R., and N., J. (2021) *IoT based Anti-poaching and Fire Alarm System for Forest*. in 2021 IEEE Mysore Sub Section International Conference (MysuruCon). Hassan, India.
70. Mishra, V., Kumar, T., Bhalla, K., and Patil, M.M. (2018) *SuJAL: Design and Development of IoT-Based Real-Time Lake Monitoring System*. in 2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C). Bangalore, India.
71. Saharuddin, H.I., Rahman, T.A., Elijah, O., and Khairodin, F.N. (2019) *IoT Application in River Monitoring: Methods and Challenges*. in 2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). Kuala Lumpur, Malaysia.
72. Xu, G., Shi, Y., Sun, X., and Shen, W. (2019). Internet of Things in Marine Environment Monitoring: A Review. *Sensors*. 19(7):1711.
73. Ismail, S., Dawoud, D.W., Ismail, N., Marsh, R., and Alshami, A.S. (2022). IoT-Based Water Management Systems: Survey and Future Research Direction. *IEEE Access*. 10:35942-35952.
74. Farhady, H., Lee, H., and Nakao, A. (2015). Software-Defined Networking: A survey. *Computer Networks*. 81:79-95.
75. Masoudi, R. and Ghaffari, A. (2016). Software defined networks: A survey. *Journal of Network and Computer Applications*. 67:1-25.
76. Alam, I., et al. (2021). A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV. *ACM Computing Surveys*. 53(2):1-40.
77. Jazaeri, S.S., Jabbehdari, S., Asghari, P., and Haj Seyyed Javadi, H. (2021). Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions. *Cluster Computing*. 24(4):3187-3228.
78. Rafique, W., Qi, L., Yaqoob, I., Imran, M., Rasool, R.U., and Dou, W. (2020). Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*. 22(3):1761-1804.
79. Kumari, A., Tanwar, S., Tyagi, S., and Kumar, N. (2018). Fog computing for Healthcare 4.0 environment: Opportunities and challenges. *Computers & Electrical Engineering*. 72:1-13.
80. Hu, P., Dhelim, S., Ning, H., and Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*. 98:27-42.
81. Verma, R. and Chandra, S. (2021). A Systematic Survey on Fog steered IoT: Architecture, Prevalent Threats and Trust Models. *International Journal of Wireless Information Networks*. 28(1):116-133.

82. Omoniwa, B., Hussain, R., Javed, M.A., Bouk, S.H., and Malik, S.A. (2019). Fog/Edge Computing-Based IoT (FECIoT): Architecture, Applications, and Research Issues. *IEEE Internet of Things Journal*. 6(3):4118-4149.
83. Yu, W., et al. (2018). A Survey on the Edge Computing for the Internet of Things. *IEEE Access*. 6:6900-6919.
84. Pan, J. and Mcelhannon, J. (2018). Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet of Things Journal*. 5(1):439-449.
85. Sasaki, Y. (2022). A Survey on IoT Big Data Analytic Systems: Current and Future. *IEEE Internet of Things Journal*. 9(2):1024-1036.
86. Mohammadi, M., Al-Fuqaha, A., Sorour, S., and Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*. 20(4):2923-2960.
87. Bansal, M., Chana, I., and Clarke, S. (2021). A Survey on IoT Big Data. *ACM Computing Surveys*. 53(6):1-59.
88. Zhang, J. and Tao, D. (2021). Empowering Things With Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal*. 8(10):7789-7817.
89. Murshed, M.G.S., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., and Hussain, F. (2022). Machine Learning at the Network Edge: A Survey. *ACM Computing Surveys*. 54(8):1-37.
90. Dai, H.-N., Zheng, Z., and Zhang, Y. (2019). Blockchain for Internet of Things: A Survey. *IEEE Internet of Things Journal*. 6(5):8076-8094.
91. Shammar, E.A., Zahary, A.T., and Al-Shargabi, A.A. (2021). A Survey of IoT and Blockchain Integration: Security Perspective. *IEEE Access*. 9:156114-156150.
92. Belotti, M., Bozic, N., Pujolle, G., and Secci, S. (2019). A Vademecum on Blockchain Technologies: When, Which, and How. *IEEE Communications Surveys & Tutorials*. 21(4):3796-3838.
93. Bhutta, M.N.M., et al. (2021). A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access*. 9:61048-61073.
94. Lao, L., Li, Z., Hou, S., Xiao, B., Guo, S., and Yang, Y. (2021). A Survey of IoT Applications in Blockchain Systems. *ACM Computing Surveys*. 53(1):1-32.
95. Vaezi, M., et al. (2022). Cellular, Wide-Area, and Non-Terrestrial IoT: A Survey on 5G Advances and the Road Towards 6G. *IEEE Communications Surveys & Tutorials*. 24(2):1117 - 1174.
96. Shah, J. and Mishra, B. (2016) *IoT enabled environmental monitoring system for smart cities*. in 2016 International Conference on Internet of Things and Applications (IOTA). Pune, India.

97. Xiaojun, C., Xianpeng, L., and Peng, X. (2015) *IOT-based air pollution monitoring and forecasting system*. in 2015 International Conference on Computer and Computational Sciences (ICCCS). Greater Noida, India.
98. Li, L., Li, S., and Zhao, S. (2014). QoS-Aware Scheduling of Services-Oriented Internet of Things. *IEEE Transactions on Industrial Informatics*. 10(2):1497-1505.
99. Yao, J. and Ansari, N. (2019). Caching in Energy Harvesting Aided Internet of Things: A Game-Theoretic Approach. *IEEE Internet of Things Journal*. 6(2):3194-3201.
100. Herrero, R. (2020). MQTT-SN, CoAP, and RTP in wireless IoT real-time communications. *Multimedia Systems*. 26(6):643-654.
101. Khan, F.A., Noor, R.M., Mat Kiah, M.L., Noor, N.M., Altowaijri, S.M., and Rahman, A.U. (2019). QoS Enabled Layered Based Clustering for Reactive Flooding in the Internet of Things. *Symmetry*. 11(5):634.
102. Badawy, M.M., Ali, Z.H., and Ali, H.A. (2020). QoS provisioning framework for service-oriented internet of things (IoT). *Cluster Computing*. 23(2):575-591.
103. Al-Ali, A.R., Zualkernan, I.A., Rashid, M., Gupta, R., and Alikarar, M. (2017). A smart home energy management system using IoT and big data analytics approach. *IEEE Transactions on Consumer Electronics*. 63(4):426-434.
104. Bello, O. and Zeadally, S. (2019). Toward efficient smartification of the Internet of Things (IoT) services. *Future Generation Computer Systems*. 92:663-673.
105. Huang, J., Duan, Q., Zhao, Y., Zheng, Z., and Wang, W. (2017). Multicast Routing for Multimedia Communications in the Internet of Things. *IEEE Internet of Things Journal*. 4(1):215-224.
106. Sun, W., Wang, Z., and Zhang, G. (2021). A QoS-guaranteed intelligent routing mechanism in software-defined networks. *Computer Networks*. 185:107709.
107. Gravalos, I., Makris, P., Christodoulopoulos, K., and Varvarigos, E.A. (2018). Efficient Network Planning for Internet of Things With QoS Constraints. *IEEE Internet of Things Journal*. 5(5):3823-3836.
108. Zafar, S., Jangsher, S., Bouachir, O., Aloqaily, M., and Ben Othman, J. (2019). QoS enhancement with deep learning-based interference prediction in mobile IoT. *Computer Communications*. 148:86-97.
109. Tello-Oquendo, L., Lin, S.-C., Akyildiz, I.F., and Pla, V. (2019). Software-Defined architecture for QoS-Aware IoT deployments in 5G systems. *Ad Hoc Networks*. 93:101911.
110. Prerna, D., Tekchandani, R., Kumar, N., and Tanwar, S. (2021). An Energy-Efficient Cache Localization Technique for D2D Communication in IoT Environment. *IEEE Internet of Things Journal*. 8(6):4816-4829.

111. Abedin, S.F., Alam, M.G.R., Kazmi, S.M.A., Tran, N.H., Niyato, D., and Hong, C.S. (2019). Resource Allocation for Ultra-Reliable and Enhanced Mobile Broadband IoT Applications in Fog Network. *IEEE Transactions on Communications*. 67(1):489-502.
112. Al-Shammari, B.K.J., Al-Aboody, N., and Al-Raweshidy, H.S. (2018). IoT Traffic Management and Integration in the QoS Supported Network. *IEEE Internet of Things Journal*. 5(1):352-370.
113. Montazerolghaem, A. and Yaghmaee, M.H. (2020). Load-Balanced and QoS-Aware Software-Defined Internet of Things. *IEEE Internet of Things Journal*. 7(4):3323-3337.
114. He, Y., Tang, L., and Ren, Y. (2019). Maximizing Sleeping Capacity Based on QoS Provision for Information-Centric Internet of Things. *IEEE Access*. 7:111084-111094.
115. Yu, R., Xue, G., and Zhang, X. (2017). QoS-Aware and Reliable Traffic Steering for Service Function Chaining in Mobile Networks. *IEEE Journal on Selected Areas in Communications*. 35(11):2522-2531.
116. Tan, L., Zhu, Z., Ge, F., and Xiong, N. (2015). Utility Maximization Resource Allocation in Wireless Networks: Methods and Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 45(7):1018-1034.
117. Ye, Q. and Zhuang, W. (2017). Distributed and Adaptive Medium Access Control for Internet-of-Things-Enabled Mobile Networks. *IEEE Internet of Things Journal*. 4(2):446-460.
118. Ghafoor, K.Z., Kong, L., Rawat, D.B., Hosseini, E., and Sadiq, A.S. (2019). Quality of Service Aware Routing Protocol in Software-Defined Internet of Vehicles. *IEEE Internet of Things Journal*. 6(2):2817-2828.
119. Zhang, Y., et al. (2020). Cooperative Edge Caching: A Multi-Agent Deep Learning Based Approach. *IEEE Access*. 8:133212-133224.
120. Yao, J. and Ansari, N. (2019). Joint Content Placement and Storage Allocation in C-RANs for IoT Sensing Service. *IEEE Internet of Things Journal*. 6(1):1060-1067.
121. Khalid, A., Ul Ain, Q., Qasim, A., and Aziz, Z. (2021). QoS Based Optimal Resource Allocation and Workload Balancing for Fog Enabled IoT. *Open Computer Science*. 11(1):262-274.
122. Alrawahi, A.S., Lee, K., and Lotfi, A. (2019). A Multiobjective QoS Model for Trading Cloud of Things Resources. *IEEE Internet of Things Journal*. 6(6):9447-9463.
123. Jutila, M. (2016). An Adaptive Edge Router Enabling Internet of Things. *IEEE Internet of Things Journal*. 3(6):1061-1069.
124. Hasan, M.Z., Al-Turjman, F., and Al-Rizzo, H. (2018). Analysis of Cross-Layer Design of Quality-of-Service Forward Geographic Wireless Sensor Network Routing Strategies in Green Internet of Things. *IEEE Access*. 6:20371-20389.

125. Deng, G.-C. and Wang, K. (2018) *An Application-aware QoS Routing Algorithm for SDN-based IoT Networking*. in 2018 IEEE Symposium on Computers and Communications (ISCC). Natal, Brazil.
126. Yao, J. and Ansari, N. (2021). Caching in Dynamic IoT Networks by Deep Reinforcement Learning. *IEEE Internet of Things Journal*. 8(5):3268-3275.
127. Asad, M., Qaisar, S., and Basit, A. (2020). Client-Centric Access Device Selection for Heterogeneous QoS Requirements in Beyond 5G IoT Networks. *IEEE Access*. 8:219820-219836.
128. Govindan, K. and Azad, A.P. (2015) *End-to-end service assurance in IoT MQTT-SN*. in 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). Las Vegas, NV, USA.
129. Aazam, M., Harras, K.A., and Zeadally, S. (2019). Fog Computing for 5G Tactile Industrial Internet of Things: QoE-Aware Resource Allocation Model. *IEEE Transactions on Industrial Informatics*. 15(5):3085-3092.
130. Zhang, Z., Lung, C.-H., Lambadaris, I., and St-Hilaire, M. (2018) *IoT Data Lifetime-Based Cooperative Caching Scheme for ICN-IoT Networks*. in 2018 IEEE International Conference on Communications (ICC). Kansas City, MO, USA.
131. Basir, R., Qaisar, S.B., Ali, M., Naeem, M., Joshi, K.C., and Rodriguez, J. (2020) *Latency-Aware Resource Allocation in Green Fog Networks for Industrial IoT Applications*. in 2020 IEEE International Conference on Communications Workshops (ICC Workshops). Dublin, Ireland.
132. Awan, I., Younas, M., and Naveed, W. (2014) *Modelling QoS in IoT Applications*. in 2014 17th International Conference on Network-Based Information Systems. Salerno, Italy.
133. Zhao, L., Sun, W., Shi, Y., and Liu, J. (2018). Optimal Placement of Cloudlets for Access Delay Minimization in SDN-Based Internet of Things Networks. *IEEE Internet of Things Journal*. 5(2):1334-1344.
134. Ahammad, I., Khan, M.A.R., and Salehin, Z.U. (2021). QoS Performance Enhancement Policy through Combining Fog and SDN. *Simulation Modelling Practice and Theory*. 109:102292.
135. Hosen, A.S.M.S., et al. (2020). A QoS-Aware Data Collection Protocol for LLNs in Fog-Enabled Internet of Things. *IEEE Transactions on Network and Service Management*. 17(1):430-444.
136. Simiscuka, A.A. and Muntean, G.-M. (2018) *A Relay and Mobility Scheme for QoS Improvement in IoT Communications*. in 2018 IEEE International Conference on Communications Workshops (ICC Workshops). Kansas City, MO, USA.
137. Saha, N., Bera, S., and Misra, S. (2021). Sway: Traffic-Aware QoS Routing in Software-Defined IoT. *IEEE Transactions on Emerging Topics in Computing*. 9(1):390-401.

138. Ezdiani, S., Acharyya, I.S., Sivakumar, S., and Al-Anbuky, A. (2017). Wireless Sensor Network Softwarization: Towards WSN Adaptive QoS. *IEEE Internet of Things Journal*. 4(5):1517-1527.
139. Ojha, T., Misra, S., Raghuwanshi, N.S., and Poddar, H. (2019). DVSP: Dynamic Virtual Sensor Provisioning in Sensor–Cloud-Based Internet of Things. *IEEE Internet of Things Journal*. 6(3):5265-5272.
140. Zhou, Z., Dong, M., Ota, K., Wang, G., and Yang, L.T. (2016). Energy-Efficient Resource Allocation for D2D Communications Underlying Cloud-RAN-Based LTE-A Networks. *IEEE Internet of Things Journal*. 3(3):428-438.
141. Chen, G., Wu, J., Yang, W., Bashir, A.K., Li, G., and Hammoudeh, M. (2021). Leveraging Graph Convolutional-LSTM for Energy-Efficient Caching in Blockchain-Based Green IoT. *IEEE Transactions on Green Communications and Networking*. 5(3):1154-1164.
142. Tharinda Nishantha Vidanagama, V.G., Arai, D., and Ogishi, T. (2015) *M2M gateway selection scheme for smart wireless devices: an energy consumption perspective*. in 2015 10th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT). Colombo, Sri Lanka.
143. Sood, K., Karmakar, K.K., Yu, S., Varadharajan, V., Pokhrel, S.R., and Xiang, Y. (2020). Alleviating Heterogeneity in SDN-IoT Networks to Maintain QoS and Enhance Security. *IEEE Internet of Things Journal*. 7(7):5964-5975.
144. Costa, F.M., Georgantas, N., Gomes, R.D.A., Da Rocha, R.C.A., and Bouloukakakis, G. (2018) *Cross-layer QoS-Aware Resource Allocation for IoT-Enabled Service Choreographies*. in Proceedings of the 5th Workshop on Middleware and Applications for the Internet of Things. Rennes, France.
145. Huang, X., Cui, Y., Chen, Q., and 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.
146. Simiscuka, A.A. and Muntean, G.-M. (2018) *Age of Information as a QoS Metric in a Relay-Based IoT Mobility Solution*. in 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). Limassol, Cyprus.
147. Fontes, F., Rocha, B., Mota, A., Pedreiras, P., and Silva, V. (2020) *Extending MQTT-SN with Real-Time Communication Services*. in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Vienna, Austria.
148. Asif-Ur-Rahman, M., et al. (2019). Toward a Heterogeneous Mist, Fog, and Cloud-Based Framework for the Internet of Healthcare Things. *IEEE Internet of Things Journal*. 6(3):4049-4062.
149. Clemente, J., Valero, M., Mohammadpour, J., Li, X., and Song, W. (2017) *Fog computing middleware for distributed cooperative data analytics*. in 2017 IEEE Fog World Congress (FWC). Santa Clara, CA, USA.

150. Ahmad, E., Alaslani, M., Dogar, F.R., and Shihada, B. (2020). Location-Aware, Context-Driven QoS for IoT Applications. *IEEE Systems Journal*. 14(1):232-243.
151. Yu, R., Xue, G., and Zhang, X. (2019). Provisioning QoS-Aware and Robust Applications in Internet of Things: A Network Perspective. *IEEE/ACM Transactions on Networking*. 27(5):1931-1944.
152. Zhu, H., Cao, Y., Wei, X., Wang, W., Jiang, T., and Jin, S. (2019). Caching Transient Data for Internet of Things: A Deep Reinforcement Learning Approach. *IEEE Internet of Things Journal*. 6(2):2074-2083.
153. Bitam, S., Zeadally, S., and Mellouk, A. (2018). Fog computing job scheduling optimization based on bees swarm. *Enterprise Information Systems*. 12(4):373-397.
154. Yao, J. and Ansari, N. (2019). Fog Resource Provisioning in Reliability-Aware IoT Networks. *IEEE Internet of Things Journal*. 6(5):8262-8269.
155. Yousefpour, A., et al. (2019). FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework. *IEEE Internet of Things Journal*. 6(3):5080-5096.
156. Lu, R., Heung, K., Lashkari, A.H., and Ghorbani, A.A. (2017). A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT. *IEEE Access*. 5:3302-3312.
157. Gharaibeh, A., Khreishah, A., Mohammadi, M., Al-Fuqaha, A., Khalil, I., and Rayes, A. (2017). Online Auction of Cloud Resources in Support of the Internet of Things. *IEEE Internet of Things Journal*. 4(5):1583-1596.
158. Li, X., Tan, L., and Li, F. (2019). Optimal Cloud Resource Allocation With Cost Performance Tradeoff Based on Internet of Things. *IEEE Internet of Things Journal*. 6(4):6876-6886.
159. Yao, J. and Ansari, N. (2019). QoS-Aware Fog Resource Provisioning and Mobile Device Power Control in IoT Networks. *IEEE Transactions on Network and Service Management*. 16(1):167-175.
160. Gomes, R., Bouloukakakis, G., Costa, F., Georgantas, N., and Da Rocha, R., *QoS-Aware Resource Allocation for Mobile IoT Pub/Sub Systems*, in *Lecture Notes in Computer Science*. 2018, Springer International Publishing. 70-87.
161. Han, Y., Wang, R., and Wu, J. (2020). Random Caching Optimization in Large-Scale Cache-Enabled Internet of Things Networks. *IEEE Transactions on Network Science and Engineering*. 7(1):385-397.
162. Mukherjee, M., Guo, M., Lloret, J., Iqbal, R., and Zhang, Q. (2020). Deadline-Aware Fair Scheduling for Offloaded Tasks in Fog Computing With Inter-Fog Dependency. *IEEE Communications Letters*. 24(2):307-311.
163. Sun, X. and Ansari, N. (2018). Dynamic Resource Caching in the IoT Application Layer for Smart Cities. *IEEE Internet of Things Journal*. 5(2):606-613.

164. Wang, H., Xiong, D., Wang, P., and Liu, Y. (2017). A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices. *IEEE Access*. 5:16393-16405.
165. Banaie, F., Hossein Yaghmaee, M., Hosseini, S.A., and Tashtarian, F. (2020). Load-Balancing Algorithm for Multiple Gateways in Fog-Based Internet of Things. *IEEE Internet of Things Journal*. 7(8):7043-7053.
166. Abdullah, S. and Yang, K. (2013) *A QoS aware message scheduling algorithm in Internet of Things environment*. in 2013 IEEE Online Conference on Green Communications (OnlineGreenComm). Piscataway, NJ, USA.
167. Murtaza, F., Akhunzada, A., Islam, S.U., Boudjadar, J., and Buyya, R. (2020). QoS-aware service provisioning in fog computing. *Journal of Network and Computer Applications*. 165:102674.
168. Sun, X. and Ansari, N. (2018). Traffic Load Balancing Among Brokers at the IoT Application Layer. *IEEE Transactions on Network and Service Management*. 15(1):489-502.
169. Jiang, F., Yuan, Z., Sun, C., and Wang, J. (2019). Deep Q-Learning-Based Content Caching With Update Strategy for Fog Radio Access Networks. *IEEE Access*. 7:97505-97514.
170. Silva, A.L.M., Pérez-Alcázar, J.D.J., and Kofuji, S.T. (2019). Interoperability in semantic Web of Things: Design issues and solutions. *International Journal of Communication Systems*. 32(6):e3911.
171. Fan, Q. and Ansari, N. (2020). Towards Workload Balancing in Fog Computing Empowered IoT. *IEEE Transactions on Network Science and Engineering*. 7(1):253-262.
172. Park, T. and Saad, W. (2019). Distributed Learning for Low Latency Machine Type Communication in a Massive Internet of Things. *IEEE Internet of Things Journal*. 6(3):5562-5576.
173. Ni, Y., He, J., Cai, L., Pan, J., and Bo, Y. (2019). Joint Roadside Unit Deployment and Service Task Assignment for Internet of Vehicles (IoV). *IEEE Internet of Things Journal*. 6(2):3271-3283.
174. Jie, Y., Guo, C., Choo, K.-K.R., Liu, C.Z., and Li, M. (2020). Game-Theoretic Resource Allocation for Fog-Based Industrial Internet of Things Environment. *IEEE Internet of Things Journal*. 7(4):3041-3052.
175. Zhang, S. and Liu, J. (2020). Optimal Probabilistic Caching in Heterogeneous IoT Networks. *IEEE Internet of Things Journal*. 7(4):3404-3414.
176. Hua, Y., Guan, L., and Kyriakopoulos, K.G. (2020). A Fog caching scheme enabled by ICN for IoT environments. *Future Generation Computer Systems*. 111(10):82-95.
177. Khodaparas, S., Benslimane, A., and Yousefi, S. (2020). A software-defined caching scheme for the Internet of Things. *Computer Communications*. 158(May):178-188.

178. Al-Jamali, N.A.S. and Al-Raweshidy, H.S. (2021). Intelligent Traffic Management and Load Balance Based on Spike ISDN-IoT. *IEEE Systems Journal*. 15(2):1640-1651.
179. Yu, R., Kilari, V.T., Xue, G., and Yang, D. (2019) *Load Balancing for Interdependent IoT Microservices*. in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Paris, France.
180. Nezami, Z., Zamanifar, K., Djemame, K., and Pournaras, E. (2021). Decentralized Edge-to-Cloud Load Balancing: Service Placement for the Internet of Things. *IEEE Access*. 9:64983-65000.
181. Karkouch, A., Mousannif, H., Al Moatassime, H., and Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications*. 73:57-81.
182. Internet: Madden, S. Intel Lab Data. (2004) WEB: <http://db.csail.mit.edu/labdata/labdata.html>. Son Erişim Tarihi: 30.05.2020
183. Jang, J.-S.R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*. 23(3):665-685.
184. Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*. 521(7553):436-444.
185. Qin, Y., Zhang, S., Zhu, X., Zhang, J., and Zhang, C. (2007). Semi-parametric optimization for missing data imputation. *Applied Intelligence*. 27(1):79-88.
186. Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*. 39(1):1-22.
187. Internet: Delalleau, O., Courville, A., and Bengio, Y. Efficient EM Training of Gaussian Mixtures with Missing Data. (2018) WEB: <https://arxiv.org/abs/1209.0521>. Son Erişim Tarihi: 31/05/2022
188. Eirola, E., Lendasse, A., Vandewalle, V., and Biernacki, C. (2014). Mixture of Gaussians for distance estimation with missing data. *Neurocomputing*. 131:32-42.
189. Elbes, M., Alzubi, S., Kanan, T., Al-Fuqaha, A., and Hawashin, B. (2019). A survey on particle swarm optimization with emphasis on engineering and network applications. *Evolutionary Intelligence*. 12(2):113-129.
190. García-Laencina, P.J., Sancho-Gómez, J.-L., Figueiras-Vidal, A.R., and Verleysen, M. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*. 72(7-9):1483-1493.
191. Lobato, F., et al. (2015). Multi-objective genetic algorithm for missing data imputation. *Pattern Recognition Letters*. 68:126-131.
192. Devi Priya, R., Sivaraj, R., and Sasi Priyaa, N. (2017). Heuristically repopulated Bayesian ant colony optimization for treating missing values in large databases. *Knowledge-Based Systems*. 133:107-121.

193. Nekouie, A. and Moattar, M.H. (2019). Missing value imputation for breast cancer diagnosis data using tensor factorization improved by enhanced reduced adaptive particle swarm optimization. *Journal of King Saud University - Computer and Information Sciences*. 31(3):287-294.
194. Richman, M.B., Trafalis, T.B., and Adrianto, I., *Missing Data Imputation Through Machine Learning Algorithms*, in *Artificial Intelligence Methods in the Environmental Sciences*. 2009, Springer Netherlands. 153-169.
195. Tutz, G. and Ramzan, S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics & Data Analysis*. 90:84-99.
196. Troyanskaya, O., et al. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*. 17(6):520-525.
197. Abdella, M. and Marwala, T. (2005) *The use of genetic algorithms and neural networks to approximate missing data in database*. in IEEE 3rd International Conference on Computational Cybernetics, 2005. ICC3 2005. Mauritius.
198. Nelwamondo, F.V., Golding, D., and Marwala, T. (2013). A dynamic programming approach to missing data estimation using neural networks. *Information Sciences*. 237:49-58.
199. Ravi, V. and Krishna, M. (2014). A new online data imputation method based on general regression auto associative neural network. *Neurocomputing*. 138:106-113.
200. Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2004) *Extreme learning machine: a new learning scheme of feedforward neural networks*. in 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541). Budapest, Hungary.
201. Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*. 70(1-3):489-501.
202. Sovilj, D., et al. (2016). Extreme learning machine for missing data using multiple imputations. *Neurocomputing*. 174:220-231.
203. Laña, I., Olabarrieta, I.I., Vélez, M., and Del Ser, J. (2018). On the imputation of missing data for road traffic forecasting: New insights and novel techniques. *Transportation Research Part C: Emerging Technologies*. 90:18-33.
204. Silva-Ramírez, E.-L., Pino-Mejías, R., López-Coello, M., and Cubiles-De-La-Vega, M.-D. (2011). Missing value imputation on missing completely at random data using multilayer perceptrons. *Neural Networks*. 24(1):121-129.
205. Folguera, L., Zupan, J., Cicerone, D., and Magallanes, J.F. (2015). Self-organizing maps for imputation of missing data in incomplete data matrices. *Chemometrics and Intelligent Laboratory Systems*. 143:146-151.
206. Saitoh, F. (2016) *An ensemble model of self-organizing maps for imputation of missing values*. in 2016 IEEE 9th International Workshop on Computational Intelligence and Applications (IWCIA). Hiroshima, Japan.

207. Nishanth, K.J. and Ravi, V. (2016). Probabilistic neural network based categorical data imputation. *Neurocomputing*. 218:17-25.
208. Li, T., et al. (2017). Interval kernel Fuzzy C-Means clustering of incomplete data. *Neurocomputing*. 237:316-331.
209. Sefidian, A.M. and Daneshpour, N. (2019). Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model. *Expert Systems with Applications*. 115:68-94.
210. Zhang, L., Lu, W., Liu, X., Pedrycz, W., and Zhong, C. (2016). Fuzzy C-Means clustering of incomplete data based on probabilistic information granules of missing values. *Knowledge-Based Systems*. 99:51-70.
211. Mamdani, E.H. and Assilian, S. (1999). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Human-Computer Studies*. 51(2):135-147.
212. Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-15(1):116-132.
213. Akay, D., Chen, X., Barnes, C., and Henson, B. (2012). ANFIS modeling for predicting affective responses to tactile textures. *Human Factors and Ergonomics in Manufacturing & Service Industries*. 22(3):269-281.
214. Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*. 8(3):338-353.
215. Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*. 8(3):199-249.
216. Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning—II. *Information Sciences*. 8(4):301-357.
217. Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning—III. *Information Sciences*. 9(1):43-80.
218. Hu, Y.-C. (2007). Simple Fuzzy Grid Partition for Mining Multiple-Level Fuzzy Sequential Patterns. *Cybernetics and Systems*. 38(2):203-228.
219. Castellanos, F. and James, N. (2009) *Average Hourly Wind Speed Forecasting with ANFIS*. in 11th Americas Conference on Wind Engineering. San Juan, Puerto Rico.
220. Moradi, F., Bonakdari, H., Kisi, O., Ebtehaj, I., Shiri, J., and Gharabaghi, B. (2019). Abutment scour depth modeling using neuro-fuzzy-embedded techniques. *Marine Georesources & Geotechnology*. 37(2):190-200.
221. Dunn, J.C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*. 3(3):32-57.
222. Dunn, J.C. (1974). Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*. 4(1):95-104.

223. Bezdek, J.C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*. 10(2-3):191-203.
224. Abdulshahed, A.M., Longstaff, A.P., and Fletcher, S. (2015). The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing*. 27:158-168.
225. Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*. 9(8):1735-1780.
226. Internet: Lipton, Z.C., Berkowitz, J., and Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. (2015) WEB: <https://arxiv.org/abs/1506.00019>. Son Erişim Tarihi: 31/05/2022
227. Wei, D., et al. (2017). Research on Unstructured Text Data Mining and Fault Classification Based on RNN-LSTM with Malfunction Inspection Report. *Energies*. 10(3):406.
228. Internet: Abadi, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. (2016) WEB: <https://arxiv.org/abs/1603.04467>. Son Erişim Tarihi: 31/05/2022
229. Internet: Chollet, F. Keras. (2015) WEB: <https://github.com/fchollet/keras>. Son Erişim Tarihi: 30/05/2022
230. Internet: Çevre Şehircilik ve İklim Değişikliği Bakanlığı SİM (Sürekli İzleme Merkezi). (2022) WEB: <http://sim.csb.gov.tr/SERVICES/airquality>. Son Erişim Tarihi: 31/05/2022
231. Mahmud, R., Kotagiri, R., and Buyya, R., *Fog Computing: A Taxonomy, Survey and Future Directions*, in *Internet of Things*. 2018, Springer Singapore. 103-130.
232. Yousefpour, A., Ishigaki, G., and Jue, J.P. (2017) *Fog Computing: Towards Minimizing Delay in the Internet of Things*. in 2017 IEEE International Conference on Edge Computing (EDGE). Honolulu, HI, USA.
233. García-Pérez, C.A. and Merino, P. (2018). Experimental evaluation of fog computing techniques to reduce latency in LTE networks. *Transactions on Emerging Telecommunications Technologies*. 29(4):e3201.
234. Siasi, N. and Jaesim, A. (2020) *Priority-Aware SFC Provisioning in Fog Computing*. in 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). Las Vegas, NV, USA.
235. Kreutz, D., Ramos, F.M.V., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*. 103(1):14-76.
236. Kirkpatrick, K. (2013). Software-defined networking. *Communications of the ACM*. 56(9):16-19.

237. Internet: Department for Environment Food & Rural Affairs, Daily Air Quality Index. (2022) WEB: <https://uk-air.defra.gov.uk/air-pollution/daq>. Son Erişim Tarihi: 31/05/2022
238. Internet: The United States Environmental Protection Agency Air Quality Index (AQI) Basics. (2022) WEB: <https://www.airnow.gov/aqi/aqi-basics/>. Son Erişim Tarihi: 31/05/2022
239. Internet: CITEAIR. Air Quality in Europe. (2007) WEB: <http://www.airqualitynow.eu/>. Son Erişim Tarihi: 31/05/2022
240. Nord, J.H., Koohang, A., and Paliszkievicz, J. (2019). The Internet of Things: Review and theoretical framework. *Expert Systems with Applications*. 133:97-108.
241. Alsaryrah, O., Mashal, I., and Chung, T.-Y. (2018) *Energy-aware services composition for Internet of Things*. in 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Singapore.
242. Arellanes, D. and Lau, K.-K. (2020). Evaluating IoT service composition mechanisms for the scalability of IoT systems. *Future Generation Computer Systems*. 108:827-848.
243. Meo, M. (2020). Guest Editorial: Special Issue on Energy Efficiency for Internet of Things. *IEEE Transactions on Green Communications and Networking*. 4(4):944-945.
244. Khanouche, M.E., Atmani, N., and Cherifi, A. (2020). Improved Teaching Learning-Based QoS-Aware Services Composition for Internet of Things. *IEEE Systems Journal*. 14(3):4155-4164.
245. Urbietta, A., González-Beltrán, A., Ben Mokhtar, S., Anwar Hossain, M., and Capra, L. (2017). Adaptive and context-aware service composition for IoT-based smart cities. *Future Generation Computer Systems*. 76:262-274.
246. Xiao, R., Wu, Z., and Wang, D. (2019). A Finite-State-Machine model driven service composition architecture for internet of things rapid prototyping. *Future Generation Computer Systems*. 99:473-488.
247. Safaei, A., Nassiri, R., and Rahmani, A.M. (2022). Enterprise service composition models in IoT context: solutions comparison. *The Journal of Supercomputing*. 78(2):2015-2042.
248. Khanouche, M.E., Amirat, Y., Chibani, A., Kerkar, M., and Yachir, A. (2016). Energy-Centered and QoS-Aware Services Selection for Internet of Things. *IEEE Transactions on Automation Science and Engineering*. 13(3):1256-1269.
249. Haghi Kashani, M., Rahmani, A.M., and Jafari Navimipour, N. (2020). Quality of service-aware approaches in fog computing. *International Journal of Communication Systems*. 33(8):e4340.
250. Khanouche, M.E., Mouloudj, S., and Hammoum, M. (2019) *Two-steps QoS-aware services composition algorithm for Internet of Things*. in Proceedings of the 3rd International Conference on Future Networks and Distributed Systems. Paris, France.

251. Adel Serhani, M., El-Kassabi, H.T., Shuaib, K., Navaz, A.N., Benatallah, B., and Beheshti, A. (2020). Self-adapting cloud services orchestration for fulfilling intensive sensory data-driven IoT workflows. *Future Generation Computer Systems*. 108:583-597.
252. Arshad, R., Zahoor, S., Shah, M.A., Wahid, A., and Yu, H. (2017). Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond. *IEEE Access*. 5:15667-15681.
253. Shaikh, F.K., Zeadally, S., and Exposito, E. (2017). Enabling Technologies for Green Internet of Things. *IEEE Systems Journal*. 11(2):983-994.
254. Godinho, N., Curado, M., and Paquete, L. (2019) *Optimization of Service Placement with Fairness*. in 2019 IEEE Symposium on Computers and Communications (ISCC). Barcelona, Spain.
255. Li, S., Huang, J., Cheng, B., Cui, L., and Shi, Y. (2019) *FASS: A Fairness-Aware Approach for Concurrent Service Selection with Constraints*. in 2019 IEEE International Conference on Web Services (ICWS). Milan, Italy.
256. Shi, H., Prasad, R.V., Onur, E., and Niemegeers, I.G.M.M. (2014). Fairness in Wireless Networks: Issues, Measures and Challenges. *IEEE Communications Surveys & Tutorials*. 16(1):5-24.
257. Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R.H., Morrow, M.J., and Polakos, P.A. (2018). A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. *IEEE Communications Surveys & Tutorials*. 20(1):416-464.
258. Naha, R.K., et al. (2018). Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. *IEEE Access*. 6:47980-48009.
259. Salman, O., Elhajj, I., Chehab, A., and Kayssi, A. (2018). IoT survey: An SDN and fog computing perspective. *Computer Networks*. 143:221-246.
260. Sheikh Sofla, M., Haghi Kashani, M., Mahdipour, E., and Faghieh Mirzaee, R. (2022). Towards effective offloading mechanisms in fog computing. *Multimedia Tools and Applications*. 81(2):1997-2042.
261. Akasiadis, C., Tzortzis, G., Spyrou, E., and Spyropoulos, C. (2015) *Developing complex services in an IoT ecosystem*. in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). Milan, Italy.
262. Yachir, A., Amirat, Y., Chibani, A., and Badache, N. (2016). Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things. *IEEE Transactions on Automation Science and Engineering*. 13(1):85-102.
263. Adewuyi, A.A., Cheng, H., Shi, Q., Cao, J., Wang, X., and Zhou, B. (2022). SC-TRUST: A Dynamic Model for Trustworthy Service Composition in the Internet of Things. *IEEE Internet of Things Journal*. 9(5):3298-3312.

264. Adewuyi, A.A., Cheng, H., Shi, Q., Cao, J., Macdermott, A., and Wang, X. (2019). CTRUST: A Dynamic Trust Model for Collaborative Applications in the Internet of Things. *IEEE Internet of Things Journal*. 6(3):5432-5445.
265. Chai, Z.-Y., Du, M.-M., and Song, G.-Z. (2021). A fast energy-centered and QoS-aware service composition approach for Internet of Things. *Applied Soft Computing*. 100:106914.
266. Alsaryrah, O., Mashal, I., and Chung, T.-Y. (2018). Bi-Objective Optimization for Energy Aware Internet of Things Service Composition. *IEEE Access*. 6:26809-26819.
267. Alsaryrah, O., Mashal, I., and Chung, T.-Y. (2019) *A Fast IoT Service Composition Scheme for Energy Efficient QoS Services*. in Proceedings of the 2019 7th International Conference on Computer and Communications Management. Bangkok, Thailand
268. Kurdi, H., Ezzat, F., Altoaimy, L., Ahmed, S.H., and Youcef-Toumi, K. (2018). MultiCuckoo: Multi-Cloud Service Composition Using a Cuckoo-Inspired Algorithm for the Internet of Things Applications. *IEEE Access*. 6:56737-56749.
269. Sun, M., Zhou, Z., and Duan, Y. (2018) *Energy-Aware Service Composition of Configurable IoT Smart Things*. in 2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). Shenyang, China.
270. Sun, M., Zhou, Z., Wang, J., Du, C., and Gaaloul, W. (2019). Energy-Efficient IoT Service Composition for Concurrent Timed Applications. *Future Generation Computer Systems*. 100:1017-1030.
271. Zhou, Z., Zhao, D., Liu, L., and Hung, P.C.K. (2018). Energy-aware composition for wireless sensor networks as a service. *Future Generation Computer Systems*. 80(3):299-310.
272. Li, T., He, T., Wang, Z., and Zhang, Y. (2018). An Approach to Iot Service Optimal Composition for Mass Customization on Cloud Manufacturing. *IEEE Access*. 6:50572-50586.
273. Esmaeilifard, R. and Naderi, M. (2021). Distributed composition of complex event services in IoT network. *The Journal of Supercomputing*. 77(6):6123-6144.
274. Razian, M., Fathian, M., Wu, H., Akbari, A., and Buyya, R. (2021). SAIoT: Scalable Anomaly-Aware Services Composition in CloudIoT Environments. *IEEE Internet of Things Journal*. 8(5):3665-3677.
275. Sefati, S. and Navimipour, N.J. (2021). A QoS-Aware Service Composition Mechanism in the Internet of Things Using a Hidden-Markov-Model-Based Optimization Algorithm. *IEEE Internet of Things Journal*. 8(20):15620-15627.
276. Souri, A. and Ghobaei-Arani, M. (2021). Cloud manufacturing service composition in IoT applications: a formal verification-based approach. *Multimedia Tools and Applications*. 81:26759–26778.

277. Gharineiat, A., Bouguettaya, A., and Ba-Hutair, M.N. (2021). A Deep Reinforcement Learning Approach for Composing Moving IoT Services. *IEEE Transactions on Services Computing*. 15:1-14.
278. Ghari Neiat, A., Bouguettaya, A., and Sellis, T., *Spatio-Temporal Composition of Crowdsourced Services*, in *Service-Oriented Computing*. 2015, Springer Berlin Heidelberg. 373-382.
279. Neiat, A.G., Bouguettaya, A., Sellis, T., and Mistry, S. (2017). Crowdsourced Coverage as a Service: Two-Level Composition of Sensor Cloud Services. *IEEE Transactions on Knowledge and Data Engineering*. 29(7):1384-1397.
280. Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012) *Fog computing and its role in the internet of things*. in Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12. Helsinki, Finland.
281. Hernández-Nieves, E., Hernández, G., Gil-González, A.-B., Rodríguez-González, S., and Corchado, J.M. (2020). Fog computing architecture for personalized recommendation of banking products. *Expert Systems with Applications*. 140:112900.
282. Mukherjee, M., Shu, L., and Wang, D. (2018). Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges. *IEEE Communications Surveys & Tutorials*. 20(3):1826-1857.
283. Okay, F.Y. and 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.
284. Tange, K., De Donno, M., Fafoutis, X., and Dragoni, N. (2020). A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities. *IEEE Communications Surveys & Tutorials*. 22(4):2489-2520.
285. Masdari, M., Nozad Bonab, M., and Ozdemir, S. (2021). QoS-driven metaheuristic service composition schemes: a comprehensive overview. *Artificial Intelligence Review*. 54(5):3749-3816.
286. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2):182-197.
287. Campos Ciro, G., Dugardin, F., Yalaoui, F., and Kelly, R. (2016). A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*. 49(12):1272-1277.
288. Zhang, X., Tian, Y., Cheng, R., and Jin, Y. (2015). An Efficient Approach to Nondominated Sorting for Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*. 19(2):201-213.
289. Cui, L., et al. (2019). Joint Optimization of Energy Consumption and Latency in Mobile Edge Computing for Internet of Things. *IEEE Internet of Things Journal*. 6(3):4791-4803.



GAZİ GELECEKTİR..