

**NESNELERİN İNTERNETİNDE KENAR HESAPLAMA VE
MAKİNE ÖĞRENMESİ KULLANILARAK YENİ BİR GÖREV
TAMAMLAMA ALGORİTMASI**

MUHAMMET TAY

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
DOÇ. DR. ARAFAT ŞENTÜRK**

DÜZCE, 2024

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

NESNELERİN İNTERNETİNDE KENAR HESAPLAMA VE MAKİNE
ÖĞRENMESİ KULLANILARAK YENİ BİR GÖREV TAMAMLAMA
ALGORİTMASI

Muhammet TAY tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Arafat ŞENTÜRK

Düzce Üniversitesi

Jüri Üyeleri

Doç. Dr. Arafat ŞENTÜRK

Düzce Üniversitesi

Prof. Dr. Devrim AKGÜN

Sakarya Üniversitesi

Doç. Dr. Mehmet ŞİMŞEK

Milli Savunma Üniversitesi

Doç. Dr. Süleyman UZUN

Sakarya Uygulamalı Bilimler Üniversitesi

Doç. Dr. Muhammed Enes BAYRAKDAR

Düzce Üniversitesi

Tez Savunma Tarihi: 05/06/2024

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

5 Haziran 2024

Muhammet TAY





TEŐEKKÜR

İlk olarak, doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı değerli danışmanım Doç. Dr. Arafat ŐENTÜRK'e en içten dileklerle teşekkür ederim. Bana rehberlik ederek, sabrını ve bilgisini paylaşarak bu çalışmanın gelişimine katkıda bulundu. Değerli önerileri ve yönlendirmeleri, bu tezin kalitesini artırdı.

Ayrıca, tezin her aşamasında verdikleri fikir ile çalışmalarına ışık tutan değerli hocalarım Prof. Dr. Devrim AKGÜN ve Doç Dr. Mehmet ŐİMŐEK'e teşekkür etmek isterim. Değerli geri bildirimleri ve destekleri sayesinde bu çalışmanın kapsamını genişletmek ve derinleştirmek mümkün oldu.

Son olarak, süreç boyunca yardımlarını ve desteklerini esirgemeyen sevgili eşime sonsuz teşekkürlerimi sunarım. Ayrıca aileme ve arkadaşlarıma içten teşekkürlerimi sunmak istiyorum. Sürekli destekleri ve cesaretlendirmeleri olmasaydı, bu çalışma tamamlanamazdı.

5 Haziran 2024

Muhammet TAY

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	viii
ÇİZELGE LİSTESİ.....	ix
KISALTMALAR.....	x
SİMGELER.....	xii
ÖZET.....	xiii
ABSTRACT.....	xiv
EXTENDED ABSTRACT.....	xv
1. GİRİŞ.....	1
1.1. ÇALIŞMANIN AMACI VE ÖNERİLEN ÇÖZÜM YÖNTEMİ.....	3
1.2. TEZ ORGANİZASYONU.....	5
2. NESNELERİN İNTERNETİ, KENAR VE BULUT BİLİŞİM.....	6
2.1. NESNELERİN İNTERNETİ.....	7
2.2. BULUT BİLİŞİM.....	8
2.2.1. Bulut Bilişim Modelleri.....	8
2.2.2. Dağıtım Modelleri.....	10
2.2.3. Bulut Tabanlı Hizmetler.....	13
2.3. KENAR BİLİŞİM.....	15
2.3.1. Kenar Bilişim Mimarileri.....	16
2.4. NESNELERİN İNTERNETİ, KENAR VE BULUT BİLİŞİM İLİŞKİSİ.....	17
2.4.1. Hiyerarşik Mimari.....	18
2.4.2. Hiyerarşik Mimarinin Avantajları.....	21
2.4.2.1. Cevap Süresi.....	21
2.4.2.2. Enerji Tüketimi.....	21
2.4.2.3. Bant Genişliği.....	22
2.4.2.4. Depolama.....	22
2.4.3. Hiyerarşik Mimarideki Zorluklar.....	22
2.4.3.1. Hareketlilik.....	23
2.4.3.2. Kaynak Yönetimi.....	23
2.4.3.3. Güvenlik ve Gizlilik.....	23
2.4.3.4. Görev Aktarmı.....	24
2.5. KENAR BİLİŞİMDE GÖREV AKTARIMI.....	24
3. NESNELERİN İNTERNETİ, KENAR BİLİŞİM VE BULUT BİLİŞİMDE KAYNAK KULLANIMI.....	26
3.1. GİRİŞ.....	26
3.2. OPTİMİZASYON TABANLI ÇALIŞMALAR.....	26
3.2.1. Ölçeklenebilirlik ve Hareketlilik.....	26
3.2.2. Görev Aktarımı.....	27
3.2.3. Yorumlanabilirlik.....	28
3.2.4. Enerji Tüketimi.....	28
3.2.5. Planlama.....	29

3.3. MAKİNE ÖĞRENMESİ TABANLI ÇALIŞMALAR.....	31
3.3.1. Denetimli Öğrenme	31
3.3.2. Denetimsiz Öğrenme.....	32
3.3.3. Pekiştirmeli Öğrenme	34
3.4. HİBRİD ÇALIŞMALAR.....	35
3.5. İOT, KENAR VE BULUT BİLİŞİMİN KARŞILAŞTIRILMASI.....	38
3.6. SONUÇ	40
4. MATERYAL VE METOD	43
4.1. VERİ SETİ	43
4.2. VERİ ÖN İŞLEME	47
4.2.1. Veri Seti Normalleştirme Teknikleri.....	47
4.2.2. Veri Setinin Matlab ile Normalleştirilmesi	48
4.3. NİTELİKLERİN SEÇİLMESİ.....	49
4.3.1. KA Modeli İçin Niteliklerin Seçimi	50
4.3.2. DR Modeli İçin Niteliklerin Seçimi	51
4.4. WEKA ARACI İLE VERİ SETİNİN EĞİTİLMESİ.....	51
4.4.1. Karar Ağacı Algoritması	52
4.4.2. Doğrusal Regresyon Algoritması.....	53
4.5. EDGE CLOUDSİM BENZETİM ARACI.....	54
5. NESNELERİN İNTERNETİNDE KENAR HESAPLAMA VE MAKİNE ÖĞRENMESİ KULLANILARAK YENİ BİR GÖREV TAMAMLAMA ALGORİTMASI	56
5.1. SİSTEM MODELİ.....	56
5.2. KA-KBH	58
5.3. DENEYSEL SONUÇLAR.....	61
6. ÖNERİLEN YÖNTEMİN DEĞERLENDİRİLMESİ.....	63
6.1. KARŞILAŞTIRILAN ALGORİTMALAR.....	63
6.2. BENZETİMİN OLUŞTURULMASI VE PARAMETRELER	65
6.3. SİMÜLASYON SONUÇLARI VE TARTIŞMA.....	66
7. SONUÇ VE ÖNERİLER	73
8. KAYNAKLAR.....	75
ÖZGEÇMİŞ	90

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1. Üç katmanlı karar verici mimari.....	4
Şekil 2. Bulut Bilişim Modelleri.....	8
Şekil 3. Bulut Bilişim Hizmetleri.....	13
Şekil 4. Kenar hesaplama sistemleri ve IoT ilişkisi.....	16
Şekil 5. Üç katmanlı kenar hesaplama mimarisi.....	19
Şekil 6. ML tabanlı algortiamaların cevap süresi.....	40
Şekil 7. ML tabanlı çalışmaların doğruluk oranları.....	40
Şekil 8. Eğitilmiş veri setinin dağılımı.....	49
Şekil 9. WEKA arayüzü.....	52
Şekil 10. EdgeCloudSim benzetim aracı arayüzü.....	55
Şekil 11. Üç katmanlı araç kenar hesaplama sistemi mimarisi.....	57
Şekil 12. KA-KBH algortiması sözde kodu.....	61
Şekil 13. Araç hareket alanı ve RSU'ları konumlandırma.....	65
Şekil 14. Karşılaştırma algoritmalarının başarısız görev yüzdeleri.....	67
Şekil 15. Bilgi uygulaması tabanlı başarısız görevler.....	68
Şekil 16. VM kapasitesine göre başarısız görevler.....	69
Şekil 17. Ortalama deneyim kalitesi (QoE).....	70
Şekil 18. Kenar sunucuda hizmet süresi.....	71
Şekil 19. Bulut sunucuda hizmet süresi.....	72

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 1. Optimizasyon tabanlı çalışmaların analizi.....	30
Çizelge 2. Hibrit çalışmaların analizi	37
Çizelge 3. IoT, Kenar ve Bulut Bilişim.	38
Çizelge 4. Ham veri setinden bir kesit.	45
Çizelge 5. KA-KBH yönteminde kullanılan veri seti nitelikleri.	50
Çizelge 6. DR algoritmasının nitelikleri.	51
Çizelge 7. KA tabanlı görev aktarım kararı için model parametreleri.....	62
Çizelge 8. Veri setine göre ayrıntılı doğruluk oranları.	62
Çizelge 9. Benzetim parametreleri.....	66
Çizelge 10. Uygulama özellikleri.	66



KISALTMALAR

AC	Accuracy
ACO	Ant colony optimization
AFRM	Fuzzy resource management framework
AI	Artificial intelligence
ANN	Artificial neural network
AP	Access point
API	Application programming interface
ARIMA	Otomatik gerilemeli entegre hareketli Ortalama
ARMA	Otomatik gerileyen hareketli ortalama
AWS	Amazon web services
BPNN	Geri yayımlı sinir ağı
CC	Cloud computing
CN	Cell network
CPU	Central processing unit
CRM	Müşteri ilişkileri yönetimi
D2D	Device to device
DB	Database
DBaaS	Database as a service
DBN	Deep belief network
DNN	Deep neural network
DQL	Deep q-learning
DRAM	Delayed rejection adaptive metropolis
DT	Decisin tree
DVM	Destek vektör makineleri
EC	Edge computing
EFC	Edge, fog and cloud
ELBS	Bulanık ve olasılıksal sinir ağlarına dayalı etkili yük dengeleme stratejisi
ENN	Elman sinir ağı
ESA	Evrişimsel sinir ağı
FC	Fog computing
FI	Fuzzy interface
FOTO	Fruit fly optimization based task offloading
FRess	Fog resource selection services
GA	Genetik algoritma
GCP	Google cloud platform
GI	Task length
GPIS	Capacity of cloud
GPS	Global positioning system
GSM	Global system of mobile
Het-MEC	Heterojen MEC
IaaS	Infrastructure as a service
IoT	Internet of things
IoV	Internet of vehicle
IWPSO	Geliştirilmiş ağırlık bazlı particle swarm optimization
KA	Karar Ağacı
KB	Kenar bilişim
KBB	Kenar ve bulut bilişim
KH	Kenar hesaplama

LR	Logistic regression
LSTM	Long short-term memory
MAB	Multi armed bandit
MAN	Metropolitan area network
MANET	Mobile ad hoc networks
MDP	Markov decision process
MEC	Mobile edge computing
MEETS	Maksimum enerji verimli görev zamanlaması
MeFoRE	Media fog kaynak tahmini
ML	Machine learning
MLP	Multi layer perceptron
MMPP	Markov modulated poisson process
MSaaS	Multimedia sensing as servise
MySQL	My structured query language
NP	Nondeterministic polynomial time
NI	Not identified
NP	Nunlinear problem
PaaS	Platform as a service
PSO	Particle swarm optimization
QoE	Quality of experince
QoS	Quality of service
RAM	Random access memory
RDS	Relational database service
RF	Random forest
RL	Reinforcement learning
RSU	Road side unit
RT	Response time
SaaS	Software as a service
SLA	Service level agreement
SMA	Simple moving average
SPA	Project allocation
Tn	True negative
TSOM	İki aşamalı optimizasyon
UCC	User-centric cooperation
V2I	Vehicle to infrastructure
V2R	Vehicle to road side
V2V	Vehicle to vehicle
VANET	Vehicular area network
VEC	Vehicle edge computing
VM	Virtual machine
WLAN	Wireless local area network
YSA	Yapay sinir ağı
YTA	Yazılım tabanlı ağlar

SİMGELER

Gbps	Gigabayt per second
ms	Milisaniye
Q	Bekleme Süresi
S	Enropy
Gain	Kazanç
T_n	True negative
G	Gecikme
B	Başarılı
t	Görev (Task)
r	Rastgele (random)
i	Araç
Y_n	Görev aktarımı
S_{min}	Sonuç (En düşük gecikmeye sahip offload yönte)



ÖZET

NESNELERİN İNTERNETİNDE KENAR HESAPLAMA VE MAKİNE ÖĞRENMESİ KULLANILARAK YENİ BİR GÖREV TAMAMLAMA ALGORİTMASI

Muhammet TAY
Düzce Üniversitesi

Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Arafat ŞENTÜRK

Haziran 2024, 89 sayfa

Araçların interneti (IoV) alanında hızla gelişmeler yaşanıyor ve talepler her geçen gün artıyor. Yakın gelecekte, bağlantılı, otonom, paylaşımlı ve elektrikli araçlara olan ilgi artacak. Bu talep, IoV alanında büyük veri akışı ve depolama gibi temel sorunları beraberinde getirecek. Ayrıca, IoV sistemlerinin gecikme hassasiyeti ve veri kaybını en aza indirme ihtiyacı önem taşıyor. Kenar bilişim araçlarının kullanımı, bu sorunların çözümünde önemli bir rol oynayabilir. KB sistemlerinde gecikme, bant genişliği ve enerji tüketimi gibi temel faktörlerin iyileştirilmesi gerekiyor. Bu iyileştirmeler, genellikle karmaşık ve doğrusal olmayan problemlerle karşılaşılabilen zorlu konular arasında yer alıyor. Bu sorunların çözümü için etkili optimizasyon ve makine öğrenimi tekniklerinden yararlanılabilir. Bu çalışmada, görev aktarım oranını ve hizmet süresini artırmak amacıyla iki aşamalı bir makine öğrenimi yöntemi önerilmektedir. Önerilen yöntemde, ilk aşamada, görevin hangi hesaplama aracına gönderileceğine karar vermek için karar ağacı algoritması kullanılıyor. İkinci aşamada ise, gecikme hassasiyeti olan bir hesaplama aracı seçmek için doğrusal regresyon yöntemi kullanılmıştır. Önerilen yöntemin performansı, edgeCloudSim benzetim aracı kullanılarak analiz edilmiş ve elde edilen sonuçlar, önerilen yöntemin literatürdeki diğer algoritmalarından daha iyi sonuçlar sunduğunu göstermiştir.

Anahtar sözcükler: Bulut bilişim, Kenar bilişim, Görev aktarımı, Karar ağacı.

ABSTRACT

A NOVEL TASK OFFLOADING ALGORITHM USING EDGE COMPUTING AND MACHINE LEARNING IN THE INTERNET OF THINGS

Muhammet TAY

Düzce University

Graduate School of Natural and Applied Sciences, Department of Computer
Engineering.

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Arafat ŞENTURK

June 2024, 89 pages

Rapid advancements are occurring in the Internet of Vehicles (IoV) domain, with demands increasing each passing day. In the near future, there will be a rising interest in connected, autonomous, shared, and electric vehicles. This surge in demand will bring forth fundamental challenges in IoV, such as large data flow and storage. Additionally, there is a crucial need to minimize latency and data loss in IoV systems. The utilization of Edge Computing tools can play a significant role in addressing these challenges. Improvements in EC systems, including factors like latency, bandwidth, and energy consumption, are imperative. These enhancements often involve tackling complex and non-linear issues. Effective optimization and machine learning techniques can be leveraged to solve these problems. In this study, a two-stage machine learning approach is proposed to increase task transfer rates and service duration. In the first stage, a decision tree algorithm is employed to determine which computing tool a task should be sent to. In the second stage, linear regression is used to select a delay-sensitive estimator. The performance of the proposed method is analyzed using the edgeCloudSim simulation tool, revealing superior results compared to other algorithms in the literature.

Keywords: Cloud computing, Edge Computing, Task ofloading, Decision tree.

EXTENDED ABSTRACT

A NOVEL TASK OFFLOADING ALGORITHM USING EDGE COMPUTING AND MACHINE LEARNING IN THE INTERNET OF THINGS

Muhammet TAY

Düzce University

Graduate School of Natural and Applied Sciences, Department of Computer
Engineering.

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Arafat ŞENTURK

June 2024, 89 pages

1. INTRODUCTION

Internet of things (IoT) stands out as a technology where objects can communicate with each other and other systems by connecting to the internet. This technology, which can be used in a wide range of applications from smart homes to industrial environments, enables remote control by increasing communication between devices. However, issues such as security and cyber-attack risks are also important issues.

The number of IoT devices is growing rapidly, the amount of data generated by these devices is a challenge. While the development of 5G technology enables data transfer at higher speeds, it also requires efficient processing and storage of that data.

Since IoT devices have limited resources, they need energy-saving edge computing (EC) technologies. In order to utilize the resources of sensors more efficiently, we propose a hierarchical architecture that integrates edge, and cloud (EC) systems. This architecture aims to save energy by making intelligent decisions about which systems to use and when.

The aim of the work is to enable sensors to save energy, optimize data analysis and manage IoT networks more effectively. A hierarchical decision-making architecture aims to provide more effective response times in emergency and delay-sensitive systems. Moreover, it offers advantages such as reducing energy consumption, increasing storage capacity, and preventing data loss with a more effective collaborative EC system.

As a result, this proposed model aims to utilize IoT technology in a more sustainable and efficient way. The hierarchical architecture is an important step forward in this field by aiming to achieve the best performance despite the complexity and energy limitations of IoT devices.

2. MATERIAL AND METHODS

This paper focuses on offloading methods based on machine learning. Offloading refers to the offloading of computational operations from one device to another and is analyzed in the context of edge computing. Edge computing is a paradigm that enables information processing functions to be performed closer to their location. Machine learning-based offloading is a strategy used to balance the workload across devices and optimize operations. This paper examines the theoretical foundations of this strategy and discusses various offloading algorithms and the factors affecting their performance.

It is stated that the main components of the vehicle edge computing (VEC) architecture are the cloud layer, the edge cloud layer and the smart vehicle layer. The cloud layer emphasizes the advantages of large data processing and storage, and the edge cloud layer provides the connection between smart vehicles and the cloud. It is emphasized that the smart vehicle layer offers various services to users and has capabilities such as providing V2V and V2I communication, sensing the environment and understanding driver behavior.

Under the resource utilization heading, it was explained that the VEC architecture has three layers: the cloud layer, the edge cloud layer and the intelligent vehicle layer. Focusing on the advantages of the cloud layer, its functions such as data collection, data mining, analysis optimization, storage and computation of complex data are explained. The edge cloud layer provides the connection between smart vehicles and the cloud and offers various services. The smart vehicle layer offers various services to users by sorting them into infotainment, network, storage and computation.

In addition, the advantages of VEC are discussed in detail. Short response time, manageability of energy consumption, efficiency of bandwidth utilization and storage advantages are among the highlights. The various services provided by VEC and how these services are realized through the collaboration between smart vehicles, road-side units (RSUs) and the cloud are explained in detail.

Finally, the challenges faced by the VEC system are analyzed. These challenges include the high mobility of vehicles, resource management, security and privacy issues, and mission offloading. Heterogeneous MEC systems are proposed, and the coordination and interaction of these systems raise new research challenges.

In section 2.4.2, the advantages of the hierarchical architecture are discussed in detail.

Short response time, efficiency of bandwidth utilization and storage advantages are the main highlights. Furthermore, the various services provided by the VEC and how these services are realized through the collaboration between smart vehicles, RSUs and the cloud are described in detail.

This paragraph discusses the various services provided and challenges faced by VEC, focusing on the key components, architecture, and advantages of vehicle edge computing systems.

This paper addresses machine learning-based offloading in edge computing. Machine learning is the process by which computer algorithms learn through experience and is usually categorized into three groups: supervised, unsupervised and reinforcement learning. In this paper, we analyze and propose the adaptation of existing machine learning algorithms that can be effective in resource allocation.

One proposed method used in the study utilizes a three-tier VEC architecture. In this architecture, vehicles are at the bottom layer, edge servers are at the second layer (RSUs), and cloud computing technology is at the third layer. The data generated by the vehicles can be processed in three different ways: wireless transmission to the RSU, transmission to the edge server, and transmission to the cloud via GSM. The advantages of each transmission path should be chosen based on various criteria (capacity, cost, latency).

The paper emphasizes that factors such as access technologies, infrastructure topology, and delay affect the performance of VECs. Furthermore, the steps of the proposed algorithm such as training dataset, data preprocessing, and training the dataset are explained in detail. The decision tree (DT) based task completion mechanism is evaluated using a simulation tool with the trained dataset and feature selection. The study shows that the proposed algorithm provides a significant increase in success rates compared to other benchmarking algorithms.

3. RESULTS AND DISCUSSIONS

In the evaluation phase of the study, the performance of different offloading strategies is compared, and the results are analyzed. The advantages and limitations of machine learning-based offloading are identified and its suitability for real-world applications is discussed. The results show that machine learning-based offloading can be used effectively in an edge computing environment. Moreover, this study will lay the foundation for future research and contribute to the development of the edge computing

field.

In this research, a solution proposal is presented on machine learning based offloading methods in edge computing. It has been determined from the literature that traditional machine learning algorithms can be effective in resource allocation and therefore analyzing and adapting existing algorithms can be an effective solution. In this study, a three-layer VEC architecture is used to solve the offloading problem. In this architecture, vehicles with wireless communication capability are at the bottom layer, RSUs acting as edge servers are at the second layer and cloud computing technology is at the top layer. The advantages of each data transfer method should be evaluated according to criteria such as capacity, cost, latency and energy consumption, and the most appropriate transfer method should be selected.

The access technologies used in VECs, and the topology of the infrastructure significantly affect the system performance. The proposed architecture predicts the use of advanced applications in multi-layer multiple access VEC architectures. In the proposed architecture, RSUs can connect to the MAN and share their computing capacity through task migration. The VEC layer can be considered as a shared resource pool and MAN can provide a solution to the handover problem.

In this study, a detailed training and testing process was carried out on the proposed DT based task completion mechanism. The training data set includes the latency of the network data transmission tools and the capacity of the computational tools expressed in the system model. The preprocessing operations performed on the data set aimed to obtain a balanced training data set. The proposed algorithm worked successfully on the obtained data set and its performance was more effective than other methods in the literature.

According to the simulation results, the proposed algorithm achieved a higher Tn (true negative) value compared to other algorithms such as simple moving average (SMA), multi armed bandit (MAB) and multi-layer perceptron (MLP). This shows that the proposed algorithm is effective in successfully detecting failed tasks. Moreover, scenario-based simulation results show that the proposed algorithm works effectively in scenarios where vehicles are traveling at different speeds between moving vehicles.

In conclusion, the proposed KA-KBH algorithm for task completion mechanism, the effective use of machine learning-based offloading methods in edge computing has achieved successful results and has made a significant contribution to the literature.

4. CONCLUSION AND OUTLOOK

In VEC systems, a dynamic and heterogeneous network organization has been identified as a critical problem in terms of delivering offloaded tasks to the right computing agent. In this context, the rapidly changing needs of dynamic structures necessitate improvements in this area. The main goal of the research is to make communication between tools more efficient by adopting a three-tier VEC architecture approach. In line with this goal, three important communication options for VEC are proposed and a more effective offloading strategy is proposed by using DT and LR algorithms in a hybrid way to determine the most effective one among these options.

The proposed method can be advantageous for future VEC systems, given the number of tools used in the simulation. The edgeCloudSim simulator was used in the development of this new method and the Weka tool was evaluated in a versatile way for the accurate analysis of the data obtained. The results of this analysis show that combining machine learning approaches contributes to better results for an effective offloading selection. It is also emphasized that the proposed approach provides an effective solution for systems where packet loss rate is significant.

In the next phase of the research, the proposed approach will be extended architecturally to provide more effective solutions in denser networks. These extensions aim to improve the performance of VEC systems in more complex and large-scale environments. This study has made a significant contribution to the literature on machine learning-based offloading methods in edge computing and provides important findings that can guide future research.

1. GİRİŞ

Nesnelerin interneti (IoT), nesnelerin (örneğin cihazlar, araçlar, evler, endüstriyel ekipmanlar gibi) internete bağlanarak birbirleriyle ve diğer sistemlerle iletişim kurabildiği bir teknoloji olarak tanımlanabilir [1], [2]. Bu nesneler, çevrelerindeki fiziksel dünyayı algılayan ve veri toplayan, bu verileri işleyen mikrokontrolörler ve veri iletişimi için kablosuz veya kablolu ağ bağlantısı olan donanım ve yazılım bileşenleri ile donatılmıştır [3].

IoT teknolojisi, birçok farklı alanda kullanılabilir. Örneğin, akıllı evlerde evdeki cihazlar (örneğin termostatlar, ışıklar, kapı kilidi) internete bağlanarak birbirleriyle iletişim kurabilir ve kullanıcı tarafından uzaktan kontrol edilebilir [1], [2], [3], [4]. Endüstriyel ortamlarda ise, cihazlar ve ekipmanlar internete bağlanarak süreçleri izleyerek veri toplayabilir, böylece işletmeler daha verimli ve güvenli çalışabilir [5].

Ancak, IoT teknolojisi güvenlik konusunda önemli sorunlar doğurabilir. IoT cihazları, siber saldırılara karşı savunmasızdır ve kişisel verilerin korunması konusunda endişeler doğurmaktadır. Bu nedenle, IoT teknolojisini kullanırken, güvenlik ve gizlilik önlemlerinin alınması önemlidir[4].

Son yıllarda donanım, yazılım ve iletişim teknolojilerinde yaşanan gelişmelerle birlikte Nesne kullanımı artmaktadır. Bu artışla birlikte internete bağlı cihaz sayısı 20 milyonu geçmişken, bu cihazların oluşturduğu verilerle başa çıkmak daha da zorlaşmaktadır [6]. Fiziksel dünyanın gözlemlenmesi için kullanılan nesnelerin oluşturdukları ham verilerin iletilmesi, analiz edilmesi, anlamlandırılması ve depolanması gerekmektedir [7]. Bu gereklilik, üreticiler, tüketiciler ve hizmet sağlayıcılar için uygun koşulların oluşturulmasını gerektirir. Nesneler tarafından üretilen ham veriler çeşitli yapılarla sahiptir. Bu nedenle, veriden anlamlı sonuçlar çıkarabilmek için algoritmalar ve fiziksel ya da sanal araçlar kullanılması gerekmektedir [8].

Yeni nesil hücresele ağ (5G) teknolojisinin hayata geçmiş olması ile birlikte sunmuş olduğu avantajlar sayesinde internete bağlı olan cihaz sayısının artması ön görülebilir bir durumdur [9]. 5G teknolojisinin avantajları; bant genişliği, gecikme süresinin azlığı ve yüksek veri transfer hızıdır [10]. IoT alanında kullanılan nesnelerin 5G hücresele ağ

yapısına entegrasyonu ve avantajlarından yararlanabilmek için gerekli yazılım ve donanım ihtiyacının karşılanması gerekmektedir [11], [12]. Özellikle acil durum, sağlık veya araçların interneti (Internet of vehicle - IoV) gibi gecikme duyarlı sistemlerin üretmiş oldukları verilerde gecikme hayati nitelik taşıyabilir. Algılanan verilerin gerçek zamanlı işlenmesi, gecikme duyarlı olması açısından kenar ve bulut bilişim (KBB) çözüm olarak kullanılmaktadır [11], [12].

Ağa bağlı nesnelerin depolama, hesaplama ve batarya kapasiteleri sınırlıdır [13]. Nesnelerin kapasitelerinin daha etkili kullanımı için KBB etkili olabilir. KBB depolama veya karmaşık hesaplama konularında hiyerarşik olarak (sırasıyla IoT – Kenar - Bulut) artan bir avantaj sağlarken, iletilen verinin işlenip anlamlı sonuçlar elde edilerek nesneye veya başka bir ortama iletilme süresi (cevap süresi / response time) aynı ölçüde artmaktadır [14]. Bu nedenle, verinin sadece depolanması veya anlamlı bir sonuç çıkarılması yeterli değildir, aynı zamanda gecikme, bant genişliği ve enerji tüketimi gibi niteliklerin daha etkili kullanılması gerekmektedir. Acil durum, sağlık ve gecikme duyarlı sistemler, gerçek zamanlı çalışmaya yakın cevaplar gerektiren sistemlerdir. Bu noktada, kenar bilişim (KB) hesaplama araçları devreye girerek gerçeğe yakın cevaplar sunabilir [15]. Bu etkili çözümlerin başarılı olabilmesi için, Bulut Sistemler ile IoT arasında bulunan KB hiyerarşik olarak önemli bir rol oynayabilir [16].

Kaynak tahsisi, kenar hesaplama sistemlerindeki RAM, bant genişliği, enerji miktarı, depolama ve CPU gibi hizmetlerin talep eden sistemler arasında etkili bir şekilde paylaşılmasını amaçlamaktadır [14]. Etkili bir kaynak tahsisi yapmanın birçok faydası öngörülmektedir, bunlar arasında enerji kullanımının azaltılması, depolama kapasitesinin artırılması, veri kaybının önlenmesi, gecikmenin azaltılması ve veri analizinin daha etkili hale getirilmesi yer almaktadır [17].

IoT, sınırlı bir kaynak kapasitesine sahip olduğu için KB hesaplama araçlarına ihtiyaç duyulmaktadır. Etkili bir kaynak paylaşımı, nesnelerin daha az enerji tüketmesini ve daha hızlı cevap vermesini sağlamaktadır [4]. Tam da bu noktada, kaynak tahsisinin daha etkili olması için KB sistemleri ile çalışabilen bir sisteme ihtiyaç duyulmaktadır.

Bu ihtiyaca yönelik olarak, kaynakların daha etkin bir şekilde tahsis edilmesini sağlayan bir hiyerarşik mimari çerçeve önerilmektedir. Bu çerçeve, görev aktarımını optimize etmek ve sistem performansını artırmak için farklı seviyelerdeki kaynakları yönetme yeteneği sunmaktadır.

1.1. ÇALIŞMANIN AMACI VE ÖNERİLEN ÇÖZÜM YÖNTEMİ

Nesneler var olan kaynaklarıyla, yüksek depolama kapasitesi ve karmaşık hesaplama yapma kabiliyetinden yoksundur. Ayrıca, nesnelerin enerjileri kullanım alanlarına bağlı olarak sınırlıdır ve karmaşık hesaplamalar yapılması halinde enerji problemleri ortaya çıkacaktır [18]. Bu problemlere çözüm olarak, KBB'den yararlanmak gerekir [19], [20], [21], [22]. KB teknolojileri hiyerarşik olarak sırasıyla karmaşık hesaplamaların yapılmasına ve daha etkili depolama alanlarına sahip oldukları için IoT ile birlikte kullanılarak nesnelerin enerji tasarrufu yapmalarına yardımcı olur. Ancak, KBB hiyerarşik olarak artan bir kapasiteye sahipken, gecikme konusunda tam tersi bir durum söz konusudur [21]. Bu nedenle, cevap süresinin önemli olduğu sistemlerde Kenar Hesaplama (KH) araçlarının kullanılması etkili bir çözüm olarak görünse de yoğun IoT'de veya karmaşıklık düzeyi yüksek hesaplamalarda bu durum tersine dönebilir [23]. Tam da bu noktada, KB sistemlerinin kaynaklarını daha etkili kullanılması önemli bir çözüm yöntemi olabilir.

Bu tez çalışmasında, KBB ile çalışabilen hiyerarşik bir mimari/ çerçeve (framework) önerilmektedir. Buna göre iletilecek paket niteliğine göre bir sınıflandırma yapılarak KBB'nin her biri için bir eşik değeri belirlenmesi hedeflenmektedir. Belirlenen bu eşik değere göre her bir sistemde hesaplama araçlarının ne kadar kullanılacağı ve hangi hesaplama aracının kullanılması daha etkili bir çözüm oluşturacağına karar verilmektedir.

Önerilen model KBB ile etkileşimli hiyerarşik bir yapı olacaktır. Buna göre; hesaplama, depolama, enerji tüketimi, gecikme, bant genişliği ve kuyruk yoğunluğu gibi kriterler dikkate alınarak karar ağacı tabanlı – kenar bulut hesaplama (KA-KBH) metoduyla ilgili seçime karar verecektir.



Şekil 1. Üç katmanlı karar verici mimari.

KA-KBH, hesaplama ihtiyaçlarına göre (Kenar veya Bulut) hangi hesaplama aracının kullanılacağına karar veren bir sistemdir. Burada hesaplanacak işlemin Şekil 1’de gösterildiği gibi en hızlı yanıt verebilecek sisteme yönlendirilmesi hedeflenmektedir ve bu süreçte hesaplama kapasitesi, bant genişliği, işlem önceliği (acil durumlar için), enerji tüketim oranı ve kuyruk yoğunluğu gibi parametreler kullanılmaktadır. Hiyerarşik bir karar verici sistemin geliştirilmesinin nedenleri şunlardır:

- Acil durum ve gecikme duyarlı sistemlerde daha etkili bir cevap süresinin sağlanması ile riskli durumlara daha etkili cevap süresine çözüm sağlamak.
- Daha etkili bir iş birliğine dayanan KB sistemi ile kenar hesaplama araçlarının hem kendi içerisinde hem de hiyerarşik olarak kaynak paylaşımını sağlamak.
- Enerjisini yenileme imkânı bulunmayan nesnelerin enerjilerinin daha uzun bir süre kullanabilmeleri.
- Daha etkili bir ağ yönetim sistemi ile tek bir noktadan ağın kontrol ve yapılandırması.

Makine öğrenimi ve yapay zekâ tekniklerinin, özellikle karar ağaçları ve doğrusal regresyon gibi algoritmaların kullanıldığı, kenar hesaplama yapılarında görev aktarımı (task offloading) için bir öneri sunulmuş. Önerilen yönteme göre KA algoritması

kullanılarak sırasıyla kenar veya buluta görev aktarımı yapılır. Algoritmanın amaçları arasında olan daha etkili görev aktarımı sayesinde başarısız görevlerin azaltılması sağlanmıştır.

Makalenin ana amacı, görev gönderim hatalarını minimize etmek ve bu gönderim için en uygun hesaplama aracını seçmektir. Makine öğrenimi tabanlı bir karar ağacı modeli kullanılarak, bir görevin başarılı olup olmayacağı tahmin edildikten sonra, doğrusal regresyon ile en etkili hesaplama aracının seçimi yapılır. Algoritmanın temel adımları arasında, eğitilmiş veri seti ile algoritmanın çalıştırılması, sonuçlara göre görev aktarımı aracının seçilmesi sağlanır. KA-KBH ile önceki verilere dayanarak başarılı olup olmayacağı belirlenir görev aktarım aracı seçilmiş olur. Eğer her iki araç içinde başarısız olacağı değerlendirilirse hesaplama aracı rastgele seçilir.

1.2. TEZ ORGANİZASYONU

Bu tez çalışması ifade edilen amaç ve önerisi doğrultusunda kullanılan yöntemler ve öneriler aşağıda ifade edildiği gibi organize edilmiştir. Buna göre 2. Bölümde IoT ve KBB hakkında bilgiler verilmektedir ve önemi vurgulanmaktadır. Çalışmanın 3. Bölümünde yapılan çalışmalar detaylı incelenerek karşılaştırılmıştır. 4. Bölümde önerilen yöntem için kullanılan materyal ve metot detaylı bir şekilde anlatılmıştır. 5. Bölümde önerilen yöntem olan KA-KBH'nin çalışma prensibi anlatılmıştır. 6. Bölümde ise önerin yöntemin değerlendirilmesi yapılmıştır. 7. Bölümde ise elde edilen sonuçlar ifade edilmiştir.

2. NESNELERİN İNTERNETİ, KENAR VE BULUT BİLİŞİM

IoT, fiziksel nesnelere birbirleriyle iletişim kurma ve veri paylaşma yeteneği sayesinde tanımlanan bir teknoloji olarak öne çıkar. Bu teknolojinin kullanım alanları oldukça geniş ve çeşitli sektörlerde birçok avantaj sunmaktadır. Örneğin, sağlık sektöründe, giyilebilir cihazlar ve akıllı sağlık takip sistemleri aracılığıyla hastaların sağlık parametreleri sürekli olarak izlenebilir ve sağlık profesyonellerine gerçek zamanlı veri sağlanabilir [24].

Bulut bilişim, bilgisayar sistemleri ve kaynaklarına internet aracılığıyla erişim sağlayan ve paylaşılan bir bilgi işlem modelidir. Bu modern yaklaşım, veri depolama, bilgi işleme ve yazılım hizmetlerini genellikle internet üzerinden erişilebilen uzaktaki sunucular aracılığıyla sağlar [25]. Kullanıcılar, ihtiyaçlarına bağlı olarak bu bulut hizmetlerini istedikleri zaman ve istedikleri yerden kullanabilirler. Bulut bilişim, esneklik, ölçeklenebilirlik ve maliyet etkinliği gibi avantajlar sunarak paylaşılan kaynakları etkin bir şekilde kullanır [25].

Bulut tabanlı hizmetler genellikle farklı kategorilerde sunulur ve çeşitli ihtiyaçları karşılamak için tasarlanmıştır. Bu hizmetler depolama, veri tabanı, analitik hizmetler ve yapay zekâ gibi çeşitli alanları kapsar [26].

Kenar bilişim, bilgi işleme ve veri depolama işlemlerinin genellikle yerel cihazlarda veya ağ kenarında gerçekleştirildiği son derece önemli bir bilişim modelidir. Bu yaklaşım, işlemlerin merkezi bir bulut yerine daha yakın bir konumda gerçekleştirilmesine imkân tanıyarak; ağ trafiğini azaltmak, gecikme sürelerini en aza indirmek, bant genişliğini optimize etmek ve genel güvenliği artırmak gibi önemli avantajlar sunar. Kenar bilişim, özellikle gerçek zamanlı uygulamalarda, hızlı tepki sürelerinin kritik olduğu durumlarda büyük önem taşır. Bu model, IoT cihazları, akıllı şehir projeleri, endüstriyel otomasyon ve benzeri alanlarda sıkça tercih edilir [27]. Örneğin, bir fabrikada algılayıcılar tarafından toplanan verilerin yerel bir bilgisayar veya cihaz üzerinde işlenmesi, sadece önemli sonuçların buluta iletilmesini sağlar. Bu yaklaşım, daha hızlı cevap süreleri ve daha etkili veri yönetimi sunarak, endüstriyel süreçlerin verimliliğini artırır [28].

2.1. NESNELERİN İNTERNETİ

IoT fiziksel nesnelerin internet üzerinden birbirleriyle iletişim kurma ve veri paylaşma yeteneği ile tanımlanan bir teknolojidir [3]. IoT'nin kullanım alanları oldukça geniştir [13], [21], [24], [29].

IoT, fiziksel nesnelerin internet üzerinden birbirleriyle iletişim kurma ve veri paylaşma yeteneği ile tanımlanan bir teknoloji olarak öne çıkmaktadır [3]. Bu teknolojinin kullanım alanları oldukça geniş bir yelpazeye yayılmış durumda ve farklı sektörlerde çeşitli avantajlar sunuyor.

Sağlık sektörü, IoT'nin sunduğu olanakları değerlendirerek önemli bir dönüşüm yaşamaktadır. Giyilebilir cihazlar ve akıllı sağlık takip sistemleri sayesinde, hastaların önemli parametreleri sürekli olarak izlenebilir ve bu veriler sağlık profesyonellerine gerçek zamanlı olarak ulaştırılabilir. Bu durum, hastaların daha yakından takip edilmesini ve tedavi süreçlerinin daha etkili bir şekilde yönetilmesini sağlamaktadır [30].

Akıllı şehir uygulamaları, IoT'nin şehir planlaması alanındaki potansiyelini ortaya çıkarmaktadır. Trafik yönetimi, enerji verimliliği ve çevresel izleme gibi konularda IoT, şehirlerin daha sürdürülebilir ve akıllı bir hale gelmesine katkıda bulunmaktadır. Bu uygulamalar, şehirlerin daha etkili ve çevre dostu olmasını sağlamaktadır [31].

Tarım sektöründe, IoT'nin kullanımıyla birlikte önemli verimlilik artışları elde edilebilmektedir. Nem ölçümü, otomatik sulama sistemleri ve çiftlik hayvanlarını izleme gibi uygulamalar, tarım süreçlerini daha akıllı ve sürdürülebilir hale getirerek kaynakların daha etkin bir şekilde kullanılmasına olanak sağlamaktadır [32].

Endüstriyel IoT, fabrikalarda otomasyon, cihaz bakım öngörüsü ve üretim süreçlerinin izlenmesi gibi alanlarda avantajlar sunmaktadır. Bu sayede, üretim süreçleri daha verimli hale getirilebilir, bakım maliyetleri düşürülebilir ve üretimdeki olası aksaklıklar önceden tahmin edilebilir [33].

Son olarak, akıllı ev teknolojileri aracılığıyla IoT'nin ev yaşamına dahil edilmesi giderek artmaktadır. Güvenlik sistemleri, enerji yönetimi ve ev aletlerinin uzaktan kontrolü gibi uygulamalar, ev sahiplerine daha güvenli ve konforlu bir yaşam sunmaktadır [28], [34].

Tüm bu uygulamaların ötesinde, IoT uygulamaları ile birlikte daha fazla verimlilik, daha iyi hizmet ve yaşam koşulları sağlama imkânı sunmaktadır. Ancak, bu teknolojinin hızla gelişen yapısı ve getirdiği zorluklar, kullanıcı gizliliği, güvenlik ve standartlar gibi

konuların dikkatle ele alınması gerektirmektedir [35], [36], [37]. Bu noktalara odaklanarak, IoT'nin gelecekteki potansiyelini daha etkili bir şekilde değerlendirmek mümkündür [28].

2.2. BULUT BİLİŞİM

Bulut bilişim, bilgisayar sistemleri ve kaynaklarının paylaşılmasını öngören bir modeldir [26], [38]. Veri depolama, bilgi işleme ve yazılım hizmetlere uzaktan erişim olanağı sağlar. Bulut hizmetlerini istedikleri zaman ve istedikleri yerden kullanabilirler. Bulut bilişim paylaşılan kaynakları etkin bir şekilde kullanır [39].

2.2.1. Bulut Bilişim Modelleri

Bulut bilişim, kullanıcılara çeşitli hizmet modelleri sunar. Bu hizmet modelleri, kullanıcıların ihtiyaçlarına ve gereksinimlerine göre farklı seviyelerde kontrol ve sorumluluk sağlar. Bulut bilişimde Şekil 2'de gösterildiği gibi üç temel hizmet modeli bulunmaktadır.



Şekil 2. Bulut Bilişim Modelleri.

Altyapı Servisleri (Infrastructure as a Service - IaaS), bilişim dünyasında kullanıcılara sağlanan kapsamlı bir hizmet modelidir. Bu model, temel bilgi işlem kaynaklarını sanal bir ortamda sunarak kullanıcılara geniş bir esneklik ve özelleştirme imkânı tanır [40]. Kullanıcılar, bu hizmet modeli aracılığıyla ihtiyaçlarına uygun sanal makineler, depolama alanları ve ağ kaynakları gibi temel altyapı unsurlarını edinebilirler.

Bu kapsamda, öncü bulut bilişim sağlayıcıları, kullanıcılara geniş bir altyapı yelpazesi sunarak iş süreçlerini optimize etmelerine olanak tanır. Sanal makineler, kullanıcıların ihtiyaçlarına uygun ölçekte işlem kapasitesi sağlayarak esnek bir çalışma ortamı sunar. Depolama alanları, büyük veri setlerini güvenli bir şekilde saklamak için geniş depolama kapasitesi ve yedekleme seçenekleri sunar. Ağ kaynakları ise kullanıcıların uygulamalarını ve verilerini hızlı ve güvenli bir şekilde iletmelerini sağlar.

Özellikle bu hizmet modeli, işletmelerin kendi fiziksel altyapılarını yönetme yükünden kurtularak, hızlı bir şekilde büyümelerine ve değişen iş gereksinimlerine uyum sağlamalarına olanak tanır [40]. IaaS, bir organizasyonun altyapıya ilişkin sorumluluklarından kurtularak, daha fazla odaklanma ve kaynaklarını iş stratejilerine odaklama fırsatı sunar. Bu sayede, kullanıcılar hızlı bir şekilde ölçeklendirme yapabilir [41], maliyet etkinliği sağlar ve genel iş sürekliliğini artırır. Bu hizmet modeli, gelecekteki bilişim ihtiyaçlarını karşılamak üzere esnek ve ölçeklenebilir bir altyapı sunarak modern işletmelerin rekabet avantajını artırmalarına yardımcı olur [41].

Platform Servisleri (Platform as a Service - PaaS), bilişim teknolojilerinde bir dönüşümü temsil eden kapsamlı bir hizmet modelidir. Bu model, kullanıcılara uygulama geliştirme ve dağıtma süreçlerini büyük ölçüde kolaylaştıran bir platform sunar [40]. PaaS, geliştiricilere sadece uygulama kodlarına odaklanma özgürlüğü tanıyarak, alt yapı ve konfigürasyon detaylarıyla ilgilenme zorunluluğunu ortadan kaldırır. Bu sayede, geliştiriciler, yaratıcı süreçlerine daha fazla odaklanabilir ve uygulama yaşam döngüsünü daha etkili bir şekilde yönetebilir.

PaaS'in sunduğu bu avantajlar, özellikle uygulama geliştirme süreçlerini hızlandırmak, iş birliğini artırmak ve maliyetleri optimize etmek isteyen şirketler için önemli bir değer sunar [42]. Bu hizmet modeli, kullanıcılarına geniş bir yazılım yelpazesi sunarak, çeşitli programlama dilleri ve geliştirme ortamları üzerinde esneklik sağlar. Örneğin, Google App Engine gibi öncü PaaS sağlayıcıları, geliştiricilere uygulama kodlarını yüklemek ve çalıştırmak için gereken tüm araçları sunarak uygulama geliştirme süreçlerini büyük ölçüde basitleştirir.

Ayrıca, PaaS, ölçeklenebilirlik, güvenlik ve performans gibi önemli konularda da çözümler sunar [42]. PaaS, uygulamalarını daha geniş kitlelere yaymak veya yoğun kullanım durumlarına hızla uyum sağlama istediklerinde otomatik ölçekleme ve yük dengeleme gibi özelliklerle bu ihtiyaçlara yanıt verir [43]. Bu da şirketlerin kullanıcı taleplerine daha hızlı yanıt vermesini sağlayarak rekabet avantajı elde etmelerine olanak tanır. Sonuç olarak, PaaS, geliştiricilere daha yaratıcı ve verimli bir çalışma ortamı sunarak, modern uygulama geliştirme süreçlerindeki hız ve etkinliği artırır.

Yazılım Servisleri (Software as a Service - SaaS), kullanıcılara geniş bir uygulama yelpazesine internet üzerinden anında erişim sağlar. Bu model, geleneksel yazılım kullanımını kökten değiştirerek, kullanıcıların bir dizi uygulamaya web tarayıcıları

aracılığıyla hızlı ve güvenilir bir şekilde ulařmalarını mümkün kılar [44]. SaaS, kullanıcılara genellikle yerel kurulum ve yönetimle ilgili endişeleri ortadan kaldıran bir hizmet sunar.

Bu hizmet modeli, günlük iş süreçlerini daha verimli ve etkili hale getirme amacı güden şirketler için önemli avantajlar sunar. Google Workspace, Microsoft 365 ve Salesforce gibi öncü SaaS sağlayıcıları, kullanıcılara ofis üretkenlik araçlarından müşteri ilişkileri yönetimi (CRM) uygulamalarına kadar geniş bir yelpazede hizmet sunarak iş süreçlerini optimize etmelerine yardımcı olur. Bu uygulamalara herhangi bir yerden, herhangi bir cihazdan ve herhangi bir zamanda erişim, iş sürekliliğini artırırken esnek çalışma modellerini destekler [44].

SaaS'in sunduđu bu kolaylık, yazılım güncellemelerini otomatikleştirme, veri güvenliđi ve yedekleme gibi konularda da kullanıcılara sorunsuz bir deneyim sunar. Ayrıca, bulut tabanlı altyapısı sayesinde, kullanıcılar veri depolama ve işleme kapasitelerini kolayca ölçeklendirebilirler. Bu, özellikle hızla büyüyen işletmeler için önemli bir avantajdır çünkü altyapı kaynakları anında talep üzerine sağlanabilir [45].

Sonuç olarak, SaaS, kullanıcıların karmaşık yazılım kurulumlarına ve yönetimine harcadıkları zamandan tasarruf etmelerini sağlayarak, iş süreçlerini daha hızlı, daha verimli ve daha esnek hale getirir. Bu da şirketlerin rekabet avantajını artırarak, müşteri memnuniyetini ve iş performansını iyileştirir.

Bu hizmet modelleri, kullanıcılara farklı seviyelerde kontrol ve sorumluluk sağlar. IaaS, altyapı seviyesinde daha fazla kontrol sağlarken, PaaS uygulama geliřtirmeye daha fazla odaklanmayı ve SaaS tamamen yönetilen bir hizmeti içerir. Kullanıcılar ihtiyaçlarına bađlı olarak bu hizmet modellerinden birini veya birkaçını kullanabilirler.

2.2.2. Dađıtım Modelleri

Bulut biliřimde dađıtım modelleri, bir bulut ortamının fiziksel ve organizasyonel yapılandırmasını tanımlar. Bu modeller, kaynakların konumunu ve kullanıcılara nasıl sunulduđunu belirler. Temelde üç ana bulut dađıtım modeli bulunmaktadır [46].

Genel Bulut (Public Cloud): biliřim dünyasında geniş bir kullanıcı kitlesine açık olarak sunulan dinamik ve ölçeklenebilir bir bulut ortamını ifade eder. Bu hizmet modeli, bir hizmet sağlayıcının çeřitli müşteriler arasında kaynakları paylaştıđı ve genellikle internet üzerinden erişilebilen bir altyapı sunar. Genel bulut, kullanıcılara ölçeklenebilirlik,

maliyet etkinliđi ve hızlı dađıtım gibi bir dizi avantaj sunarak modern iřletmelerin ihtiyalarına cevap verir [47].

Öleklenebilirlik, genel bulutun en belirgin avantajlarından biridir. Kullanıcılar, ihtiyalarına göre kaynakları esnek bir şekilde öleklendirebilirler. Bu, iř yüklerinin arttıđı dönemlerde ek kaynaklara hızlı bir şekilde erişim sağlama ve gereksiz kapasiteyi azaltma esnekliđi sunar. Ayrıca, genel bulut hizmet sağlayıcıları geniş bir hizmet yelpazesi sunar, bu da kullanıcıların farklı ihtiyalarına uygun çözümleri seçmelerini mümkün kılar.

Maliyet etkinliđi, genel bulutun bir diđer önemli avantajıdır. Geleneksel altyapıya kıyasla, kullanıcılar sadece kullandıkları kaynaklar için ödeme yaparlar. Donanım alımı, bakımı ve güncellemeleri gibi maliyetli iřler hizmet sağlayıcı tarafından yönetilir. Bu, iřletmelerin başlangı maliyetlerini düşürür ve kaynakları daha etkili bir şekilde kullanmalarını sağlar.

Kullanıcılar, hizmet sağlayıcının sunduđu hızlı ve otomatik özellikler sayesinde uygulamalarını daha hızlı bir şekilde dađıtabilirler. Bu da iřletmelerin rekabet avantajını artırır ve yeni projelere daha çabuk yanıt vermelerini sağlar.

Bu bağlamda, önde gelen genel bulut hizmet sağlayıcılarından birkaçı, örneđin AWS, Microsoft Azure ve Google Cloud Platform (GCP), kullanıcılara geniş bir hizmet yelpazesi sunarak, iřletmelerin çeřitli ihtiyalarına uygun çözümler sunar. Bu geniş çeřitlilik, kullanıcıların özel ihtiyalarına uygun bulut tabanlı çözümleri seçmelerini ve iř süreçlerini daha etkili bir şekilde optimize etmelerini sağlar.

Özel Bulut (Private Cloud): biliřim alanındaki bir bulut biliřim modelidir ve genellikle belirli bir kuruluřa özgü olarak tasarlanan, yönetilen ve sürdürölen bir bulut ortamını ifade eder [46]. Bu özel bulut modeli, genellikle řirket içinde veya özel bir veri merkezinde konumlandırılır, bu da kullanıcıya tam kontrol ve özel bir altyapı sağlar. Bu modelde, biliřim kaynakları sadece belirli bir organizasyonun kullanımına açıktır, böylece kullanıcılar özel ihtiyalarına uygun olarak altyapılarını özelleřtirebilir ve yönetebilirler.

Özel bulutun birçok avantajı bulunmaktadır, bunlar arasında daha fazla kontrol, güvenlik ve özelleřtirme ön planda yer alır [46]. Daha fazla kontrol, organizasyonların biliřim kaynaklarını tamamen denetleme ve yönetme yeteneđi anlamına gelir. Bu, özellikle özel iř gereksinimleri olan kuruluřlar için kritiktir çünkü altyapının iř ihtiyalarına uygun şekilde yapılandırılması ve yönetilmesi gerekebilir.

Güvenlik açısından, özel bulut modeli, organizasyonların hassas verilerini ve uygulamalarını güvende tutmalarına yardımcı olabilir. Bu, kritik iş bilgilerinin daha güçlü bir şekilde korunmasını sağlar ve uygun güvenlik protokollerinin uygulanmasına olanak tanır. Ayrıca, özel bulutun sağladığı özelleştirme imkanları, kullanıcıların altyapılarını spesifik iş ihtiyaçlarına göre uyarlamalarını mümkün kılar, bu da daha etkili ve verimli bir bilişim ortamı oluşturabilir [46].

Bu bağlamda, özel bulut modeli, genellikle regülasyonlara tabi olan sektörlerde veya özel güvenlik gereksinimlerine sahip kuruluşlarda tercih edilmektedir. Organizasyonların özel bulut altyapısını benimsemeleri, kendi özel bulutlarını oluşturarak veya özel bulut hizmet sağlayıcılarından hizmet alarak, özel ihtiyaçlarına daha uygun, güvenli ve ölçeklendirilebilir bir bilişim ortamı sağlama yeteneği sunar.

Hybrid Bulut (Hybrid Cloud): bilişim teknolojilerindeki çeşitlenmiş ihtiyaçları karşılamak amacıyla geliştirilen bir bulut bilişim modelidir ve genel bulut ile özel bulutun avantajlarını bir araya getirir. Bu modelde, organizasyonlar kritik iş uygulamalarını ve hassas verilerini özel bulutta güvenli bir şekilde saklarken, daha ölçeklenebilir veya esnek iş yüklerini genel bulutta barındırabilirler. Hybrid bulutun sunduğu avantajlar arasında esneklik, özelleştirme ve iş yüklerini optimize etme yeteneği öne çıkar [48].

Esneklik, hybrid bulutun belki de en önemli avantajlarından biridir [46]. Organizasyonlar, iş süreçlerine özel gereksinimleri olan uygulamalarını özel bulutta tutarken, daha değişken ve ölçeklenebilir iş yüklerini genel bulutta barındırarak ihtiyaçlarına uygun bir çözüm elde ederler. Bu, organizasyonlara geniş bir adaptasyon yeteneği sunar ve değişen iş gereksinimlerine hızlı bir şekilde yanıt vermelerini sağlar.

Özelleştirme, organizasyonların iş uygulamalarını ve hizmetlerini spesifik ihtiyaçlarına göre düzenleme yeteneği anlamına gelir. Hybrid bulut, özel bulutta daha hassas verileri yönetirken, genel bulutun sunduğu geniş hizmet yelpazesi sayesinde farklı iş ihtiyaçlarına hitap eder. Bu da kullanıcıların bilişim kaynaklarını en verimli şekilde kullanmalarına ve özel ihtiyaçlarına uygun çözümler geliştirmelerine olanak tanır.

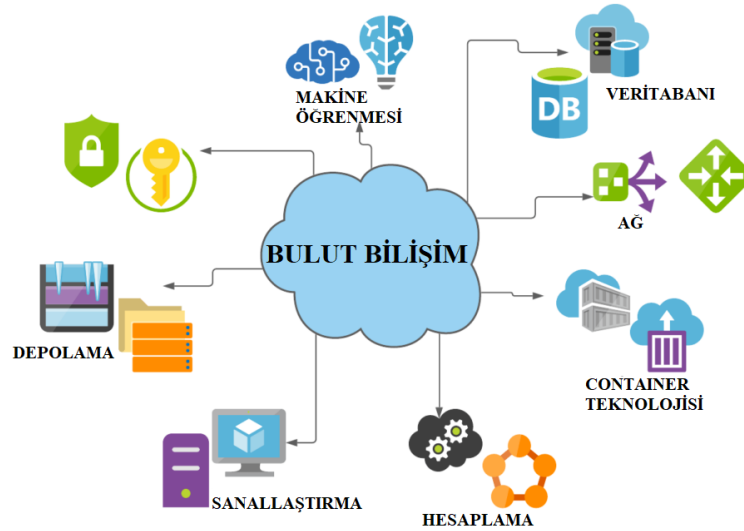
İş yüklerini optimize etme yeteneği ise hybrid bulutun performans ve maliyet avantajlarını bir araya getirir. Özel bulutta saklanması gereken kritik iş uygulamaları ile genel bulutta barındırılabilen daha yoğun ve ölçeklenebilir iş yükleri arasında bu model, iş yüklerini en uygun şekilde yöneterek kaynakları optimize eder [48].

Kuruluşların belirli iş ihtiyaçlarına yönelik bu karma (hybrid) bulut stratejileri, bilişim

altyapısını, güvenliği ve maliyet etkinliği daha etkili bir şekilde yönetmelerine olanak tanır. Bu dağıtım modelleri, organizasyonların ihtiyaçlarına, güvenlik gereksinimlerine ve veri kontrolüne göre seçilebilir. Hibrit bulut, bu iki yaklaşımı birleştirerek esneklik sağlamak için kullanılır ve işletmelere bilişim altyapılarını daha etkili bir şekilde yönetme imkanı sunar [48].

2.2.3. Bulut Tabanlı Hizmetler

Bulut tabanlı hizmetler genellikle farklı kategorilerde sunulur ve çeşitli ihtiyaçları karşılamak için tasarlanmıştır. Bu hizmetler depolama, veri tabanı, analitik hizmetler ve yapay zekâ gibi Şekil 3'te gösterilen hizmetleri kapsamaktadır. Aşağıda bu hizmetlerin birkaçı açıklanmıştır.



Şekil 3. Bulut Bilişim Hizmetleri.

Depolama Hizmetleri: Kullanıcılara çeşitli ve geniş kapsamlı veri depolama ihtiyaçlarına yönelik olarak, ölçeklenebilir ve güvenli depolama alanı sağlamak, günümüz bilişim ihtiyaçlarının karşılanması açısından kritik bir öneme sahiptir. Bu bağlamda, öncü bulut depolama hizmetleri, çeşitli dosya türleri, resimler, videolar ve diğer medya dosyaları gibi geniş bir veri yelpazesini depolamak ve yönetmek için kullanılır. Bulut depolama hizmeti sunan öncü kuruluşlar büyük miktardaki veriyi güvenli bir şekilde depolama imkânı tanır [49]. Dosyaların yanı sıra, metin belgeleri, veri tabanları ve uygulama yedekleme dosyaları gibi çeşitli veri türlerini depolamak için ideal bir çözümdür.

Öncü bulut depolama hizmetleri, kullanıcılara veri depolama konusunda geniş bir esneklik ve özellik yelpazesi sunar. Dosyaların yanı sıra büyük veri setleri, uygulama yedekleme dosyaları veya kullanıcıların özel veri depolama gereksinimlerini karşılamak için güvenli, hızlı ve ölçeklenebilir çözümler sunarlar. Bu sayede, kullanıcılar verilerini güvenle depolayabilir, yönetebilir ve ihtiyaçlarına uygun şekilde paylaşabilirler.

Veri Tabanı Hizmetleri (DBaaS - Database as a Service): Kullanıcılara geniş bir esneklik ve ölçeklenebilirlik sunan bir platform olarak, öncü veri tabanı hizmetleri, veri tabanı yönetimini daha etkili ve kullanıcı dostu bir hale getirir [50]. Kullanıcıların önceden belirlenmiş yapılandırmalar ve otomatik yedekleme özellikleri ile veri tabanlarını yönetmelerini kolaylaştırır.

Öncü platformlar kullanıcılarına veri tabanı yönetimini basitleştirirken aynı zamanda ölçeklenebilir, güvenilir ve performans odaklı çözümler sunar [50]. Bu, kullanıcıların uygulama geliştirme süreçlerini hızlandırmasına, kaynakları daha etkili kullanmasına ve iş yüklerini daha dinamik bir şekilde yönetmesine olanak tanır. Bu hizmetlerin sunduğu geniş özellik yelpazesi, kullanıcıların özel veri tabanı ihtiyaçlarına uygun çözümler geliştirmelerine ve veri yönetimi süreçlerini daha etkili bir şekilde optimize etmelerine olanak tanır.

Analitik Hizmetleri: Büyük veri analizi ve veri keşfi, günümüzde işletmelerin başarılı ve bilgi odaklı kararlar almasında kritik bir rol oynamaktadır. Bu bağlamda, BigQuery ve Snowflake gibi öncü bulut tabanlı hizmetler, kullanıcılara büyük veri setlerini etkili bir şekilde yönetme, analiz etme ve paylaşma imkânı sunarak veri odaklı iş süreçlerini güçlendirmektedir [51]. Büyük veri analizi ve veri keşfi süreçlerini optimize ederek, işletmelerin rekabet avantajı elde etmelerine ve hızla değişen pazar koşullarına uyum sağlamalarına olanak tanır [51]. Esneklik, ölçeklenebilirlik ve güvenlik açısından zengin özellik setleriyle, işletmelerin ihtiyaçlarına uygun çözümler sunarak bilgi yönetimi süreçlerini daha etkili bir hale getirir.

Yapay Zekâ ve Makine Öğrenmesi Hizmetleri: Makine öğrenimi ve yapay zekâ alanındaki ilerlemeler, bulut bilişim platformlarının kullanıcılarına daha etkili ve hızlı çözümler sunmasını sağlamıştır. Bu bağlamda, bulut tabanlı hizmetlerin sunduğu önceden eğitilmiş modellerle makine öğrenimi ve yapay zekâ uygulamalarını hızlandırma kapasitesi giderek artmaktadır [22], [52]. Görüntü işleme, metin analizi, konuşma tanıma gibi bir dizi özellik sunarak kullanıcılara geniş bir yapay zekâ yelpazesi sağlar. Bu

hizmetler, kullanıcıların karmaşık veri setlerini anlamalarına, dil işleme yetenekleriyle metin analizi yapmalarına ve ses verilerini etkili bir şekilde işlemelerine olanak tanır.

Bu bahsedilen hizmetler, bulut bilişimdeki kilit hizmetler arasında önemli bir konumda bulunmaktadır. Kullanıcılar, ihtiyaçlarına ve iş gereksinimlerine uygun olarak bu hizmetleri seçebilir ve çeşitli iş senaryolarını destekleyebilir. Her bir hizmet, endüstri standartlarında güvenlik ve performans sağlar ve yapay zekâ alanında gelişmeleri takip eden kullanıcılara rekabet avantajı sunar. Bu hizmetler sayesinde, kullanıcılar karmaşık yapay zekâ uygulamalarını hızlı bir şekilde geliştirebilir, eğitebilir ve dağıtabilir, böylece iş süreçlerini daha akıllı ve verimli hale getirebilirler [53].

2.3. KENAR BİLİŞİM

Kenar bilişim, bilgi işleme ve veri depolama işlemlerinin, genellikle yerel cihazlarda veya ağ kenarında gerçekleştirildiği son derece önemli bir bilişim modelidir [54], [55]. Bu benzersiz yaklaşım, işlemlerin merkezi bir bulut yerine daha yakın bir konumda gerçekleştirilmesine imkan tanıyarak, bir çok avantaj sunar. Kenar bilişim, özellikle ağ trafiğini azaltmak, gecikme sürelerini minimuma indirmek, bant genişliğini optimize etmek ve genel güvenliği artırmak gibi önemli avantajlara sahiptir [56], [57].

Bu model, özellikle gerçek zamanlı uygulamalarda, hızlı tepki sürelerinin kritik olduğu durumlarda büyük önem taşımaktadır [55]. Kenar bilişim, IoT cihazları, akıllı şehir projeleri, endüstriyel otomasyon ve benzeri alanlarda sıkça tercih edilir [54]. Örneğin, bir fabrikada nesnelere tarafından toplanan verilerin yerel bir bilgisayar veya cihaz üzerinde işlenmesi, sadece önemli sonuçların buluta iletilmesini sağlar. Bu yaklaşım, daha hızlı yanıt süreleri ve daha etkili veri yönetimi sunarak, endüstriyel süreçlerin verimliliğini artırır [58].

Kenar bilişim, bulut bilişimle birlikte kullanılarak, her iki modelin avantajlarını birleştiren hibrid bilişim çözümlerini oluşturabilir. Bu, hem yerel cihazlarda hem de bulutta işlemlerin gerçekleştirilmesini içerir ve bu çeşitlilik, bir çok iş ihtiyaçlarına uygun esnek çözümler elde etmeyi mümkün kılar. Bu entegrasyon, özellikle büyük veri analitiği, yapay zekâ uygulamaları ve çeşitli endüstriyel uygulamalarda beklenen performans, güvenlik ve verimlilik sağlamak için önemli bir strateji haline gelir [59].

Kenar bilişimdeki kenar cihazları ve donanımlar, veri işleme, depolama ve iletişim yetenekleri ile donatılmış cihazları ifade eder. Bu cihazlar, verilerin üretildiği veya

tüketildiği yerlere daha yakın konumlandırılır, bu da daha düşük gecikme süreleri, daha iyi bant genişliği kullanımı ve daha hızlı yanıt süreleri sağlar [54], [56], [57], [60].

Akıllı nesnelere, akıllı kameralar, akıllı araçlar, kenar sunucuları, IoT cihazları ve ağ kenarı yönlendiricileri ve anahtarları Şekil 4’te gösterildiği gibi bu kapsamda ifade edilebilir.



Şekil 4. Kenar hesaplama sistemleri ve IoT ilişkisi.

Bu cihazlar ve donanımlar, KB uygulamalarının çeşitli ihtiyaçlarını karşılamak üzere tasarlanmıştır. İhtiyaca bağlı olarak, farklı sektörlerde ve uygulama senaryolarında kullanılırlar.

2.3.1. Kenar Bilişim Mimarileri

Kenar bilişim, bilgi işleme ve veri depolama işlemlerinin, veri kaynaklarına daha yakın konumlandırılan cihazlar üzerinde gerçekleştirildiği bir bilişim modelidir. KB mimarileri, genellikle ağ trafiğini azaltmak, gecikmeyi düşürmek ve daha hızlı yanıt süreleri elde etmek amacıyla tasarlanmıştır. KB ile ilgili mimari yaklaşımlar şunlardır [56], [57], [61]:

Merkezi Bulut Mimarisi:

- Veri işleme ve depolama büyük ölçüde merkezi bir bulutta gerçekleştirilir.
- Kenar cihazları, veriyi toplar ve işler, ardından sonuçları merkezi buluta gönderir.
- Bu mimari, genellikle geniş bir coğrafi alana yayılmış cihazlarla kullanılır.

Dağıtık Kenar Mimarisi:

- Veri işleme ve depolama yetenekleri kenar cihazlarına daha fazla dağılmıştır.
- Merkezi bulut, daha çok iş akışı yönetimi ve koordinasyonu sağlar.

- Bu yaklaşım, kenar cihazlarının daha fazla özerkliğe ve işleme yeteneğine sahip olduğu durumlar için uygundur.

Merkezi Olmayan (Yıldız Topolojisi) Mimari:

- Merkezi bir bulut, çeşitli kenar cihazlarına doğrudan bağlanır.
- Her kenar cihazı, veri işleme ve depolama yeteneklerine sahip olabilir.
- Bu mimari, belirli bir merkezi kontrol ve yönetim gerektiren uygulamalar için kullanışlıdır.

Hiyerarşik Kenar Mimarisi:

- Kenar cihazları farklı seviyelerde, genellikle bir hiyerarşi içinde düzenlenmiştir.
- Ana kenar cihazları genel işleme yeteneklerine sahiptir ve ikincil kenar cihazlarına yönlendirilen işlemleri yönetir.
- Terminal kenar cihazları, daha spesifik görevleri yerine getirir ve son kullanıcıya daha yakın konumlandırılır.

KB mimarileri, farklı uygulama senaryolarına ve endüstri ihtiyaçlarına göre özelleştirilebilir. Yapılacak seçim, kullanılacak teknolojilere, cihazların yeteneklerine ve coğrafi gereksinimlere bağlı olarak değişiklik gösterebilir.

2.4. NESNELERİN İNTERNETİ, KENAR VE BULUT BİLİŞİM İLİŞKİSİ

IoT, bulut ve KB arasındaki entegrasyon, veri, işlem ve bilgi akışını etkileyen önemli bir konudur. Bilişim modelleri arasındaki etkileşim, belirli uygulama gereksinimlerine, performans hedeflerine ve güvenlik önlemlerine bağlı olarak çeşitlilik gösterir. İletişimi etkileyen ana faktörler veri akışı, koordinasyon, güvenlik, gizlilik gerçek zamanlılık, yük dengesi ve yük dağılımı olarak ifade edilir [21], [62].

Kenar cihazları, genellikle yerelde veri toplar ve ön işleme yapar. Daha sonra bu veri, ihtiyaç duyulan durumlarda buluta iletilir. Bulut, kenar cihazlarından gelen veriyi alır, daha geniş analitik işlemler ve depolama için kullanabilir [63].

İşlemlerin bir kısmı kenar cihazlarında, diğer kısmı ise bulutta gerçekleştirilebilir. Bu durumda, işlem adımları ve koordinasyonu için etkili bir iletişim mekanizması gereklidir. Örneğin, bir nesne algılama uygulamasında, kenar cihazları nesnelere tespit edebilir ve bu

bilgileri buluta ileterek daha geniş analizlere tabi tutar.

Genellikle bu iki bilişim modelinin bir arada kullanıldığı senaryolar, belirli işlemlerin ve verilerin bulut üzerinde gerçekleştirilirken, diğerlerinin yerel cihazlarda veya ağ kenarında işlendiği karmaşık sistemleri ifade eder [64]. Bu entegrasyon; performans, güvenlik ve verimlilik açısından avantajlar sunarak bilişim teknolojilerini daha etkili bir şekilde kullanmayı sağlar [65]. Özellikle IoT, akıllı şehir projeleri ve endüstriyel otomasyon gibi alanlarda, bulut bilişim ve kenar bilişimin bir arada kullanılmasıyla karma bilişim çözümleri oluşturulmaktadır. Bu hiyerarşik birliktelik, hem bulut bilişimin sağladığı geniş kapasite ve uzak erişim avantajlarını hem de kenar bilişiminin hızlı tepki, düşük gecikme ve yerel işleme gücü avantajlarını birleştirerek daha kapsamlı bir bilişim altyapısı oluşturur [20], [37], [66]. Bu sayede, endüstriler daha verimli, hızlı ve güvenli bir şekilde çalışabilirler.

Veri, kenar ve bulut arasında iletilirken güvenli bir şekilde aktarılmalıdır. Şifreleme, kimlik doğrulama ve diğer güvenlik önlemleri uygulanmalıdır. Hassas verilerin bulut ve kenar arasında geçişi sırasında gizliliği korumak kritiktir.

Kenar cihazları genellikle gerçek zamanlı uygulamalar için kullanılır. Bu durumda, kenar cihazlarından alınan verilerin ve işlemlerin hızlı bir şekilde gerçekleştirilmesi önemlidir. Bulut, daha genel işlemleri yönetir ve daha büyük depolama ve hesaplama kapasitesine sahiptir.

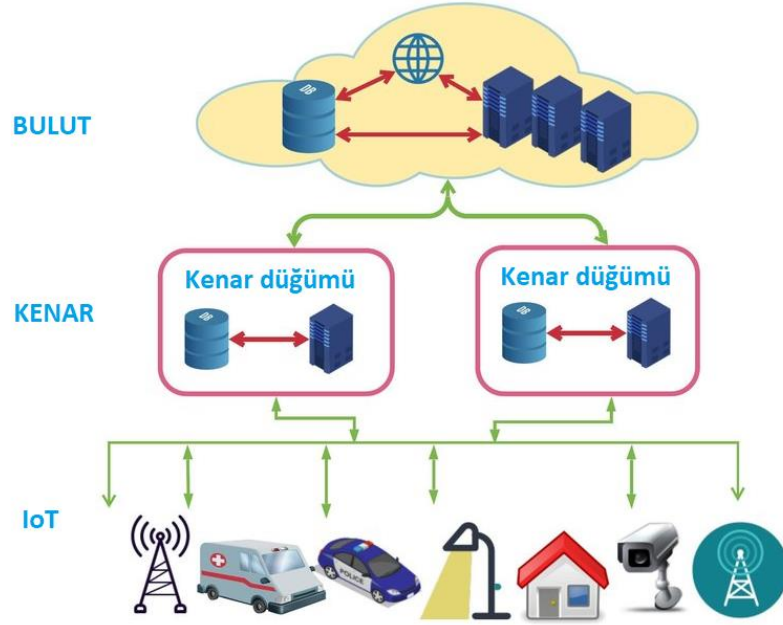
Sistemdeki değişikliklere (yeni cihazların eklenmesi, mevcut cihazların kaldırılması, ağ durumundaki değişiklikler vb.) dinamik olarak uyum sağlamak için etkili bir iletişim stratejisi gerekir.

İletişim stratejisi, bulut ve kenar arasında iş yükünü dengeli bir şekilde dağıtmalıdır. Özellikle yüksek trafik ve talep durumlarında, verimli bir görev aktarımı sağlamak önemlidir [67].

Bulut ve kenar bilişim arasındaki etkili iletişim, belirli uygulama senaryolarında en iyi performansı ve verimliliği sağlamak için tasarlanmalıdır. Bu iletişim stratejisi, uygulama gereksinimleri, donanım yetenekleri ve güvenlik standartları gibi faktörlere uyum sağlar.

2.4.1. Hiyerarşik Mimari

Hiyerarşik mimari, Şekil 5'te gösterildiği gibi üç katman dayanır: bulut katmanı, kenar katmanı ve IoT katmanı.



Şekil 5. Üç katmanlı kenar hesaplama mimarisi.

Bulut katmanının en önemli avantajları, kenar hesaplama gücünden daha fazla veri toplama, veri madenciliği, analiz optimizasyonu, depolama, toplu işleme ve karmaşık verilerin hesaplanmasıdır [68]. Bunun yanı sıra, bulut çok büyük miktarda veriyi ve karmaşık hesaplamaları çok kısa sürede hesaplayabilir. Bulut altyapısı depolama ve hesaplama olmak üzere iki bölümden oluşur. Daha sonra veya daha uzun süre kullanılacak ve gerçek zamanlı hesaplama gerektirmeyen farklı uç nesnelere aracılığıyla toplanan veriler, gelecekteki analizler için kalıcı olarak saklanır ve yazılım tabanlı ağlar (YTA) denetleyicisi aracılığıyla bulut katmanına gönderilir [69]. Hesaplama kısmı, karmaşık hesaplama görevlerini hesaplar ve analiz eder ve daha hızlı performans sunar. Kenar nesnelere aracılığıyla buluta gönderilen hesaplamalı görevler, gecikmeye duyarlı görevler değildir.

Uç bulut katmanı, IoT katmanı ile bulut katmanı arasındaki bağlantıyı sağlar. Bu amaçla, araçlarda kablosuz iletişim protokolleri olarak 802.11p, 3GPP, 3G, 4G, LTE, 5G gibi protokoller kullanılır. Bu protokoller, araçların gerçek zamanlı etkileşim ve düşük gecikme ile hizmet alabilmesi için kullanılır. Ayrıca, konum farkındalığı, acil durum yönetimi, ön belleğe alma, içerik keşfi ve hesaplama gibi hizmetler de sağlanarak hizmet kalitesi artırılır [31], [70]. Uç bulut katmanı, aynı zamanda, AR, ortam tanıma, video analitiği, sağlık tanıma ve insan davranışı tanıma gibi uygulamalarda çok düşük gecikme gerektiren hızlı yanıt gerektiren uygulamalara da hizmet verir [71]. Bu amaçla, bu katman araçlara/son kullanıcılara aşağıdaki hizmetleri sağlar:

Bu katman, araçtan araca (V2V) ve araçtan altyapıya (V2I) iletişim sağlamak için kullanılır. V2V iletişiminde, araçlar aynı iletişim menzili içinde birbirleriyle iletişim kurabilirler, böylece bilgi diğer cihazlara yayılabilir. Örneğin, bir araç anormal bir hareket gösterirse, yön değiştirme, hız limiti ihlali veya mekanik bir arıza durumunda, acil durum mesajları komşu araçlara, konum, hız ve hareket yönünü içeren bilgilerle gönderilebilir. V2I iletişiminde ise, kablosuz ağlar üzerinden yol kenarı birimleri, mikro baz istasyonları ve kenar sunucular gibi altyapılar aracılığıyla araçlar arasında operasyonel veri alışverişi gerçekleştirilir. Bu katman, birçok bileşenden oluşur ve tümü YTA üzerinden yönetilir.

Araçlar, yakın zamanda daha fazla iletişim kurma, hizmetleri paylaşma ve depolama sağlama eğilimindedir [72]. Bu bağlamda, IoT katmanı adı verilen bir grup nesne, kablosuz ağ üzerinden bilgi işlem ve depolama kaynaklarını paylaşmak için bir araya getirilir. Bu katman, gömülü nesnelere, GPS, kamera, radar ve diğer cihazlar gibi çeşitli kaynaklardan bilgi toplayarak görev yapar. Toplanan bu bilgiler, uygulama katmanında çeşitli hizmetler için girdi olarak kullanılabilir veya kenar katmanına depolanabilir. Bu katman nesnelere iletişim, depolama, yapay zekâ ve öğrenme yeteneklerine sahip olması için tasarlanmıştır ve sürücünün niyetini anlamak için bu yetenekler kullanılabilir. Bu katman, diğer nesnelere gelen hizmetleri kenar hesaplama araçları ile hesaplama ve yük dengeleme işlemlerini gerçekleştirebilir ve nesnelere için gerekli tüm hizmetleri sağlayabilir.

Bulut hizmetleri, uzaktaki bir bulutta işletilmektedir. Bu hizmetler, yüklenen bilgileri kenar sunuculardan alabilirler. Bulut hizmetleri, hesaplama ve depolama kapasitesi açısından oldukça zengindir ve daha geniş bir alana yayılabilirler. Ayrıca, mobil nesnelere ve kenar sunucular gibi diğer kaynaklardan yüklenen bilgilerle birleştirildiğinde, kapsanan alanın küresel bir görünümü elde edilebilir. Bulut modeli, küresel düzeyde yönetim ve merkezi kontrol sağlayarak, en uygun kararların alınmasına yardımcı olabilir.

RSU'lar genellikle şehirlerde yol kenarında dağıtılan VEC'lerde bulunan kenar sunucular olarak hizmet verirler. Araçlara göre zengin iletişim, hesaplama ve depolama kaynaklarına sahiptirler. RSU'lar, araçlardan gönderilen bilgileri almak, toplanan bu bilgileri işlemek ve hatta bu bilgileri buluta yüklemekle sorumludurlar. Hesaplama boşaltma ve önbelleğe alma teknolojileri sayesinde, RSU'lar katı performans gereksinimlerini karşılamak için faydalıdır. Ayrıca araçlar için video akışı, trafik kontrolü ve yol navigasyonu gibi çeşitli hizmetler sağlayabilirler.

Araç terminalleri, VEC içinde temel olarak araçlar tarafından temsil edilir. Bu araçlar, sıradan mobil cihazlardan farklı olarak, çevreyi algılamak için donanımlı araç cihazları (kamaralar, radarlar, GPS vb.) kullanarak hem içeriden hem dışarıdan bilgi toplayabilirler. Ayrıca, diğer araçlar veya RSU'larla bilgi alışverişinde bulunmak ve bilgi paylaşmak için V2V ve V2R iletişim yöntemlerini kullanabilirler. Bazı hesaplama görevlerini kenar sunuculara veya buluta aktarmanın yanı sıra, araçlar görevlerin bir kısmını yerel olarak yürütebilirler. Boş depolama alanı, araçlar için popüler içerikleri önbelleğe almak için kullanılabilir.

2.4.2. Hiyerarşik Mimarinin Avantajları

2.4.2.1. Cevap Süresi

Yanıt süresi, VEC'deki kenar sunuculara yüklenen ve geri gönderilen verilerin teslim süresiyle sunucularda işleme süresinin toplamından oluşur. Kenar sunucular, bulut tabanlı çözümlerden farklı olarak araç kullanıcılarına daha yakın konumlandırıldığından, yanıt süresi oldukça kısa olur. Bu özellik özellikle güvenlik uygulamaları gibi gecikmeye duyarlı uygulamalar için faydalıdır.

Veri depolama ve analizi için bulut bilişim katmanını tamamlayan bir kenar katmanı olarak adlandırılan araçlar, veri alanı ile bilgi işlem alanını birleştirirler. Ağ ucundan sunulan düşük veri depolama ve gecikme gerektiren hizmetler, ağ trafiğini hafifletmek ve yanıt süresini azaltmak için faydalıdır. Diğer yandan, büyük veri analizi, veri belirsizlikleriyle başa çıkmak ve gizli bilgileri ortaya çıkarmak için güçlü bir metodoloji sunar. Geçmiş verilerden öğrenilen bilgilere dayanarak, depolama, iletişim, bilgi işlem ve enerji kaynaklarının dinamik olarak ayarlanması ile denetleyici, QoS ve QoE gereksinimleri karşılanabilir. Bu, enerji verimliliği açısından önemlidir [73].

2.4.2.2. Enerji Tüketimi

Nesnelerin giderek daha yaygın hale gelmesi iletişim, bilgi işlem ve depolama gibi çeşitli nesne hizmetlerinin büyüyeceği anlamına gelmektedir. Ancak, bu hizmetlerin büyük miktarda enerji tüketmesi beklenmektedir. Sınırlı enerjiye sahip elektrikli araçlar, VEC tarafından desteklenerek, bu uygulamalar için yeterli enerji desteğini sağlayabilirler.

Enerji tüketimi, VEC sistemlerinin performansını değerlendirmek için önemli bir gösterge olduğundan, önbelleğe alma stratejileri ile enerji tüketimi arasındaki denge dikkatlice ele alınmalıdır. Diğer yandan, VEC sistemlerinin yanıt süresi optimizasyonu

gibi önbelleğe alma destekli performans değerlendirmesi genellikle uzun vadeli bir süreç ihtiyacı duyar. Bu nedenle, uygun bir uygulama/hizmet önbelleğe alma stratejisi tasarlamak VEC için hem önemli hem de oldukça zorlayıcı bir görev haline gelmektedir [74].

2.4.2.3. *Bant Genişliği*

Nesnelerin yaygınlaşması, üretilen veri miktarının hızla artmasına ve içerik taleplerinin de çeşitlenmesine neden olacaktır. Bununla birlikte, kullanıcıların uzak mesafelerde olması nedeniyle, bulut bilişim merkezi yönetimi, bu kadar büyük bir veri hacmini hesaplama ve içerik teslimi yapmak için gereken bant genişliğini garanti edemez. VEC, bulut bilişim kaynaklarını ağ uçlarına taşıyarak, geri yük ağlarına yönelik büyük bant genişliği stresini etkili bir şekilde azaltabilir. Bu nedenle, VEC, akıllı araçlar gibi yüksek veri hacmine sahip uygulamalar için verimli bir çözüm sunabilir.

VEC'nin diğer avantajları arasında, verilerin araçların yakınındaki sunuculara dağıtılarak iletişim bant genişliğinin azalması yer alır. Dağıtılmış kenar sunucular, merkezi bulut bilgi işlemine göre daha güvenilir hesaplama ve depolama imkanı sunarlar. Hem bulut sunucular hem de kenar sunucular, artan nesne sayısına ölçeklenebilir bir çözüm sunarlar. Uygulama türü ve gecikme gereksinimlerine bağlı olarak, veriler kenar veya bulut sunucularına aktarılabilir [75].

2.4.2.4. *Depolama*

VEC'nin buluttan farklı bir özelliği, verilerin son kullanıcılara yakın kenar sunucularında saklanabilmesidir. Önbelleğe alma teknolojisi sayesinde, depolanan verilere zamanında erişilebilir olması sayesinde uzak buluta depolama yükü azaltılabilir.

RSU'lar çeşitli hizmetler sağlayabileceği için MEC sunucularına göre daha fazla kaynağa sahiptirler. Bu nedenle, bölgedeki araçlara kenar bilgi işlem hizmetleri sunmanın yanı sıra, araçların hesaplama yükünü de üstlenebilirler. Tüm MEC sunucularının meşgul olduğu durumlarda, RSU'lar hesaplama görevlerini işlemek üzere kenar veri merkezine gönderirler [76].

2.4.3. Hiyerarşik Mimarideki Zorluklar

Uzaktaki cihazların etkin bir şekilde kenar bilgi işlemi yapabilmesi için, heterojen MEC (Het-MEC) sistemleri önerilmiştir. Bu sistemler, merkezi bir bulut ve birden çok kenar sunucudan oluşur. Farklı seviyelerdeki merkezi/uç bulutların koordinasyonu ve

etkileşimi, birçok yeni araştırma sorunu ortaya çıkarır. Son zamanlarda, sunucu seçimi, iş birliği ve hesaplama geçişiyle ilgili kapsamlı ilgili araştırmalar yapılmaktadır ve bunlar devam bölümünde tartışılmaktadır [77], [78], [79].

2.4.3.1. Hareketlilik

Araçların yüksek hareketliliği nedeniyle, araç ortamlarındaki ağ topolojisi oldukça dinamik bir şekilde değişebilir [33]. Bu durumda, bağlantılar kolayca kesilebilir ve iletişim kalitesi bozulabilir. Ayrıca, araçlar birden fazla kenar sunucu arasında geçiş yaparak devir teslimine neden olabilirler. Sık devir teslimler, gecikmelere neden olabilir ve hizmet sürekliliğini ters yönde etkileyerek kullanıcı deneyimini olumsuz yönde etkileyebilir.

Ağırlıklı olarak araçların hareketliliğini göz önünde bulundurduğumuzda, araçlar AP'ye yaklaştıkça iletim mesafesi azalır ve V2I iletişimde kanal veri iletim hızı artar. Bu nedenle, araçlar hesaplama görevlerini hangi zamanlarda/ne zaman/nasıl gönderileceği ve ne kadar iletişim ve hesaplama kaynağına ihtiyaç duyulacağı, başlangıç konumlarına, araçların hareket hızlarına ve görevlerin heterojen gecikme eşiklerine bağlı olarak belirlemelidir [80].

2.4.3.2. Kaynak Yönetimi

RAM, bant genişliği, enerji miktarı, depolama ve CPU gibi hizmetlerin paylaşılmasını içeren kaynak tahsisi ve kenar hesaplama sistemlerinde talepte bulunan sistemlerin etkili bir şekilde kaynakları kullanmasını sağlar [81]. Kaynak tahsisi, birçok açıdan fayda sağlamaktadır. Bu faydalar arasında enerji kullanımının azaltılması, veri kaybının önlenmesi, gecikmelerin azaltılması ve veri analizinin daha etkili bir şekilde yapılması yer almaktadır.

VEC, bulut bilgi işleme karşılaştırıldığında, hesaplama ve depolama kaynakları bakımından sınırlıdır. Bu nedenle, bu kaynakların etkili bir şekilde yönetilmesi hayati önem taşır. Ancak dinamik kaynak talepleri, farklı uygulama özellikleri ve karmaşık trafik ortamları gibi faktörler dikkate alındığında, kaynak tahsisinin optimizasyonu oldukça zorlu bir süreçtir.

2.4.3.3. Güvenlik ve Gizlilik

Araçlar, VEC'de dinamik topoloji değişiklikleri nedeniyle birbirlerine tam olarak güvenemeyebilir. Ayrıca, aynı fiziksel kenar sunuculara farklı araç kullanıcılarının

erişimine izin verildiğinde güvenlik ve gizlilik sorunları ortaya çıkabilir. Bu nedenle, güçlü bir koruma mekanizması sağlanması önemlidir.

Araç kenar ağının net bir şekilde tanımlanmış ve yaygın olarak kabul edilmiş bir mimarisi yoktur. Bu nedenle, araştırmacılar, araç kenar ağının amaçlarına, güvenlik ve gizlilik, ödül modeli, hizmet kalitesi ve sistem uygulaması gibi farklı yönlerine odaklanarak çeşitli mimari tasarımlar önermişlerdir. Ancak, mimari henüz gelişme aşamasında olduğu için güvenlik ve gizlilik konuları tam olarak ele alınmamıştır. Araç sisinin diğer ağlarla benzer güvenlik ve gizlilik sorunları olduğu yanıltıcı olabilir, ancak tamamen araç sisine özgü bazı sorunlar vardır. Sis, ağın ucunda yer alan küçük bir bulut olarak kabul edilebilir, ancak doğası gereği geleneksel bulut sorunlarının yanı sıra benzersiz güvenlik ve gizlilik sorunlarına sahip olacaktır. Bu alandaki araştırmacılar, araç sisindeki güvenlik sorunları hakkında net bir fikir edinmek için araç kenar mimarilerini ve en son güvenlik ve gizlilik araştırmalarını analiz etmektedirler. Analizlerine dayanarak, araç kenar ağının gereksinimlerini, sorunlarını ve açık sorunlarını belirlemişlerdir [82].

2.4.3.4. Görev Aktarmı

VEC'de kapasitenin kısıtlı olması nedeniyle, araç kullanıcıları hesaplama yoğunluğu yüksek ve gecikmeye duyarlı görevlerin kenar sunuculara yüklemek zorundadırlar. Ancak dinamik kanal ortamı ve sık sık değişen topoloji dikkate alındığında, görev aktarım kararlarının optimize edilmesi son derece önemlidir.

Hesaplama boşaltma, mobil bilgi işlem alanında aktif bir araştırma alanıdır ve birçok öneri iş yüklerini buluta boşaltmaktadır [67], [83]. Ancak, her zaman mümkün veya uygun olmadığından, Orsini ve diğerleri [84] CloudAware adlı bir uyarlanabilir MEC programlama çerçevesi önererek, görevleri kenar cihazlara yüklemeyi ve böylece esnek ve ölçeklenebilir kenar tabanlı mobil uygulamaların geliştirilmesini kolaylaştırmayı amaçlamaktadır. Bu çerçeve, görev aktarım hedefleri doğrultusunda hesaplama süresinin azaltılması, enerji tasarrufu, bant genişliğinden tasarruf etme veya düşük gecikme gibi farklı amaçlara ulaşmayı sağlar. KB alanındaki en temel gelişme büyük ölçekli coğrafi dağıtım ve düşük gecikme garantisinden yararlanmasıdır [85].

2.5. KENAR BİLİŞİMDE GÖREV AKTARIMI

IoT, bulut bilişim ve KB arasındaki etkileşimi optimize etmek için görev aktarımı kullanır [85]. Bu, nesnelere ve cihazlar tarafından toplanan verilerin işlenmesi, depolanması ve

analiz edilmesi sürecini optimize etmeye yardımcı olur [52].

Bulut bilişim, genellikle büyük veri merkezlerinde barındırılan güçlü sunuculara dayanır. IoT cihazları tarafından üretilen büyük miktarda veriyi işlemek ve depolamak için ideal bir ortam sunar. Ancak, bu veri miktarı zamanla arttıkça ve gerçek zamanlı tepki süreleri önemli hale geldikçe, tüm veri işleme süreçlerini buluta yüklemek pratik olmayabilir [86].

KB, verinin oluşturulduğu yerlere (örneğin, nesnelere, cihazlar) daha yakın yerel sunucular veya cihazlar kullanarak işlemleri gerçekleştirmeyi amaçlar. Bu, gecikme sürelerini azaltabilir ve ağ trafiğini azaltarak bant genişliği kullanımını optimize edebilir. Ancak, kenar cihazları genellikle sınırlı işlem gücüne ve depolama kapasitesine sahiptir [53].

Görev aktarımı, bu iki bilişim modelini akıllıca bir şekilde birleştirerek en iyi performansı sağlamayı amaçlar. Kritik ve hızlı yanıt gerektiren görevler kenar cihazlarında işlenebilirken, büyük veri analizi gibi daha karmaşık görevler buluta yönlendirilebilir [67]. Bu şekilde hem yerel işleme avantajından faydalanılırken hem de büyük veri analizi ve depolama için bulutun avantajlarından yararlanılabilir. Örneğin, bir akıllı şehir uygulamasında, trafik izleme nesnelere tarafından toplanan anlık verileri yerel olarak işleyerek hızlı bir trafik durumu güncellemesi yapabilirken, daha fazla analiz ve uzun vadeli veri depolama gereksinimleri için bu verileri buluta iletebilir. Bu strateji hem hızlı yanıt süreleri hem de veri yönetimi açısından etkili bir çözüm sunabilir.

3. NESNELERİN İNTERNETİ, KENAR BİLİŞİM VE BULUT BİLİŞİMDE KAYNAK KULLANIMI

3.1. GİRİŞ

Teknoloji alanındaki gelişmeler ve artan IoT cihazları, büyük veri ve son kullanıcı gereksinimlerini karşılamak için KBB'nin önemi giderek artmaktadır. IoT, gelişen ve büyüyen ağ yapısı nedeniyle mevcut ihtiyaçların sürekli olarak yeniden ele alınmasını gerektirir. Veri üretimi, iletimi ve işlenmesi teknolojileri, artan cihaz sayısıyla senkronize çalışması gereken sistemlerdir. Bu eş zamanlamayı sağlamak için, mevcut kaynakların etkili bir şekilde kullanılması önemlidir. Kaynak tahsisi etkili bir şekilde yapılırsa, enerji tasarrufu, bant genişliği ve gecikme gibi konulara da katkı sağlanabilir [87].

Yapılan literatür çalışmasına göre, merkezi yapıya kıyasla dağıtılmış yapı ve KB teknolojilerinin kullanımı, gecikme, enerji tasarrufu ve bant genişliği sorunlarının çözümü için daha uygun olduğu görülmüştür. Bu çalışmada, IoT alanındaki KB kavramları ve sağladıkları çözümler ele alınarak son yenilikler incelenmiştir. Ayrıca, halen devam eden sorunlar hakkında konuşulmuş ve KBB IoT ile karşılaştırmalı olarak analiz edilerek elde edilen sonuçlar açıklanmıştır.

3.2. OPTİMİZASYON TABANLI ÇALIŞMALAR

KB'nin IoT alanında daha etkili kullanılması için bazı zorluklar vardır. Tez çalışmasının bu bölümünde optimizasyon temelli zorluklar sınıflandırılarak beş başlık altında sıralanmış ve yapılan çalışmalarda zorluklara getirilen çözümler incelenmiştir. Bu zorluklar; ölçeklenebilirlik ve hareketlilik, görev aktarımı, yorumlanabilirlik, enerji tüketimi ve planlama sınıflarından oluşmaktadır [88].

3.2.1. Ölçeklenebilirlik ve Hareketlilik

KB'de kullanılan cihazların oyun teorisine göre kullanımının belirlenmesi ve gereksiz verilerin çalışma alanında çıkarılması Shah-Monsouri ve ark. [43] tarafından hedeflenmiştir. Bu hedef doğrultusunda hem hız hem de enerji konusunda iyileştirme yapıldığı ifade edilmektedir. Ayrıca hesaplama kaynaklarının çevrimiçi kullanılması da

hedeflenmiştir [43].

S.P. Abedin ve ark. [89], mobil kenar hesaplama alanında önemli eksikliklerin bulunduğunu ifade ettikleri çalışmalarında, kablosuz spektrumun mobil cihazlar arasında hesaplama için dağıtılması öngörülmüştür. Bu sayede 5G bağlamında iyileştirmeler sağlandığı görülmektedir.

Bir diğer çalışma ise, multimedia sensing as servise (MSaaS) in IoT alanında kullanılan video streaming işleminin büyük veri kategorisinde incelenerek kaynakların doğru tahsis edilmesi için gerekli optimizasyonu sağlamaktadır [90]. Buna göre QoE konusunda iyileştirme yapıldığı görülmektedir. Ayrıca uygulama katmanında veri çeşitliliği esneklik sağlamaktadır [90].

3.2.2. Görev Aktarımı

Bu alanında yapılan çalışmalar ise genel anlamda cevap verme sürelerinin düşürülmesi hedeflenmiştir. Kaynakların tahsisi ve cevap süresinin düşürülmesine yönelik çalışmada [91], verinin işlenmesi ve anlamlı hale getirilmesi için merkezi bir sunucuya göndermek yerine (bulut), kenar hesaplama cihazlarına gönderen "kenar yoğun" adında yeni bir model önerilmiştir [91]. Sağlık, akıllı ulaşım ve akıllı ev sistemleri için senaryolar geliştirilerek etkinliği üzerinde çalışmalar yapılmıştır. Cihazlara yönelik görev aktarımı konusunda genetik algoritma kullanılmıştır. Gelecek çalışmalara yönelik olarak farklı hesaplama modeli entegrasyonu sağlanması, heterojen cihazların dahil edilmesi hedeflenmiştir [91].

Başka bir çalışmada, DRAM algoritması kullanılarak statik kaynak tahsisi ve dinamik servis geçişi ile yük dengeleme işlemi yapılmaktadır. Gelecekte, dinamik servis geçişlerinin ağa olan etkisi ve servis geçişlerinin etkisi üzerine çalışma yapılması önerilmiştir [92]. Diğer çalışmada ise kenar ve bulut arasındaki iletişim için bir mimari önerilmiştir.

Alsarraf ve ark. [93] quality of service (QoS) ve service level agreement (SLA) açısından büyük verinin dağıtılması optimizasyonu ve kaynakların tahsisi konusunda daha etkili çözümler üretilmiştir [93].

S. S. Gill ve ark. [94], kritik uygulamalarda gerçek zamanlı kaynak tahsisi için parçacık sürüsü optimizasyonu algoritması kullanmışlardır. IFogSim benzetim aracı kullanılarak yapılan çalışmada, bant genişliği, enerji tüketimi, gecikme ve cevap süresi açısından daha

etkili sonuçlar elde edilmiştir. Gelecekte ise gecikme ve enerji tüketimi konuları üzerinde çalışma yapılması önerilmiştir [94].

F. Chiti [95] IoT ağların alanında yoğun hesaplama gerektiren son kullanıcı ile kenar hesaplama arasındaki görevleri yerine getiren bir algoritma önermiştir. Buna göre iletişim ve hesaplama maliyetinin en iyi şekilde kullanılması için potansiyel oyun teorisine bağlı bir algoritma geliştirilmiştir [95]. Bu algoritmaya göre en kötü toplam bekleme süresi, ortalama bekleme süresi ve görev başına bekleme süresi dikkate alınarak kaynak tahsisine karar verilir. Buna göre çalışma ile IoT tabanlı kenar hesaplamada daha az gecikme ile veri gönderimi sağlanmıştır.

3.2.3. Yorumlanabilirlik

Bu kapsamındaki çalışmalar ele alındığında veri alışverişi ve bilgi paylaşımı konuları dikkat çekmektedir. Aazam ve ark. [15] gecikme duyarlı sistemlerin ihtiyaç duyduğu hızlı cevap vermeyi çözmeye çalışmışlardır. Bu çalışma ile kaynak tahmini, kaynak tahsisi ve maliyet konusunda iyileştirme yapılmış ve benzetim olarak CloudSim toolkit kullanılmış.

Y. Liu ve ark. [79] KB'de işlenecek verilerin dağıtılmasını sağlayan bir mimari önermiş. İşlenecek verinin zamansal olarak buluta ya da kullanıcıya yakın bir sunucuda çalıştırılmasına olanak sağlamaktadır. Verinin tamamının transferi yerine iletilerek gerekli olan verinin iletilmesi oldukça önemli bir faktördür [79].

MEC, son kullanıcılar için gecikmeyi azaltmayı ve MEC'in 5G ve IoT bağlamında kullanımını kapsayan bir teknolojidir. Yapılan bir çalışmada, kullanıcı deneyimlerinden yola çıkılarak kaynak tahsisi, hareketlilik, güvenlik ve enerji konularında ilerleme sağlanması amacıyla derin öğrenme algoritmaları kullanılmıştır [96]. Ayrıca, yapay zekâ kullanılarak ağ trafiği yönetimi gerçekleştirilmiştir [97]. Bu amaçla, long short-term memory (LSTM) yöntemi ile eğitim yapılarak performans iyileştirmesi sağlanmıştır.

3.2.4. Enerji Tüketimi

Bir diğer önemli çalışma alanı olan enerji tüketimi konusunda da etkili çalışmalar yapılmış ancak hala tam anlamıyla enerji probleminin çözüme kavuşturulduğu söylenememektedir. KH alanında çalışan sistemler buluttan daha iyi sonuç verdiği görülmüş ve gelecek çalışma konusu olarak gerçek zamanlı optimizasyon teknikleri, kenar ve bulut arasındaki kaynak tahsisinin düzenlenmesi ve son olarak etkili planlama algoritmaları enerji konusunda önemli rol oynayabilir [98].

Heterojen ağlar için MEETS (maksimum enerji verimli görev zamanlaması) algoritmasını öneren Yang ve ark. [99], komşu spektrum ve enerji parametrelerini değerlendirerek en uygun parametre seçimini hedefleyen bir çalışma yürütmüştür. Bu çalışma ayrıca heterojen ağ yapısına uygulanarak, gelecek nesil (5G) teknolojisine uygunluğu artırılmış ve çeşitli senaryolarda görev zamanlaması alanında etkili sonuçlar elde edilmiştir [99].

3.2.5. Planlama

Kenar hesaplamada en çok çalışma yapılan konuların başında gelen planlama (scheduling) alanı oldukça önemli ve gecikme başta olmak üzere birçok alanda iyileştirme sağlanabilecek bir alan olarak karşımıza çıkmaktadır. Student Project allocation (SPA) game optimizasyonu kullanılarak kaynak tahsisinde ortak radio ve hesaplamalı kaynak tahsisine çözüm aranmıştır [100]. Ayrıca sistem performansı user-centric cooperation (UCC) dikkate alınarak daha fazla geliştirilmiştir.

Tiscornia ve ark. [101], QoS ve SLA bağlamında iyileştirme yapmış ve çalışmayı CloudSim benzetim aracı ile Amazon'un EC2 kaynak tahsisinde kullanarak analiz etmiştir. Bu algoritma, MeFoRE (Media fog kaynak tahmini) algoritmasına dayanmaktadır ve kaynaklardan vazgeçme hesaplamasına dayanır. Bu çalışmada [101], mikro düzeyde bedel tabanlı maliyet modeli önerilmiştir. Maliyet hesaplaması yapıldıktan sonra kaynak tahsisi yapılmakta ve çalışma ile hem kullanıcı bazında hem de sağlayıcı bazında iyileştirme yapılmıştır.

Bir başka çalışmada [102], Blockchain tabanlı ağlarda derin öğrenme ile kaynak tahsisi önerilmiştir. Çalışma dört aşamadan oluşmaktadır. İlk aşamada kaynak tahsisindeki sorunlar tespit edilmiş, ikinci aşamada ekonomik ve maliyet hesabı yapılmış, üçüncü aşamada ise sistem ve Blockchain tanımlanmış ve son olarak da derin öğrenme uygulanarak kaynak tahsisi için karar verilmiştir. Ayrıca, dinamik çalışan bir sisteme uygulanması amaçlanmıştır.

Önerilen bir başka çalışma ile otomatik kaynak tahsisi ve seçim işlemini yapay sinir ağı (YSA) modeline uygulanarak gerçekleştirilmiştir. Bu çalışma [103], Fog Resource Selection Services (FReSS) algoritması olarak adlandırılmıştır. Benzetim CloudSim ile gerçekleştirilmiştir.

Çizelge 1. Optimizasyon tabanlı çalışmaların analizi.

Kaynak	Hesaplama Araçları	Yöntem	Karşılaştırma Kriterleri	Karşılaştırma Algoritmaları	Benzetim Aracı	Makine Öğrenmesi var mı?	Katkı (%)
[43]	Sis, Bulut	Gecikme ve enerjide iyileştirme	Gecikme, CPU enerji tüketimi, QoE	Yerel Hesaplama, Remote Bulut Hesaplama, Social Optimal	-	-	%20
[104]	Sis, Bulut	Çoklu sis bilişim ile bulut arasında hiyerarşik bir kaynak paylaşımı	Kaynak kullanım oranları	-	-	-	-
[105]	Sis, Bulut	Gecikme ve bant genişliği niteliklerine göre sınıflandırılmaktadır.	Bant genişliği Gecikme	-	-	DNN (deep neural network)	-
[106]	Sis, Bulut	Etkili bir QoS için Sis ve Bulut önerilmiş.	Kaynak sayısı IoT kaynak sayısı Slot kapasitesi Slot sayısı Slot tahsis süresi Toplam gecikme	Paralel ve seri tahsis.	-	Optimizasyon	-
[107]	Kenar, Bulut	Kenar cihazının kapasitesine bakarak işlem yapılması hedeflenmektedir.	Network load Toplam gecikme	-	ifogsim	Optimizasyon	Gecikmede %36 daha etkili
[108]	Kenar, Sis ve Bulut	Daha etkili bir QoS	25 infrastrucure 8 QoS parametresi	AHP	-	Markov Desicion Process (MDP)	-
[109]	Kenar-Bulut	Acil durumlar için öncelikli görev aktarımı işlemi yapılmaktadır.	Gecikme	FIFO (first in first out)	-	-	-
[110]	Kenar, Sis ve Bulut	Kaynak kapasitesi, hesaplama kapasitesi ve verinin yaşam döngüsü gibi kriterler Dikkate alınarak stream yapılması öngörülmektedir.	Hesaplama kapasitesi Kaynak kapasitesi Verinin ömrü Gecikme	Principal Component Analysis (PCA)	Python	-	doğruluk: %91,53

3.3. MAKİNE ÖĞRENMESİ TABANLI ÇALIŞMALAR

Bu bölümde literatürde yer alan bazı derleme çalışmaları incelenerek avantaj ve dezavantajları ifade edilmiştir. Literatürde bulut ve kenar bilişimde kaynak tahsisi alanında yapılan ML algoritmaları detaylı bir şekilde analiz edilerek karşılaştırma yapılmış ve bu çalışmalar üç başlık altında toparlanmıştır.

Mijuskovic ve ark. [63] bulut ve kenar hesaplama araçları için uygulanan kaynak yönetimi teknikleri hakkında kapsamlı bir inceleme yapmışlardır. Öncelikle çeşitli teknikleri hedef ve metodolojilere göre keşif, yük dengeleme, boşaltma, dağıtım, QoS (hizmet kalitesi) ve enerji yönetimi gibi altı sınıfa ayrılarak değerlendirilir. Kaynak tahsisi, görev aktarımı, kaynak sağlama ve görev zamanlama gibi kaynak yönetimi türleri gibi nitelikleri kullanılarak analiz edilmektedir. Nayeri ve ark. [111] FC alanındaki Artificial Intelligence (AI) tabanlı çalışmaları kapsamlı bir şekilde incelemişlerdir. Çalışmada karşılaştırma kriterleri oldukça kapsamlı tutulmuş. AI tabanlı yerleştirme mekanizmalarını üç sınıfa ayırmışlardır: evrimsel tabanlı algoritmalar, makine öğrenimi tabanlı algoritmalar ve hibrid algoritmalar. Ayrıca, IoT uygulama yerleştiricilerinin güvenlik yönlerini tartışmışlar ve açık sorunları dile getirmişleridir. Shakarami ve ark. [56] üç katmanlı mimari yapıda kenar ve sis'i tek katmanda ele alarak kaynak sağlama ile ilgili kapsamlı bir çalışma yürütmüştür. Bu çalışmada [56] makine öğrenmesi başlığı altında birkaç çalışma ifade edilmiştir.

3.3.1. Denetimli Öğrenme

Denetimli öğrenme, etiketli veri setleri üzerinden desenleri öğrenen ve tahmin yapabilen bir makine öğrenimi yöntemidir ve bu öğrenme modelini kapsayan çalışmalar aşağıda ifade edilmiştir.

He ve ark. [112], akıllı şehir uygulamaları için büyük ölçekli veri analitiği hizmetleri sağlamak için çok katmanlı bir bilişim modeli önerdi. Deneyleri sunmak için, lojistik regresyon ve sunum amacıyla kullanılan DVM dahil olmak üzere iki sınıflandırma algoritması kullanıldı [112]. Performans ölçütleri, zaman, maliyet, ağ, enerji ve kaynak olmak üzere beş genel kategoriye ayrılır. Her ölçüt, o parametrenin alt kümelerini içerir. Zaman ölçütleri, yanıt süresini, zamanlama süresini, hizmet süresini, gecikmeyi, hesaplama süresini, görev yürütme süresini, gidiş-dönüş süresini, bekleme süresini, yayılma süresini ve son isabet/kaybı içerir. Yerleştirme süresi, ortalama geri dönüş süresi

ve dağıtım süresi de zaman ölçütlerinin alt kümeleridir [113].

Asim ve ark. [114] EC’de kaynak kullanımı probleminde öğrenme tabanlı çalışmaları inceleyerek değerlendirmişlerdir. Bu değerlendirme derin ve takviyeli öğrenme ile sınırlı kalmıştır.

Alsaffar ve ark. [115] VM'lerin kapasitesi, hizmet boyutu ve tamamlanma süresi gibi üç kriteri içeren doğrusal bir karar ağacı tabanlı bir algoritma önerdiler. Önerilen algoritma, kaynak tahsisi yanı sıra performansı da iyileştirmektedir [115].

Etemadi ve ark. [116], maliyeti ve gecikme gereksinimlerini artırmak için Bayes öğrenme yaklaşımının kaynak sağlama için kullanıldığı zamanla değişen iş akışları için bir öz-yönetim sistemi önerdi. Battula ve ark. [117], hizmetlerin zamanlaması, ölçeklenmesi ve taşınması için kaynak izleme hizmetleri üzerinde çalıştı ve kaynak kullanımını optimize etmek için destek ve güvene dayalı bir yöntem önerdi.

Bulut iş yükünü sınıflandırmak için Ghobaei-Arani [89] biyografi tabanlı optimizasyon şemalarını ve bulut uygulamalarının QoS değerini gözlemleyen en yakın komşu algoritması kullanıldı. Ayrıca, kaynak sağlama için uygun eylemi belirlemek için Bayes öğrenmesini ve iş yüküne en yakın sunucu kümesini seçmek için öklid mesafesini kullandı. Deng ve ark. [118], mobilkenarbilgi işlem ortamında kullanıcıların istekleri için coğrafi olarak dağıtılmış yakın nesnelere düzenlemek için yakınlık tabanlı bir görev atama şeması önermiştir. Buna göre, tahmine dayalı algoritmalarında, atama politikasını öğrenmek için Q-öğrenme yaklaşımını kullanıldığı ifade edilmiştir.

Haratyan ve ark.[119], bulanık bir kaynak yönetimi yaklaşımı olan AFRM ile CC'de kaynak tahsisini önerdiler. Her bir sanal makinenin son kaynak değerleri ortam nesnelere aracılığıyla toplanır ve bulanık bir denetleyiciye gönderilir. Daha sonra, AFRM, kaynakların yeniden tahsis edilmesi için bir karar vermek için alınan bilgileri analiz eder. Deneysel sonuçlar, AFRM'nin kaynak tahsisi verimliliği, fayda, hizmet seviyesi anlaşma ihlalleri ve maliyet açısından kural tabanlı ve statik bulanık yaklaşımlardan daha iyi performans gösterdiğini göstermektedir.

3.3.2. Denetimsiz Öğrenme

Etiketsiz veri setleri üzerinde desenleri keşfetmek ve veri içindeki yapılara dayalı olarak bilgi çıkarmak için kullanılan bir makine öğrenimi yaklaşımıdır. Bu yaklaşımı kapsayan çalışmalar aşağıda ifade edilmiştir.

Guevara ve ark. [120], veri odaklı akıllı uygulamaları etkinleştirmek amacıyla ağ kenar hesaplama, iletişim, depolama ve güç kaynaklarını entegre eden bir hizmet altyapısı sunan Edge AI alanındaki bir çalışma yürütmüşlerdir [60]

Talaat ve ark. [121], gerçek zamanlı sistemler için Bulanık ve Olasılıksal Sinir Ağlarına dayalı etkili yük dengeleme stratejisi (ELBS) önerdiler. Önerilen ELBS, sağlık uygulamaları gibi bulut ve kenar katmanları arasında verimli bağlantı sağlayan sistemler için uygun olduğu ifade edilmektedir. Fuzzy mantığının kullanımının ana nedeni, hızlı önceliklendirme gereksiniminden kaynaklanmaktadır. Önceden tanımlanmış öncelik, son teslim süresi ve görev boyutunu içeren üç parametre, bulanık sistemin girdileri ve yeni öncelik, bulanık sistemin çıktısı olduğu ifade edilmiştir. Olasılıksal sinir ağları, geçerli tahmin edilen hedefler üretir ve çok katmanlı nesne sinir ağlarından daha hızlı olduğundan ELBS stratejisinde kullanılır. ELBS, kritik görevi yeni bir sunucuya yeniden zamanlayarak ve bu görev için uygun ana bilgisayar bulmak için görevi geçirerek beklenmedik yüksek yük gerçek zamanlı görev hatalarını yönettiği ifade edilmiştir.

Li ve ark. [122] kenar ve bulut hesaplama sistemlerinde yük dalgalanmalarına odaklandı ve bu tür ortamlar için düşük maliyetli bir kaynak ölçeklendirme yöntemi önerdi. Ölçeklendirme politikasını sunmak için derin inanç ağı tabanlı (DBN) görev aktarımı tahminlerini ve iki aşamalı planlama yapıldığı ifade edilmiştir.

Al-Makhadmeh ve ark. [123], mevcut kaynaklara göre kullanıcı yoğunluğunu ve taleplerini dengelemek için derin öğrenmeye dayalı bir kaynak tahsisi ve ölçeklenebilirlik yapısı sunulmuştur. Buna göre algoritmanın yoğun sistemlerde etkili sonuçları olduğu ifade edilmektedir.

Luo ve ark. [9] MEC'de enerji tüketimini en aza indirmek için GA tabanlı bir nakit ödeme yerleştirme stratejisi önerdi. Enerji tüketimi, ana taşıyıcı kapasiteleri ve içerik popülerlik dağılımları dikkate alınarak ortak bir optimizasyon problemi formüle edilmiştir. Bu karmaşık ortak optimizasyon problemini çözmek için bir GA uygulanır. Benzetim sonuçları, önerilen algoritmanın optimuma yakın önbelleğe alma yerleşimini etkili bir şekilde belirlediğini ve geleneksel önbelleğe alma yerleştirme stratejilerine kıyasla enerji verimliliği açısından daha iyi performans elde ettiğini göstermektedir.

Xiong ve ark. [124] ise IoT'nin kaynak tahsisi için MDP olarak formüle edilen bir DRL tabanlı bir yaklaşım önerdiler. Ayrıca, çoklu tekrar hafızaları kullanarak küçük karşılıklı

etkileşimlerle deneyimleri ayrı ayrı depolayan ilke öğrenimi için bir derin Q-ağ algoritması geliştirdiler. Benzetim sonuçları, önerilen algoritmanın orijinal derin Q-ağ algoritmasından daha iyi performans gösterdiğini ve karşılık gelen politikanın tamamlanma süresiyle ilgili diğer politikalardan daha iyi performans gösterdiğini gösterdi.

Wang ve Su [87], bulut ortamındaki nesnelere arasındaki geniş iletişim yelpazesine başa çıkmak için bir kaynak tahsis algoritması önerdi. Algoritma, hesaplama yeteneği ve depolama faktörlerine göre işleri ve nesnelere dinamik olarak dağıtmak için tanıma teknolojisini kullanır ve dinamik hiyerarşi kullandığı için kaynakları tahsis ederken trafiği azaltabilir.

3.3.3. Pekiştirmeli Öğrenme

Tang ve ark. [125], mobil görevler için gecikmeyi, güç tüketimini ve geçiş maliyetini en aza indirmek için yeni bir konteyner geçiş DQL modeli önerdiler. Buluşsal yöntemlere dayalı olarak, keşif aşamasında rastgele eylem seçimi optimize edildi. Ayrıca, Q ağındaki DNN eğitim stratejisi iyileştirildi.

Lee ve ark. [126] tarafından önerilen bir pekiştirmeli öğrenme algoritması, park edilmiş araçları kenar sunucuları olarak ele almaktadır ve araç uygulamalarına bilgi işlem kaynaklarını tahsis etmektedir.

Alfakih ve ark. [127], kenar sunucusundaki kaynak yönetimi sorununu çözmek için pekiştirmeli öğrenme tabanlı durum-eylem-ödül-durum-eylem (RL-SARSA) algoritması önerdi ve bu algoritma sistem maliyetini en aza indirmek için en uygun kararı vermektedir [127].

Kim ve ark. [128], enerji verimliliğini sağlayan kenar ve bulut hesaplama sistemlerindeki zorlukları ele alındığı ifade edilmiştir. Buna göre DRL tabanlı otomatik ölçeklendirme yöntemi önerilmiştir. Buna göre tüm ajanların gerçekleştirilebilir eylemlerini keşfetmek için bir tür haydut yöntemi olan epsilon-açgözlü algoritması kullanarak yerel optimumdan kaçınmayı ve en yüksek ödülleri olan eylemleri tespit ettiklerini ifade etmişlerdir.

Yang ve ark. [129], MEC ağlarında DRL tabanlı kaynak tahsisinin düşük gecikmeli iletişim için nasıl çalışabileceğini araştırdılar. Sonlu blok uzunluklu kodlarla çalışan MEC ağlarında farklı stratejiler kullanarak hesaplama kaynaklarını tahsis ettiler ve benzetim sonuçları, önerilen algoritmanın rastgele ve eşit zamanlama kıyaslamalarından daha iyi

performans gösterdiğini gösterdi.

Wang ve ark.[130] ise MEC'de bilgi işlem kaynak tahsisi ve ağ kaynağı tahsisi dahil olmak üzere akıllı kaynak tahsisleri için DRL tabanlı bir yaklaşım önerdiler. İki yönlü bir yaklaşım kullandılar: ortalama hizmet süresinin en aza indirilmesi ve kaynak tahsisinin dengelenmesi. Deneysel sonuçlar, önerilen yaklaşımın geleneksel en kısa yol algoritmasından daha iyi performans gösterdiğini ortaya koydu.

Luong ve ark.[131], KH tabanlı kaynak tahsisinde DL'ye dayalı en uygun bir açık artırma önerdi. En uygun açık artırmının analitik çözümüne dayalı çok katmanlı bir NN mimarisi oluşturdu. NN önce madencilerin tekliflerinin monoton dönüşümlerini gerçekleştirir. Ardından madenciler için tahsis ve şartlı ödeme kuralları hesaplanır. Benzetim sonuçları, önerilen şemanın hızlı bir şekilde bir çözüme yakınsayabileceğini göstermektedir.

Vimal ve ark.[132], endüstriyel IoT için MEC'de kaynak tahsisini geliştirmek için RL ve çok amaçlı ACO'nun bir hibrid kullanımını önerdi. Önerilen algoritma, MEC'deki kullanıcılar için kaynakları doğru ve uygun değer şekilde tahsis eder. Deneysel sonuçlar, önerilen algoritmanın verim açısından GA ve BA'dan daha iyi performans gösterdiğini göstermektedir.

3.4. HİBRİD ÇALIŞMALAR

Kenar hesaplama aracının kapasiteleri sınırlıdır [17] ve sınırlı olan bu kapasitelerin mümkün olan en etkili şekilde tahsis edilmesi oldukça önemlidir. Bu amaçları kapsayan birçok çalışma yapılmıştır [55], [56], [57], [133]. Özellikle bulan alanda yapılan hibrid çalışmalar daha etkili sonuçlar alınması açısından önem taşımaktadır.

Mobil kenar hesaplama (MEC) ve araçlar arası ağ (VANET) alanlarında gecikme ve maliyetin düşürülmesine yönelik çalışma gerçekleştiren Saleh ve ark. [94], Random forest (RF) algoritması kullanarak %85 daha az gecikme ve maliyet sonucu elde etmiştir. İlgili makine öğrenimi algoritmasını eğitmek için T-Drive iz veri örnek veri kümesi kullanılmıştır. Diğer bir çalışmada, Wang ve ark. [134], RF algoritmasını kullanarak gecikmenin azaltılmasını hedeflemişlerdir. Algoritmanın doğruluk oranının %90,5 olduğu ifade edilmiş ve 75 kullanıcı senaryoda elde edilen gecikme süresinin 400 ms olduğu görülmüştür. Kullanılan veri seti mobil cihazlardan alınan veriye dayanmaktadır. Zhang ve ark. [135] ise, YSA tabanlı bir çalışma yürüterek enerji tüketimi oranı ve görev gecikmesine yönelik iyileştirmeler yapmışlardır. Çok kullanıcı MEC sunucusu

senaryosunda etkili bir kaynak tahsisi yapmayı hedefleyen bu çalışmada, rastgele olarak oluşturulmuş bir veri kümesi kullanılmıştır.

Deng ve ark. [83], kenar hesaplama (KH) alanında çalışarak enerji, gecikme ve maliyet konularında iyileştirme yapmak amacıyla geliştirilmiş ağırlık bazlı particle swarm optimization (IWPSO) algoritmasını önermişlerdir. Çalışmada dikkat çeken durum, 600 MB boyutundaki bir görevin hesaplanma süresinin 75 ms olduğunun ifade edilmesidir. Elde edilen sonuçların mobil cihazlar için olmadığı, bu nedenle daha etkili sonuçlar alındığı düşünülmektedir. Ayrıca, görevlerin bölünerek iletilebileceği fikri de dikkate alınmıştır. Tang ve ark. [98], gecikme ve enerji tüketiminde iyileştirme yapmak amacıyla iki aşamalı optimizasyon (TSOM) mekanizmasını önermişlerdir. Bu hedef doğrultusunda, birinci aşamada game theory kullanılarak önceliklendirme yapılmış ve ikinci aşamada görevin işleme süresinde iyileştirme yapılmıştır. Çalışmada ifade edilen 3500 bitlik bir paket için 100 ms süre gerektiği görülmektedir. Çalışmada iFogSim kullanılmıştır.

Game theory temelli bir çalışma öneren Wang ve ark. [136], görev aktarım olasılığını hesaplayan bir algoritmaya sahiptir. Bu çalışmada, hesaplama sistemine dâhil olan her aracın faydasının en üst düzeyde tutulması amaçlanmıştır. Önerilen yaklaşımın, makine öğrenmesi yaklaşımlarına göre daha etkili sonuçlar verdiği görülmüştür. Algoritma, Bölüm 6'de detaylı olarak karşılaştırılmıştır. Wang ve ark. [137] tarafından geliştirilen, oyun teorisine dayalı bir yaklaşım, araçların görev aktarım olasılığını hesaplamak için işbirlikçi olmayan bir yaklaşım kullanarak maksimum fayda sağlamayı hedeflemektedir. Strateji, her aracın, bir diğer araçtan bağımsız olarak verilen görevi en hızlı şekilde yerine getirmesi gerektiğine dayanır. Model, üç görev aktarım yönteminden (Edge, RSU aracılığıyla bulut, GSM aracılığıyla bulut) birini seçme üzerine kuruludur. Cagatay ve ark. [138] tarafından önerilen bir diğer çalışma, ML tabanlı bir yaklaşımdır. Bu yaklaşımda, iki aşamalı bir yaklaşımla görev aktarımına karar verilir. YSA algoritması kullanılarak, daha etkili bir görev aktarımı ile görev aktarım hataları konusunda etkili bir çalışma yapılmıştır. Çalışmada, YSA doğruluk oranı yüzde 91 olarak belirlenmiştir. Ayrıca, çalışma, benzetim sonuçları bölümünde karşılaştırma algoritması olarak detaylı bir şekilde incelenmiştir. Zeng ve ark. [139] tarafından önerilen ve derin öğrenme tabanlı görev aktarım stratejisi, evrimsel sinir ağı (ESA) tabanlı bir yaklaşımla aktarım yapılacak hesaplama aracının belirlenmesini önermektedir. Bu çalışmada elde edilen sonuçlara göre, benzetim parametreleri üzerinde bazı değişiklikler yapılarak, 2200 araç

yoğunluğunda ESA tabanlı algoritmanın MLP algoritmasına [138] göre % 6,3 oranında daha başarılı görev gönderme oranına sahip olduğu tespit edilmiştir. Öte yandan, önerilen çalışmada, varsayılan benzetim parametreleri ve aynı araç yoğunluğunda MLP algoritmasından yüzde 16,19 oranında daha başarılı sonuç alınarak, görev gönderim hatalarının azaltılması sağlanmıştır.

Lin ve ark. [140] tarafından yapılan çalışma, smart healthcare systems (SHS) alanında enerji tüketimi, görev süresi ve maliyet nitelikleri konusunda iyileştirme yapmayı hedeflemiştir. Çalışmada [140], "Fruit fly optimization based task offloading (FOTO)" adı verilen bir algoritma geliştirilmiştir. 1800 kullanıcıyla bir senaryoda görev başarı oranı yüzde 81 olarak ifade edilmiştir ve CloudSim 2.0 benzetim aracı kullanılmıştır.

Görev aktarımı ve derin öğrenme (DL) nitelikli iki inceleme çalışmasından biri olan Carvalho ve ark. [52] çalışmasına göre, görev başarı oranı en yüksek algoritma DT tabanlı olan çalışmalardır. Diğer bir inceleme çalışması olan Shakarami ve ark. [53] kaynak tahsisi konusunda detaylı bir literatür taraması yapmıştır ve görev aktarım alanında eksiklikler olduğunu ve iyileştirme yapılması gereken bir alan olarak görüldüğünü belirtmiştir. Ayrıca, ML tabanlı çalışmaların çoğunda benzetim aracının ifade edilmediğine dikkat çekilmiştir.

Çizelge 2. Hibrit çalışmaların analizi

Referanslar	Veri seti	1800 nesne için başarı (%)	Benzetim süresi	Doğruluk oranı (%)	Kullanılan Algoritma	Karşılaştırılan Alg.	Yıl
[134]	Gerçek Veri Seti	NI	75 kullanıcı için 200 ms	90,5	CampEDGE	AUSP	2015
[140]	NI	81	600 request için 1.6*10 ⁴ 90 GIPS	NI	FOTO	GA-ACO	2018
[30]	NI	90	için 1 saniye 90 GIPS	NI	Game Theroy	Random	2020
[138]	edgeCloudSim	92,937	için 0,85 saniye 650 MB	91	NI	MAB	2021
[83]	NI	NI	görev için 80 ms	NI	LRBB	IWPSO	2021
[139]	edgeCloudSim	93,860	NI	NI	ESA	DMnet	2022

3.5. IOT, KENAR VE BULUT BİLİŞİMİN KARŞILAŞTIRILMASI

IoT'nin yaygınlaşmasıyla birlikte, kenar ve bulut tabanlı hesaplama araçlarının nesne tabanlı sistemlere entegrasyonu hızlanmıştır [141]. Genel olarak, KBB büyük ölçekli IoT ağlarında bilgi toplama, analiz etme, karar verme ve eylem uygulama görevlerini yerine getirmek için etkin bir rol oynayabilirler [14], [66], [142], [143]. Bulut bilişim ise bir tür veri depolama ve analiz alanı olarak düşünülebilir.

KB, temel olarak nesnelere ile bulut arasında bulunan yazılım ve donanım araçlarıdır. Bu araçlar, bulut sistemleriyle aynı görevi üstlenirler. Ancak bazı uygulama alanlarında, veri analizi veya veri aktarımının gecikme duyarlı olması ve bulut hizmetlerinin bu hizmetleri tam olarak yerine getirememesi nedeniyle ortaya çıkmışlardır [77]. Başka bir ifadeyle, kenar hesaplama, bulut işlemlerinin ağına kenarına (kullanıcıya en yakın olan fiziksel konum) yakınlaştırılması ve işlemlerin dağıtılması olarak tanımlanabilir. Sağlık alanında çalışan nesnelere verilerinin gecikmeli gitmesi hayati risk taşıyabilir ve bu tür sistemlerde etkili çözümler sunulabilirler [15], [144].

Kullanım kriterine göre sınıflandırıldığında kenar (edge) cihazlarının düşük gecikme ile veri iletimi yapması avantajlıdır. Düşük gecikme gerektiren sistemlerde kullanımı oldukça avantajlı bir durum oluşturur. Öte yandan, kaynak başına işlem süresi ele alındığında bulut (cloud) cihazlarının daha avantajlı olduğu görülmektedir [21].

Çizelge 3. IoT, Kenar ve Bulut Bilişim.

Öz Nitelik	IoT	Kenar Bilişim	Bulut Bilişim
<i>Yayımlı</i>	Dağıtık	Dağıtık	Merkezi
<i>Bileşenler</i>	Fiziksel Cihaz	Kenar Nesnelere	Sanal Kaynaklar
<i>Depolama</i>	Sınırlı	Sınırlı	Sınırsız
<i>Cevap Süresi</i>	-	Hızlı	Yavaş
<i>Konumlandırma</i>	-	Kullanıcıya Yakın	Kullanıcıya Uzak
<i>Gecikme</i>	-	Düşük	Yüksek
<i>Kaynak Başına İşlem Süresi</i>	-	Düşük	Yüksek

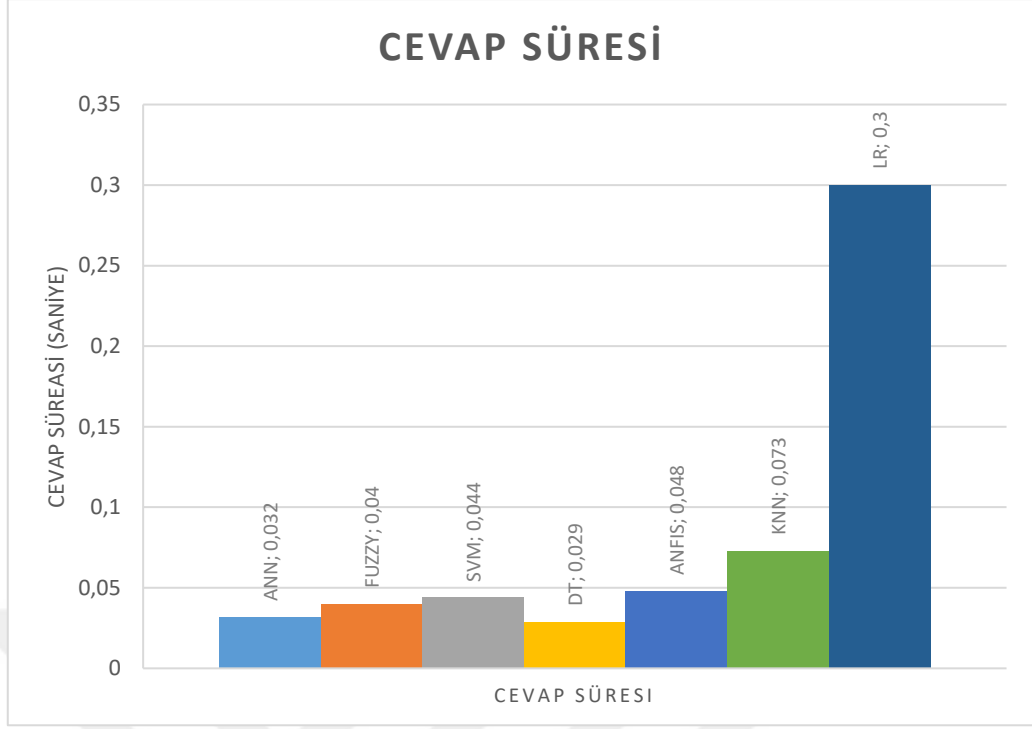
IoT'nin kullanım alanına bağlı olarak elde edilen sonuçlar Çizelge 3'te ifade getirilmiş ve belirtilen öz nitelikler için hangi sistemin daha etkili bir çözüm olabileceği belirtilmiştir.

Literatür çalışmalarından elde edilen sonuçlara göre devam eden problemler şu şekildedir:

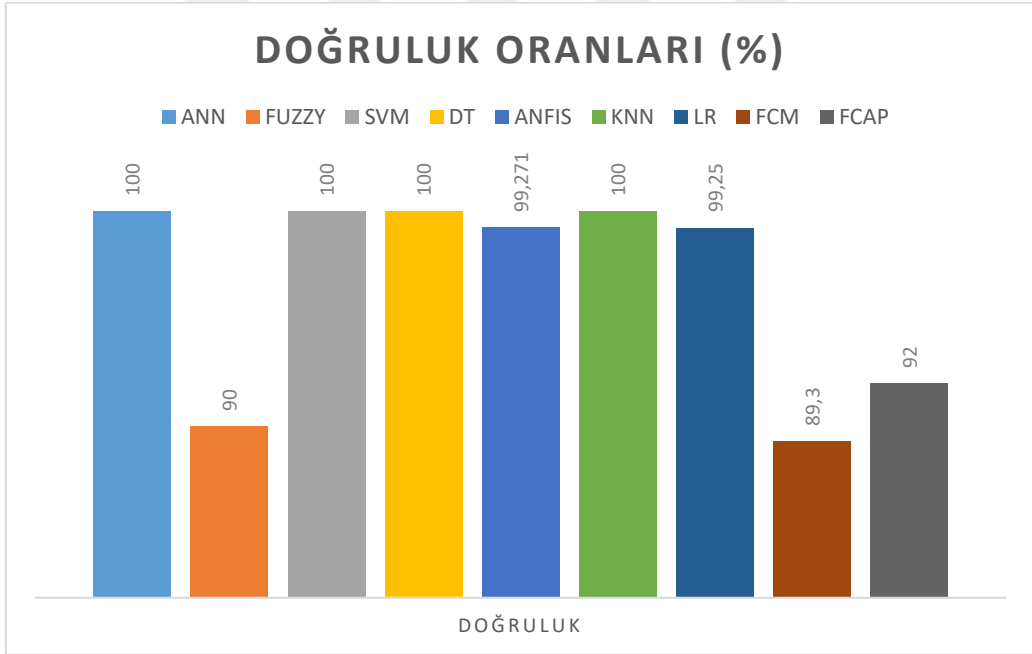
- Mobil KH'de etkili bir cevap süresi [145],
- KH'da görev aktarımı için daha etkili bir makine öğrenmesi algoritmasına ihtiyaç duyulması [145],
- BB'de güvenlik ve hesaplama için optimize edilmiş bir enerji tasarruf politikası [142].,
- MCS (Mobile crowd sensing) cihazların birbirleri ile iletişiminin sağlanması ve paylaşım yapıla bilinmesi problemi. Bu problem temelde heterojen ağ yapısı ile ilişkilendirilmektedir [61],
- Heterojen IoT ağlarında daha etkili çalışabilecek bir KB mimarisi [20],
- Hareketli bir ağ yapısına sahip olan VANET (Vehicle area networks) ve MANET (Mobile ad hoc networks) için etkili bir mobil KH önerilmesi [20],
- DoS saldırılarında yaşanacak kaynak kaybından dolayı yeni ve etkin kaynak paylaşım mimarisi [20].

Literatür kısmında ifade edilen çalışmaların bir kısmı Çizelge 2'de karşılaştırmalı olarak analiz edildi. Analiz çerçevesinde kullanılan teknik ekipmanlar, kullanılan algoritma, performans metriği, değerlendirme araçları, uygulandığı alan ve elde edilen sonuçlar ile ilgili notlar sunulmuştur. Ayrıca tabloda ifade edilen değerlendirme sonuçlarının karşılaştırılması için kullanılan fiziksel cihazların bir karşılaştırması yer almaktadır.

Tablosal değerlendirmenin yanı sıra çalışmalar Şekil 6 ve Şekil 7'de karşılaştırmalı olarak da TR ve AC açısından incelenmiştir. Bu inceleme sonucuna göre RT açısından en hızlı çalışan algoritmanın DT olduğu tespit edilirken, AC açısından bakıldığında ise birden çok algoritma en iyi sonuç olan %100 sonucunu verdiği tespit edilmiştir.



Şekil 6. ML tabanlı algortiamaların cevap süresi



Şekil 7. ML tabanlı çalışmaların doğruluk oranları

3.6. SONUÇ

Bu çalışmada, KB yapısının ve IoT alanındaki kullanımı incelenmiş ve buna göre, KB'nin hangi alanlarda daha etkili çözümler sağlayabileceği belirtilerek bir karşılaştırma yapılmıştır. Aynı zamanda, KB alanında devam eden problemler ve bu problemlere

getirilen çözümler de ele alınmıştır. KBB'in genel kullanım alanları da ele alınarak, bu alanda devam eden problemlere değinilmiştir. Bazı IoT ağları için önemli olan nitelikler; örneğin düşük gecikme, depolama, cevap süresi ve işlem süresi, KB'nin avantajlı olduğunu göstermektedir. Bu, 5G teknolojisi gereksinimlerini karşılamak açısından önemli bir etkidir. Ancak, KB'nin yaygın bir kullanımının bulunmaması, bir dezavantaj olarak görülmektedir.

EC, FC, CC, M2M (D2D), MEC, cloudlet gibi IoT kaynak tahsisi ile ilgili ana teknolojilerin belirlendiği ifade edilmektedir. IoT sistemi giderek daha karmaşık hale geldikçe ve farklı cihazlar eklenmesiyle birlikte enerji tüketimi, veri işleme ve veri çakışması gibi sorunlarla ilgili kaynak tahsisi daha zor hale gelmektedir. IoT'de yeni çözümlere ihtiyaç duyulmasından dolayı, kaynak tahsisinde birçok yeni zorluk ortaya çıkmaktadır. Kaynak keşfi ve izleme, kaynak tahsisinde önemli bir rol oynamaktadır [107]. Kaynak keşfi ve izleme, kaynak planlama ve planlama stratejisi, kaynak açıklığı ve paylaşımı ve kaynak optimizasyonu gibi faktörler gelecekteki kaynak tahsisi çalışmalarına daha fazla katkı sağlayabilir [146].

Kaynak yönetimi alanındaki problemler Ghobaei-Arani [147] tarafından altı kategori altında sınıflandırılmıştır. Bu kategoriler şunlardır: yük dengeleme, hareketlilik, zamanlama, güvenlik, enerji tüketimi ve yorumlanabilirlik. Bu çalışmanın kapsamı olan kaynak tahsisi açısından; CPU kullanımı, görev zamanlaması ve paralel zamanlama gibi alanlarda daha etkili çözümlere ihtiyaç duyulmaktadır.

Etkili bir kaynak yönetimi hala çözülmesi gereken bir problem olarak karşımıza çıkmaktadır [148]. Çizelge 2'de yer alan birçok algoritma uygulanmasına rağmen, ileride beklenen cihaz çeşitliliği ve artışı nedeniyle, kaynak tahsisinin hala bir problem olduğu görülmektedir. Sinir ağı algoritması, destek vektör makinesi algoritması, k-en yakın komşu algoritması gibi algoritmalar kullanılmasına rağmen, RA'nın hala etkili bir çözüm olmadığı anlaşılmaktadır. Yine DT, kategorik verilerin tahmin edilmesi için if-then kurallarına ve öğrenme modellerine dayalı bir sınıflandırma makine öğrenimi algoritmasıdır [66]. Karar ağacının avantajları arasında, verilerin hazırlanması için daha az ön işlem gerektirmesi, normalleştirme veya ölçeklendirme gerektirmemesi ve eksik verilerin de işlem sürecini etkilememesi bulunur. Karar ağacı yapmak ve hiper parametre ayarlamak neredeyse hiç çaba gerektirmez. Ancak, bu algoritmanın dezavantajları arasında aşırı uydurma riskinin yüksek olması, düşük tahmin doğruluğu göstermesi, sınıf etiketi sayısı arttıkça hesaplamaların karmaşıklaşması ve regresyon uygulamak için

yetersiz olması yer alır. Ayrıca tüm deęerlendirilen alıřmalarda, yanıt süresi baskın bir faktör olduęu tespit edilmiřtir.



4. MATERYAL VE METOD

Bu çalışmanın temelini oluşturan materyal ve metotlar, araştırmanın yöntemsel yaklaşımını ve veri analizi sürecini açıklamak için kullanılmıştır. Bu kapsamda önerilen yöntemde kullanılan veri seti, veri setinin normalizasyonu ve kullanılan benzetim aracına değinilmiştir.

Veri seti, görevlerin hangi hesaplama aracıyla gönderildiği, ağ gecikmeleri ve hesaplama araçlarının gecikme süreleri gibi bilgileri içeriyor. Ayrıca, önerilen yöntemin eğitiminde kullanılan veri setinin önemi vurgulanıyor.

Veri seti geniş bir içeriğe sahip olup çeşitli faktörleri ve sistem bileşenlerini kapsıyor. Başarılı görevlerin yanı sıra başarısız görevler de içeriyor ve bu, iletişim ağının güvenilirliği ve etkinliğinin değerlendirilmesine yardımcı oluyor.

Veri seti, eğitim algoritmasının çeşitli senaryolarda doğru bir şekilde öğrenmesini sağlayarak sistemin genel performansını artırıyor. Ayrıca, veri setinin dengelenmesi ve normalizasyonu gibi ön işlemler, makine öğrenimi algoritmalarının daha dengeli ve güvenilir öğrenme yeteneklerine sahip olmasına katkı sağlıyor.

EdgeCloudSim benzetim aracı, çeşitli kritik işlevleri destekleyen modüler bir mimariye sahiptir. Bu araç, çok katmanlı senaryoların benzetimini mümkün kılar ve KB senaryolarında sıkça karşılaşılan düzenleyici eylemlerini modelleme olanağı sunar.

Eğitim veri seti, EdgeCloudSim benzetim aracı tarafından üretilen veri seti kullanılarak önerilen algoritmanın eğitiminde kullanılmıştır. Deneyler, önerilen algoritmanın etkili bir şekilde çalıştığını ve daha iyi görev aktarımı seçimi ve kaynak tahsisi sağladığını göstermektedir.

Veri setinin dengeli olması ve doğru bir şekilde işlenmesi, makine öğrenimi algoritmalarının daha etkili bir şekilde çalışmasını sağlayarak sistem performansını artırır. EdgeCloudSim benzetim aracı da çeşitli senaryoların benzetimini sağlayarak sistem tasarımı ve optimizasyonunda önemli bir rol oynar.

4.1. VERİ SETİ

Önerilen yöntemin geliştirilmesinde katkısı bulunan veri seti edgeCloudSim benzetim

aracı içerisinde yer almaktadır. Veri seti içerisinde bir görevin hangi hesaplama aracı ile gönderildiği ve gönderilirken oluşan ağ gecikmeleri ve hesaplama araçlarının gecikme süreleri gibi bilgileri içermektedir.

Çizelge 4 ve Çizelge 'te ham veri setinin bir kesitine yer verilmiştir. Bu veri seti, geniş bir kapsamda ele alındığında, ağ veri iletim araçlarının gecikme süreleri ve sistem modelinde tanımlanan hesaplama araçlarının kapasitelerini içermektedir. Bu veri seti, çeşitli faktörleri ve sistem bileşenlerini içeren kapsamlı bir bilgi kaynağını temsil etmektedir. Veri setinin boyutu yaklaşık olarak 17 milyon satır veriden oluşmakta ve bu verilerin büyük bir kısmını başarılı bir şekilde tamamlanan görevler oluşturmaktadır. Gönderilen görevlerin başarı oranı, sistemdeki performans ve etkinlik açısından önemli bir göstergelyi temsil etmektedir.



Çizelge 4. Ham veri setinden bir kesit.

Decision	Result	ServiceTime	ProcessingTime	VehicleLocation	SelectedHostID	TaskLength	TaskInput	TaskOutput	WANUploadDelay	WANDownloadDelay
CLOUD_VIA_GSM	success	0,088892	0,036573	9	40	2743	21	18	0,005489	0,009696
CLOUD_VIA_RSU	success	0,118805	0,04368	13	40	3276	20	18	0,005489	0,009696
CLOUD_VIA_RSU	success	0,3429	0,267774	8	40	20084	78	19	0,005489	0,009696
CLOUD_VIA_GSM	success	0,064685	0,048836	16	40	2825	20	18	0,008094	0,002995
CLOUD_VIA_GSM	success	0,168731	0,152882	5	40	10271	19	43	0,008094	0,002995
CLOUD_VIA_RSU	success	0,217272	0,146242	5	40	10969	20	44	0,008094	0,002995
CLOUD_VIA_RSU	success	0,20315	0,13212	5	40	9909	20	43	0,008094	0,002995
CLOUD_VIA_GSM	success	0,051849	0,036	6	40	2700	19	18	0,008094	0,002995
EDGE	success	0,368907	0,298841	6	1	2989	18	21	0,008094	0,002995
EDGE	success	0,360439	0,3097	6	1	3097	22	20	0,008094	0,002995
EDGE	success	0,369294	0,299228	29	2	2993	19	21	0,008094	0,002995
CLOUD_VIA_RSU	success	0,111573	0,041266	9	40	3170	22	19	0,003207	0,007159
CLOUD_VIA_RSU	success	0,230205	0,155012	14	40	10679	19	44	0,008094	0,002995
CLOUD_VIA_RSU	success	0,131038	0,060731	23	40	3240	20	20	0,003207	0,007159
CLOUD_VIA_RSU	success	0,105011	0,043107	16	40	3233	18	21	0,003207	0,007159
CLOUD_VIA_GSM	success	0,058914	0,03956	14	40	2967	18	22	0,003207	0,007159
CLOUD_VIA_RSU	success	0,195481	0,125173	32	40	9389	19	44	0,003207	0,007159
CLOUD_VIA_RSU	success	0,123228	0,05292	24	40	3131	21	19	0,003207	0,007159
EDGE	success	0,397866	0,3278	16	2	3278	18	20	0,008094	0,002995

Çizelge 4. Ham veri setinden bir kesit (Devamı).

GSMUploadDelay	GSMDownloadDelay	WLANUploadDelay	WLANDownloadDelay	AvgEdgeUtilization	NumOffloadedTask
0,016743	0,035576	0,02323	0,036711	0	1
0,016743	0,035576	0,02323	0,036711	0,25	2
0,016743	0,035576	0,02323	0,036711	0	1
0,008544	0,007305	0,02323	0,036711	0,575	3
0,008544	0,007305	0,02323	0,036711	0,5	2
0,008544	0,007305	0,02323	0,036711	0,5	3
0,008544	0,007305	0,02323	0,036711	0,575	4
0,008544	0,007305	0,02323	0,036711	0,725	4
0,008544	0,007305	0,02323	0,036711	0,5	3
0,008544	0,007305	0,02323	0,036711	0,575	4
0,008544	0,007305	0,02323	0,036711	0,575	5
0,00811	0,011244	0,02323	0,036711	0,575	8
0,008544	0,007305	0,02323	0,036711	0,725	5
0,00811	0,011244	0,02323	0,036711	0,575	7
0,00811	0,011244	0,014827	0,036711	0,575	9
0,00811	0,011244	0,015677	0,036711	0,575	4
0,00811	0,011244	0,02323	0,036711	0,65	6
0,00811	0,011244	0,02323	0,036711	0,575	10
0,008544	0,007305	0,02323	0,036711	0,725	6

Bu veri setinde yer alan görevler, ağ veri iletim araçları tarafından taşınan verilerin ve bilgilerin başarıyla iletilmesini yansıtarak, iletişim ağının güvenilirliğini ve etkinliğini değerlendirmek için kullanılabilir zengin bir kaynak sunmaktadır. Aynı zamanda, bu veri seti, gönderilen görevlerin çeşitli başarı kriterlerini içermekte ve bu kriterler üzerinden sistem performansının analiz edilmesine imkan tanımaktadır.

Özellikle, veri setinin yaklaşık olarak %5'i başarısız görevlerden oluşmaktadır. Bu başarısız görevler, ağdaki potansiyel sorunları, kapasite sınırlarını veya iletim hatalarını yansıtarak, sistemin güvenilirliği ve kararlılığı hakkında değerli bilgiler sağlar. Başarısız görevlerin analizi, ağın veya sistem bileşenlerinin performans sorunlarını tespit etme ve çözme konusunda önemli bir rol oynar.

Bu veri setinin geniş bir içeriğe sahip olması, eğitim algoritmasının çeşitli senaryolarda ve koşullarda doğru bir şekilde öğrenmesini sağlar. Bu da, sistemin genel performansını artırarak, gelecekte karşılaşılabileceği çeşitli zorluklara daha etkili bir şekilde adapte olmasını mümkün kılar.

4.2. VERİ ÖN İŞLEME

Veri normalizasyonu, bir veri setindeki değerleri belirli bir aralığa veya standart bir dağılıma çekme işlemidir. Bu işlem, veri analizi ve makine öğrenimi uygulamalarında önemli bir adımdır [149]. Eğitim veri setinin dağılımı, özellikle eğitim aşamasında kullanılan veri setinin kalitesi ve temsili açısından oldukça önemlidir. Eğitim veri seti, genellikle başarıyla tamamlanan görevlerin örneklerini içerir, ancak bu durum, sınıflandırıcıların başarıyı tahmin etme eğilimine girmesine ve dengesiz bir örnek dağılımına neden olabilir. Özellikle, orijinal eğitim veri setindeki başarılı görevlerin sayısı başarısız görevlere göre daha fazladır.

4.2.1. Veri Seti Normalleştirme Teknikleri

Normalleştirme, veri setindeki değerleri belirli bir aralığa veya standart bir dağılıma çekme işlemidir. Bu işlem, veri analizi ve makine öğrenimi uygulamalarında yaygın olarak kullanılır. Veri setinin normalleştirilmesi, veri noktalarının dağılımını daha tutarlı hale getirir ve bazı algoritmaların daha iyi performans göstermesine yardımcı olur [150].

Normalleştirme işlemi genellikle şu adımları içerir:

Min-Max Normalleştirme: Değerleri belirli bir aralığa dönüştürmek için kullanılan bir yöntemdir. Örneğin, verileri 0 ile 1 arasında bir aralığa dönüştürebilirsiniz.

Z-Skoru Normalleştirme (Standardizasyon): Değerleri veri setinin ortalaması ve standart sapması kullanılarak standart bir normal dağılıma dönüştürmek için kullanılır. Bu yöntemde, her değerden ortalama değer çıkarılır ve standart sapmaya bölünür.

Ölçek Normalleştirme (Robust Scaling): Medyan ve çeyrekler aralığı gibi yerleşik istatistikler kullanılarak verilerin ölçeklenmesini sağlar. Bu, veri setindeki aşırı değerlerden etkilenmemek için kullanışlıdır.

Normalleştirme, veri setindeki farklı özelliklerin değerlerinin birbirine göre tutarlı hale getirilmesine ve bazı algoritmaların daha iyi çalışmasına yardımcı olabilir. Ancak, hangi normalleştirme yönteminin kullanılacağı, veri setinin özelliklerine ve uygulanacak algoritmalara bağlı olarak değişebilir.

4.2.2. Veri Setinin Matlab ile Normalleştirilmesi

Orijinal eğitim veri setinizin büyüklüğü 35 milyon örnek içerse de bu örneklerin yaklaşık 2 milyonu başarısız görevlere aittir. Bu nedenle, eğitim veri setini "başarı" ve "başarısız" görevleri dengelemek önemlidir. Bu işlem, örnek sayısını yaklaşık 4 milyona düşürür ve sırasıyla kenar, RSU aracılığıyla bulut ve CN aracılığıyla bulut senaryoları için 250 bin, 750 bin ve 1 milyon başarısız görev örneği içerir.

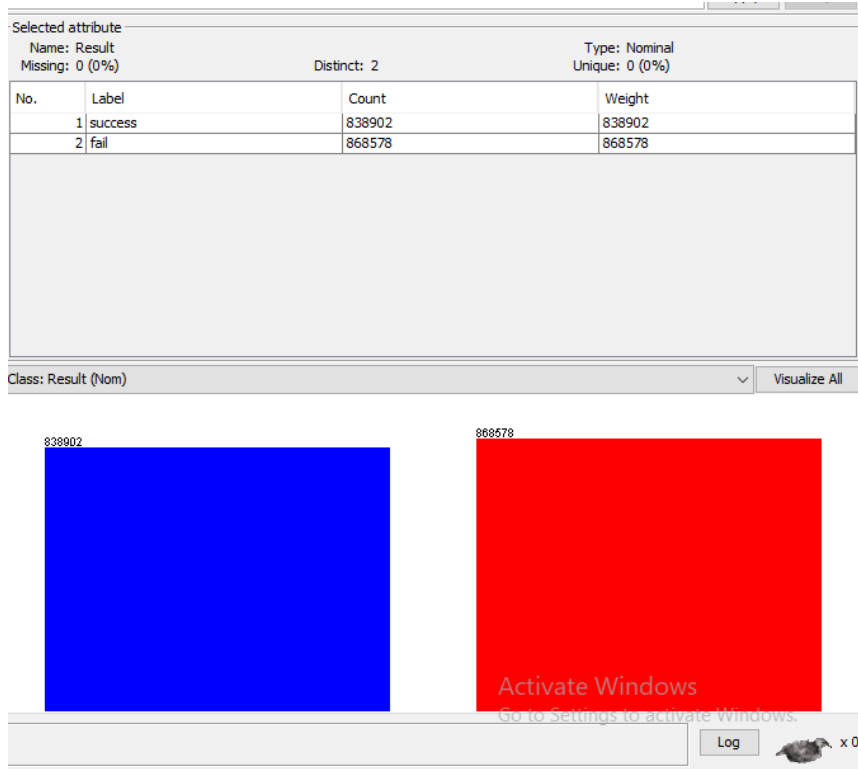
Veri normalizasyonu için ortalama normalleştirme yönetiminden yararlanılmıştır. Buna göre bir veri setindeki değerleri o veri setinin ortalaması etrafında odaklanacak şekilde normalleştirmek için kullanılan bir yöntemdir. Bu yöntem, veri setindeki değerleri ortalama değerinden çıkarak ve genellikle standart sapma ile bölerek gerçekleştirilir. Bu işlem Denklem (1)'de gösterildiği gibi 10 satırlık verinin ortalaması alınarak bir satır elde etmek için kullanılmıştır.

$$\chi' = \frac{\chi - \mu}{\sigma} \quad (1)$$

Buna göre χ' normalize edilmiş veriyi elde etmek için χ orijinal örnekten alınmış veriden μ ortalama veri çıkarılarak σ standart sapmaya bölünmesiyle elde edilmiştir.

Eğitim verilerinin dengelenmesi, özellikle Şekil 8'de görüldüğü gibi T_n oranını önemli ölçüde artırarak sınıflandırıcıların daha dengeli ve güvenilir öğrenme yeteneklerine sahip olmasına katkı sağlamaktadır. Dengelenmiş eğitim veri seti, sınıflandırıcıların başarısız

görevlerin daha etkili bir şekilde tanınmasına ve bu durumu daha iyi öğrenmesine olanak tanımaktadır.



Şekil 8. Eğitilmiş veri setinin dağılımı.

Dengelenmiş veri seti üzerinde gerçekleştirilen analizlerde, girdi ve çıktı arasındaki güçlü ilişkiyi belirlemek amacıyla farklı veri setleri test edildi. Kenar sunucuların WLAN durumu, görev süresi ve ortalama kullanımı gibi özelliklerin seçilmesi, bu analizin etkin bir şekilde gerçekleştirilmesine yardımcı oldu. Ayrıca, bulut sunucularının genellikle yeterli bilgi işlem kaynaklarına sahip olduğu düşünüldüğünden, hesaplama parametreleri bulut görev aktarımının sonucunu etkilemez. Bu nedenle, WAN ve hücresel ağ kapasitesi gibi ağ kaynakları, buluta veri aktarımı sırasında dikkate alınan kritik faktörler arasında yer alır. Bu detaylı ve dengeli eğitim veri seti, sınıflandırıcıların gerçek senaryolarda daha iyi performans sergilemesine olanak tanır.

4.3. NİTELİKLERİN SEÇİLMESİ

KA ve DR’de nitelik seçimi, modelin doğruluğunu, yorumlanmasını ve hesaplama gücünü artırmak için kritik bir öneme sahiptir [151]. Nitelik seçimi, modele dahil edilen bağımsız değişkenlerin (niteliklerin) sayısını belirlemeyi içerir. İyi bir nitelik seçimi, gereksiz veya düşük bilgi içeren niteliklerin çıkarılmasını sağlar, böylece modelin

karmaşıklığını azaltır ve aşırı uyum (overfitting) riskini azaltır. Ayrıca, nitelik seçimi, modelin daha az hesaplama gücü gerektirmesini sağlayarak hesaplama maliyetlerini azaltabilir. Sonuç olarak, doğru nitelik seçimi, daha etkili ve güvenilir tahminler sağlayabilir, modelin anlaşılmasını kolaylaştırabilir ve genel olarak modelin performansını iyileştirebilir.

4.3.1. KA Modeli İçin Niteliklerin Seçimi

Veri setinde bulunan tüm nitelikler eğitim sürecine dahi edildiğinde eğitim sürecinin uzamasına ve KA-KBH yönteminin hem doğruluğunu hem de zaman performansını etkilemekteydi. Buna çözüm bulmak adına veri setinde bulunan niteliklerden hem gecikmeyi hem de doğruluğu artırmak için bazı seçimler yapılmıştır. Buna göre KA algoritması için Çizelge 5’de yer alan nitelikler belirlenmiştir. Seçilen niteliklerin doğruluk oranları Çizelge 8’de ayrıntılı olarak verilmiştir.

Çizelge 5. KA-KBH yönteminde kullanılan veri seti nitelikleri.

Nitelik	Açıklama
TaskLength	Görev Uzunluğu
wanUploadDelay	WAN yükleme gecikmesi
wanDownloadDelay	WAN İndirme gecikmesi
gsmUploadDelay	GSM yükleme gecikmesi
gsmDownloadDelay	GSM indirme gecikmesi
wlanUploadDelay	WLAN yükleme gecikmesi
wlanDownloadDelay	WLAN indirme gecikmesi
avgEdgeUtilization	KH’nın ortalama kullanımı
meanOffTask	Ortalama görev aktarımı
Decision	Hesaplama aracının seçimi

4.3.2. DR Modeli İçin Niteliklerin Seçimi

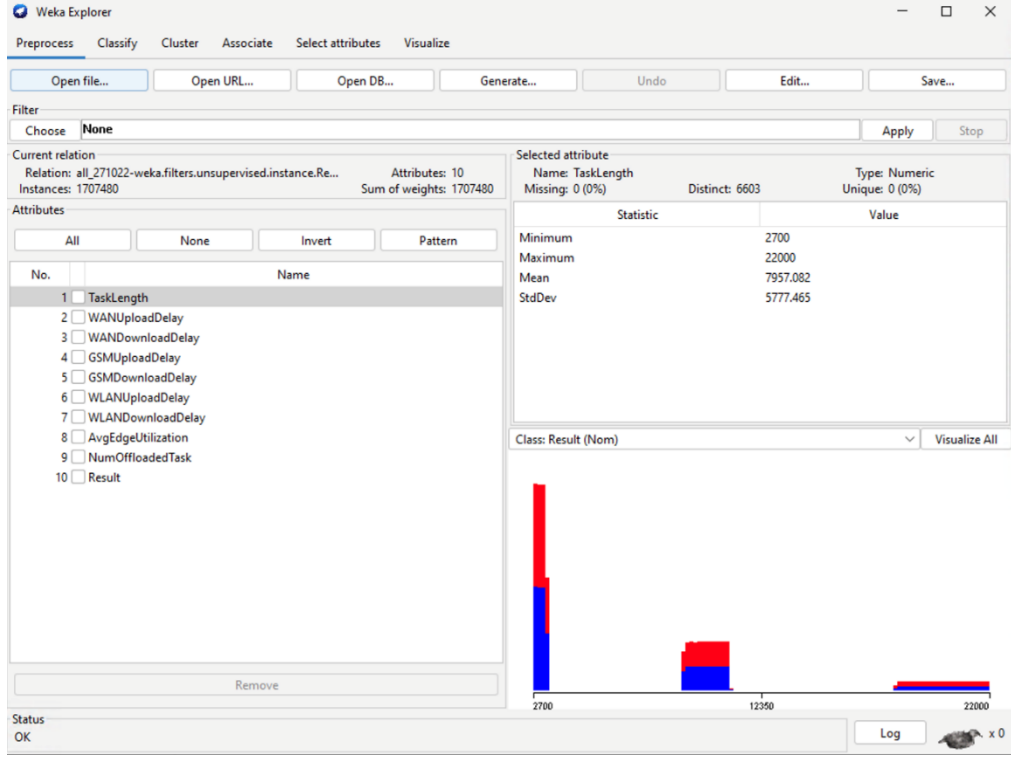
DR algoritmasının daha hızlı ve doğru çalışmasını sağlamak için belli nitelikler seçilmiştir. Bu nitelikler her hesaplama aracı için ayrı ayrı olarak ele alınmıştır dolayısıyla karar verme sürecinin hızlandırılması hedeflenmiştir. Bu doğrultuda seçilecek hesaplama aracı ve nitelikleri Çizelge 6'de detaylı olarak ifade edilmiştir.

Çizelge 6. DR algoritmasının nitelikleri.

Hesaplama aracı seçimi	Nitelik 1	Nitelik 2	Nitelik 3
KH	Görev uzunluğu	KH'nın ortalama kullanımı	-
RSU ile BH	Görev uzunluğu	WAN yükleme gecikmesi	WAN indirme gecikmesi
GSM ile BH	Görev uzunluğu	GSM yükleme gecikmesi	GSM indirme gecikmesi

4.4. WEKA ARACI İLE VERİ SETİNİN EĞİTİLMESİ

Weka (Waikato Environment for Knowledge Analysis), veri madenciliği ve makine öğrenimi alanında önde gelen bir açık kaynaklı yazılımdır [152]. İlk olarak 1993 yılında Yeni Zelanda'nın Waikato Üniversitesi'nde geliştirilmiştir. Weka'nın en önemli özelliklerinden biri, çeşitli veri madenciliği ve makine öğrenimi görevlerini gerçekleştirmek için geniş bir algoritma yelpazesine sahip olmasıdır. Sınıflandırma, kümeleme, regresyon, özellik seçimi gibi çeşitli görevler için kullanılacak birçok farklı algoritma Weka'da mevcuttur. Weka'nın Şekil 9'da gösterildiği gibi modüler ve esnek bir yapıya sahip olması da dikkat çekicidir.



Şekil 9. WEKA arayüzü.

Weka'nın bir diğer önemli avantajı, kullanıcı dostu bir arayüze sahip olmasıdır. Weka, grafik arayüzü ve komut satırı arabirimini bir arada sunar, bu da kullanıcıların tercih ettikleri şekilde programı kullanmalarını sağlar.

Önerilen hibrid yöntemde öncelikle KA algoritmasının eğitilmesi ve bu eğitim sonucunda seçilecek hesaplama aracına göre gönderilecek görevin başarılı olup olmayacağına karar verilirken ikinci aşamada ise DR yöntemi ile seçilecek aracın gecikme süresi dikkate alınarak en etkili hesaplama aracı seçimi yapılır.

4.4.1. Karar Ağacı Algoritması

EdgeCloudSim benzetim aracı tarafından üretilen veri seti, yine benzetim aracına entegre olarak kullanılan Weka [152] makine öğrenmesi aracı ile önerilen algoritmanın eğitimi gerçekleştirildi. KA algoritmasına göre en iyi nitelik seçimi için Qınlan'ın [153] önerdiği Denklem (2)'teki entropy seçimi kullanılarak kök node belirlendi.

$$Entropy(S) = - \sum_{c \in C} p(c) \log_2 p(c) \quad (2)$$

Qınlan'ın önerdiği Denklem (3), niteliklerin bilgi kazancını hesaplamak için kullanılan bir formüldür. Bilgi kazancı, KA'da nitelik azaltma sonucu elde edilir. Denklem (3), bu

bilgi kazancını hesaplamak için kullanılır. Entropy, bir durumun belirsizliğinin ölçüsüdür ve Denklem (2) kullanılarak durum entropy'si hesaplanarak bilgi kazancı hesaplanır.

$$gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3)$$

Formüle göre S veri setini temsil etmektedir. Veri setindeki her bir attribute bu sınıfa dâhildir. c ise, her bir S attribute bulunan sub classlardır. $p(c)$, c sınıfına ait veri noktalarının, kümedeki toplam veri noktalarının sayısına oranını temsil eder. Entropi değerleri 0 ile 1 arasında olabilir. S veri setindeki tüm örnekler bir sınıfa aitse, entropi sıfır olacaktır. Örneklerin yarısı bir sınıfta, diğer yarısı başka bir sınıftaysa, entropi en yüksek 1'de olacaktır. Ayrılacak en iyi özelliği seçmek ve en uygun karar ağacını bulmak için en küçük olan öznelik seçilir [154]. C4.5 algoritmasının kazancı (Weka'daki DT algoritması) Denklem (3)'te ifade edilen formülle elde edilmektedir.

4.4.2. Doğrusal Regresyon Algoritması

Doğrusal regresyon, istatistik ve makine öğrenmesinde yaygın olarak kullanılan bir tekniktir. Temel amacı, bağımlı bir değişkenin (outcome) bir veya daha fazla bağımsız değişkenle (predictor) ilişkisini modellemektir. Bu ilişkinin doğrusal olduğu varsayılır, yani bağımlı değişkenin beklenen değeri, bağımsız değişkenlerin ağırlıklı toplamı ve bir sabitin eklenmesiyle ifade edilir.

Bir doğrusal regresyon modeli, Denklem (4)'teki eşitsizlikle ifade edilir:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon \quad (4)$$

Buna göre; y , bağımlı değişkenin değerini, x_1, x_2, \dots, x_n , bağımsız değişkenlerin değerlerini, $\beta_0, \beta_1, \beta_2, \dots, \beta_n$, modelin katsayılarını (regresyon katsayıları) ve ϵ , hata terimini temsil eder.

Modelin amacı, veri setindeki bu parametreleri tahmin etmektir. Tahmin edilen değerler, gerçek gözlemler ile karşılaştırılarak modelin doğruluğu değerlendirilir. Genellikle bu doğruluk, kareler toplamı hatası (residual sum of squares) veya R^2 değeri gibi ölçütlerle değerlendirilir.

Modelin katsayıları, genellikle en küçük kareler yöntemi kullanılarak tahmin edilir. Bu yöntem, gözlemlenen değerler ile model tarafından tahmin edilen değerler arasındaki farkların karelerinin toplamını minimize ederek en uygun katsayıları bulmaya çalışır.

Doğrusal regresyon hem basit hem de çoklu bağımsız değişkenlerle çalışabilir. Basit doğrusal regresyonda, yalnızca bir bağımsız değişken kullanılırken, çoklu doğrusal regresyonda birden fazla bağımsız değişken bulunur.

Doğrusal regresyon, veri analizinde ve tahmin yapmada yaygın olarak kullanılan güçlü bir araçtır. Basit yapısı ve yorumlanabilir katsayılarıyla, ilişkileri anlamak ve geleceği tahmin etmek için geniş bir kullanım alanına sahiptir [155].

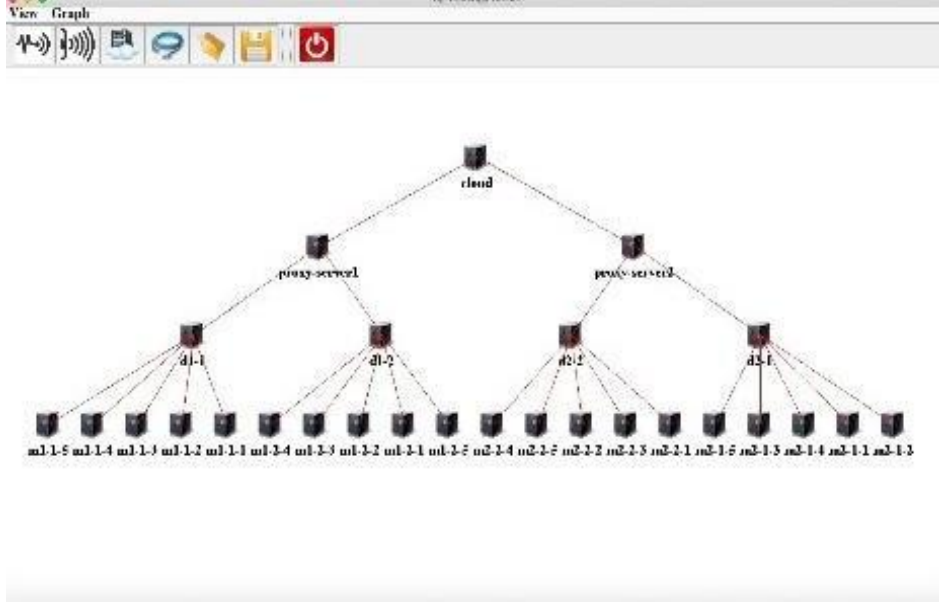
4.5. EDGE CLOUDSİM BENZETİM ARACI

EdgeCloudSim, kenar hesaplama senaryolarının simülasyonunu gerçekleştirmek için kullanılan bir araçtır. Kenar bilişim, bulut bilişim sistemlerine yakın bir konumda, kullanıcı cihazlarına veya sensörlere daha yakın olan yerel sunucular veya cihazlar aracılığıyla işlemlerin gerçekleştirilmesini sağlayan bir bilişim modelidir. Bu, uygulamaların daha hızlı yanıt vermesini, düşük gecikme sürelerini ve ağ trafiğinin azalmasını sağlar.

EdgeCloudSim, IEEE 802.11p (WLAN) ve geniş alan ağları (WAN) gibi ağ teknolojileri için özel ağ modelleriyle birlikte modüler bir mimari sunar. Bu araç, cihaz hareket modelleri ve ayarlanabilir görev aktarımı gibi çeşitli kritik işlevleri destekler. Ayrıca, çok katmanlı senaryoların simülasyonunu mümkün kılar ve KB senaryolarında sıkça karşılaşılan düzenleyici eylemleri modelleme olanağı sunar. Ayrıca simülasyon aracında gecikme metriği hesaplamaya dahil edilirken daha önceki verilerden yola çıkarak tahmini bir gecikme süresi tanımlamaktadır.

EdgeCloudSim, bulut bilişimine kıyasla kenar bilişiminin avantajlarını değerlendirmek ve KB altyapısının performansını analiz etmek için kullanılır. Ayrıca, bu simülasyon aracı, çeşitli senaryolarda ve koşullarda KB sistemlerinin tasarımını ve optimizasyonunu yapmak için kullanılabilir.

KB altyapısının performansını ölçmek, yeni algoritmalar geliştirmek, sistemleri optimize etmek ve KB uygulamalarının yaygınlaştırılmasını desteklemek için kullanılan bir araç olarak öne çıkar.



Şekil 10. EdgeCloudSim benzetim aracı arayüzü.

EdgeCloudSim, WLAN ve WAN'a özel ağ modelleme, cihaz hareket modeli ve gerçekçi, ayarlanabilir görev aktarımı gibi çeşitli kritik işlevleri desteklemek üzere modüler bir mimari sunan bir benzetim aracıdır [156]. Bu araç, çok katmanlı senaryoların benzetimini mümkün kılar; bu senaryolarda birden fazla kenar sunucusu ile üst katmanda bulunan bulut sistemleri ile koordineli bir şekilde çalışır. EdgeCloudSim, KB senaryolarında sıkça karşılaşılan düzenleyici eylemlerini modelleme olanağı tanıyan bir kenar düzenleyici modülü sunar. Hesaplama görevlerini, örneğin belirli bir kapasiteye sahip sanal makine oluşturma işlemlerini modellemek için EdgeCloudSim, hesaplama eylemlerini simüle edilebilir. EdgeCloudSim, Şekil 10'da gösterildiği gibi açık kaynak bir projedir ve modüler tasarımı ile açık kaynak kod tabanlı, kullanıcıların kendi senaryolarının özel ihtiyaçlarını benzetim aracı deneyimlerine entegre etmelerine olanak sağlar. Ayrıca bu benzetim aracı CloudSim benzetim aracı üzerine eklemeler yapılarak geliştirilmiştir.

5. NESNELERİN İNTERNETİNDE KENAR HESAPLAMA VE MAKİNE ÖĞRENMESİ KULLANILARAK YENİ BİR GÖREV TAMAMLAMA ALGORİTMASI

VEC alanında görev aktarım kararlarının verilmesi, geleneksel optimizasyon yöntemlerinin dışında NP-zor problem olarak kabul edilmektedir [52], [85]. Daha etkili sonuçlar elde etmek için makine öğrenimi yöntemlerinin kullanılması, etkili bir görev aktarım kararı alınmasına yardımcı olabilir [53], [143], [157]. Bu çalışma, üç katmanlı bir VEC mimarisi kullanılarak görev aktarım probleminin çözümünü hedeflemektedir. Bu mimari, kablosuz iletişim kurabilen araçların alt katmanda, kenar sunucularının ikinci katmanda ve bulut bilişim teknolojisinin üçüncü katmanda yer aldığı bir yapıyı içermektedir. Görev aktarımı için farklı iletişim yollarının avantajları ve seçimi, kapasite, maliyet, gecikme ve enerji tüketimi gibi kriterlere göre değerlendirilmelidir. Ayrıca, görev aktarımının etkinliğini artırmak için makine öğrenimi ve yapay zekâ tabanlı algoritmalar kullanılmaktadır.

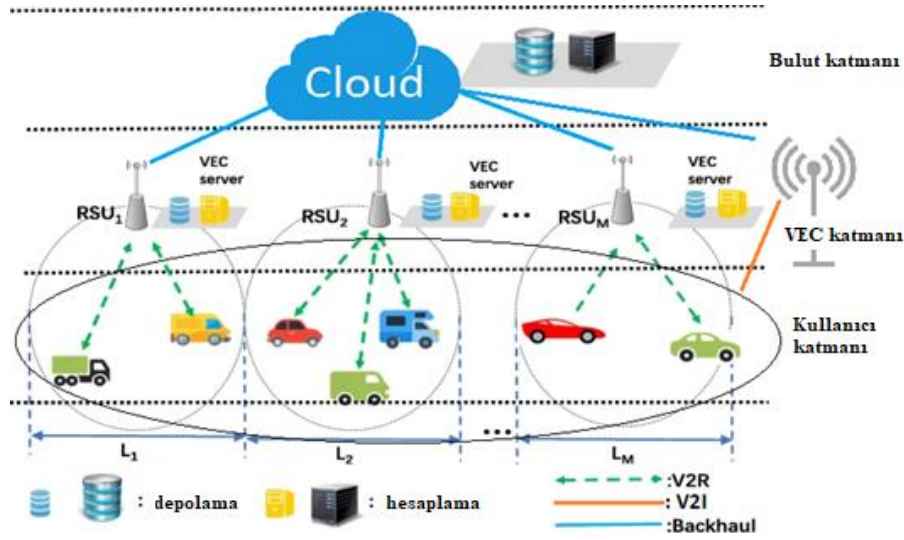
VEC'lerde kullanılan erişim teknolojileri ve altyapının topolojisi, sistem performansını önemli ölçüde etkiler [158]. Önerilen çok katmanlı VEC mimarisinde, araçlar WLAN arayüzünü kullanarak görevlerini kenar sunuculara yükleyebilir veya WAN üzerinden bulut sunucularına aktarabilirler. Bu mimari, ağ gecikmesini dikkate alarak ve farklı iletişim yollarının avantajlarını göz önünde bulundurarak görev aktarımını optimize etmeyi amaçlamaktadır.

Görev aktarımı için kullanılan karar ağaçları ve regresyon algoritması, veri setindeki görevlerin başarısını tahmin etmek için eğitildi. Görevin başarılı bir şekilde aktarılacağı en uygun hesaplama aracını seçmek için kullanılır. Deneyler, önerilen algoritmaların etkinliğini göstermektedir; dengeli bir veri seti kullanıldığında, algoritmanın doğruluk oranı %93,55 olarak elde edilmiştir. Bu sonuç, VEC'lerde görev aktarımı için makine öğrenimi tabanlı yöntemlerin kullanılmasının etkili olduğunu göstermektedir.

5.1. SİSTEM MODELİ

VEC alanında görev aktarım kararının verilmesi, geleneksel optimizasyon yöntemlerinin

dışında NP-zor problem olarak görülmektedir [67], [85]. Daha etkili sonuçlar elde etmek için ML yöntemlerinin kullanımı, etkili bir aktarım kararı alınmasına yardımcı olabilir [52]. Bu çalışmada, Şekil 11'de gösterilen 3 katmanlı bir VEC mimarisi kullanılarak görev aktarım probleminin çözümü amaçlanmaktadır.



Şekil 11. Üç katmanlı araç kenar hesaplama sistemi mimarisi.

Üç katmanlı mimariye göre, en alt katmanda kablosuz iletişim kurabilen araçlar yer alırken, ikinci katmanda RSU olarak bilinen kenar sunucusu gibi işlev gören cihazlar bulunmaktadır. Son katman ise bulut bilişim teknolojisine ayrılmıştır [159], [160]. Bu mimariye göre, bir araç tarafından üretilen veri üç farklı şekilde işlenebilir. Bu veriler, vehicle-to-infrastructure (VoI) kablosuz bağlantı yöntemiyle RSU'ya, RSU'dan edge sunucusuna, RSU aracılığıyla buluta veya GSM aracılığıyla buluta aktarılabilir.

Her aktarım yolunun kendine özgü avantajları vardır ve gönderilecek hizmetin seçimi, kapasite, maliyet, gecikme, enerji tüketimi gibi kriterlere göre yapılmalıdır. Gecikme önemli bir faktör olduğunda, ağır kenarındaki araçlarla bulut sunucuları arasında katmanlı bir artış gözlenir, çünkü bu durumda araçlara daha yakın olan aktarım yollarının gecikme süresi daha az olacaktır [16].

VEC'lerde kullanılan erişim teknolojileri ve altyapının topolojisi, sistem performansı üzerinde önemli bir etkiye sahiptir. Akıllı otoyolların, Şekil 11'de gösterildiği gibi hem kenar hem de bulut sunucuları içeren çok katmanlı çoklu erişimli VEC mimarilerindeki gelişmiş uygulamaları kullanacağını düşünülebilir. Mimarimizde araçlar, WLAN arayüzünü kullanarak görevlerini kenar sunuculara yükleyebilir. Bir görev, daha esnek ağ altyapısı sağlama olanağı sunan internet bağlantısı (WAN) aracılığıyla bulut sunucularına

da yüklenebilir. Önerilen mimaride, geniş bant bağlantısı için biri RSU üzerinden, diğeri ise hücresele ağ üzerinden olmak üzere iki alternatif bulunmaktadır. RSU'lar, [109]'da ifade edildiği gibi hücresele baz istasyonunu kullanarak bulut kaynaklarına çok sekmeli bir şekilde erişebilir. Ancak bu yaklaşım baz istasyonları üzerinde önemli bir yük oluşturmaktadır. Önerimizde, RSU'lar internet bağlantısını sağlamak için bir fiber iletişim bağlantısı kullanmaktadır. Kenar sunucuda görev aktarımı, RSU aracılığıyla bulut ve hücresele ağ (CN) aracılığıyla bulut senaryoları Şekil 11'de sırasıyla 1, 2 ve 3 durumları olarak gösterilmektedir.

RSU'lar önerilen mimaride MAN'a bağlanarak bilgi işlem kapasitelerini görev taşıma yoluyla paylaşabilirler. Bu nedenle, VEC katmanı, paylaşılan bir kaynak havuzu olarak düşünülebilir. Ayrıca; MAN, devir teslim sorununa da bir çözüm sunmaktadır. Görev aktarımı gerçekleştirilmeden önce araç hizmet veren RSU'nun menziline çıkarsa, sonuç diğere RSU'lar aracılığıyla ilgili araca çok atlamalı olarak iletilir. Devir teslim işlemi, yalnızca araç veri iletirken hizmet veren RSU'nun menziline çıktığında başarısız olur. MAN'daki ağ gecikmesi genellikle göz ardı edilse de, biz bu gecikmeyi dikkate almaktayız. Dikkate alınan bu gecikme değeri ile algoritmanın daha hassas seçim yapabilmesine olanak tanınmaktadır.

5.2. KA-KBH

Geleneksel görev aktarım yönetimleri konusunda iyileştirme yapılması hedeflendi. Bu hedef doğrultusunda, görev aktarım yönetimi için hibrid bir teknik önerilmektedir. Tez çalışmasında, görev aktarım aracının seçimine karar vermek için iki aşamalı bir makine öğrenimi tabanlı yöntem önerilmektedir. İlk aşamada, görev aktarım işleminin başarı veya başarısızlık sonucunu tahmin etmek için KA tabanlı sınıflandırma modeli kullanılmıştır. İkinci aşamada ise, başarılı olduğu tahmin edilen seçenekler içinde en uygun gecikme süresinin belirlenmesi için DR algoritması ile hizmet süresi tahmin edilmiştir.

İki aşamalı yöntem ile görevin başarı ile hizmet süresi arasındaki farklılıkları göz önünde bulundurarak tercih edilmiştir. Her iki yöntemin farklı nitelikleri olduğundan, tek bir modelle her iki ölçümü çıkarmak yerine iki basit model kullanmak daha uygun görülmüştür. Bu yaklaşım, modelin ayarlanmasını ve genelleştirilmesini kolaylaştırırken hem doğruluk oranı hem de gecikme konusunda daha etkili sonuçlar alınmasına olanak

tanımıştır.

Ayrıca, iki aşamalı yöntemin farklı girdi özelliklerini kullanabilmesi, farklı kriterlerden etkilenebilecek iki problemi ele almak için kolaylık sağlar. Bu yaklaşım, görev aktarım yönetiminde başarıyla aktarılan görevlerin sayısını artırmayı ve hizmet süresini en aza indirmeyi amaçlamaktadır. KA ve DR modelleri birçok alanda zaten kullanılmaktadır, ancak bu çalışmadaki yenilik, bu modellerin görev aktarım probleminde hibrid olarak kullanılmasıdır.

KA, temel anlamda ağaç yapısına dayanan bir makine öğrenmesi yöntemidir [161]. Buna göre belli niteliklerle elde edilecek ağaç yapısı sayesinde karmaşık nitelikteki verilerden aynı sonuca etki edecek niteliklere daha az adım ile ulaşmaktadır. Bu problemi uyguladığımız görev aktarım yapısına indirgeyecek olursak; veri setine göre, eğer kenar hesaplama araçları belli bir eşğin üzerine çıkmışsa diğer niteliklere bakılmaksızın gönderilecek görevin başarısızlıkla sonuçlanacağı kararına varılır.

Hesaplama yapılması için varsayımsal olarak n adet aktarılacak görev olduğunu ve n görevi $t = \{t_1, t_2, \dots, t_n\}$ olarak ifade edelim. Sistem modelinde ifade edildiği gibi üç ayrı seçenekle görev aktarımı yapılabilir. KA-KBH yöntemi ile görevin aktarılacağı hesaplama aracı seçiminin başarılı olup olamayacağına karar verilir. Belirlenen bu karar doğrultusunda belirli bir s zamanda görev aktarımı için servis gecikme süreleri sırasıyla $G_s^1(t_i)$, $G_s^2(t_i)$ ve $G_s^3(t_i)$ olarak ifade edilirse, t_i görevinin $G_s^j(t_i)$ yoluyla maksimum gecikme ile aktarılması B_i (başarılı olan) olarak kabul edecek olursak ve Denklem (5) te ifade edilen eşitlik sağlanması halinde görevin başarılı bir şekilde aktarıldığı anlamına gelir.

$$G_s^j(t_i) < B_i \quad (5)$$

Bir diğer önemli husus ise servis zamanının en aza indirilmesi durumudur. Servis zamanının en aza düşürülmesi için Denklem (6)'te ifade edilen eşitsizlik durumunun sağlanması gerekmektedir. Buna göre sadece son kullanıcıya en yakın hesaplama aracının gecikme süresi en az olduğu kabul edilecek olursa aşağıda ifade edilen eşitliğin sağlanmış demektir. Buna göre, bir görevin gönderildiği hesaplama aracına göre gecikmesi sırasıyla $G_s^1(t_i)$ kenar hesaplama, $G_s^2(t_i)$ bulut (WAN aracılığı ile) ve $G_s^3(t_i)$ ise bulut (GSM aracılığı ile) olarak ifade edilebilir. Denklem (6)'te ifade edilen eşitsizlik bir görevin aktarılması için en ideal gecikme süresini temsil eder ve bu sıralamaya göre seçim yapılması halinde en ideal gecikme süresinin elde edileceği anlaşılmaktadır.

$$G_s^1(t_i) < G_s^2(t_i) < G_s^3(t_i) \quad (6)$$

Öte yandan servis zamanının düşülmesi için ifade edilen eşitlik durumu bazı faktörlerden etkilenecek sıralama değişebilir. Buna göre görev boyutunun büyümesi veya birinci katmanda bulunan ve görev oluşturmada etkili olan niteliklerin artması gibi etmenler sıralamanın en altında bulunan kenar sunucuda yığılmaya ve dolayısıyla kuyrukta beklemeye neden olur. Buda dolaylı olarak gecikmelere neden olan bir durumdur. Bu durumun yaşanmaması veya sıklığının azaltılması için KA ve doğrusal regresyon (DR) algoritmalarından yararlanılmıştır.

Çalışmada görev gönderim hatalarının en aza düşürülmesi ve gecikmenin azaltılması hedeflenmiştir. Dolayısıyla başarısız görevler benzetim aracında ifade edilen senaryo için önem taşımaktadır. Bu riskin azaltılması için görevin gönderileceği hesaplama aracının ve gönderim yolunun doğru tercih edilmesi oldukça önemlidir [162].

Etkili bir görev aktarımı için ML veya yapay zekâ (AI) tabanlı çalışmalar yapılmıştır [53], [138], [139], [143], [163]. Bu çalışmalar ışığında ML tabanlı KA-KBH algoritmasını önerilmektedir. Önerilen yöntem, edgeCloudSim benzetim aracında bulunan veri seti ile görevin gönderileceği hesaplama aracının önceden başarılı olup olamayacağını tespitini sağlar. Buna göre, bir görevin önerilen yöntem ile başarılı olup olamayacağına karar verdikten sonra DR ile en etkili hesaplama aracının seçimi yapılır. Görevin başarılı olup olamayacağı ve hesaplama aracının seçimi ayrı ayrı metrikler içerdiğinden farklı algoritmalarla daha iyi sonuç alınmıştır.

Eğitim veri seti, ağ veri iletim araçlarının gecikme süreleri ve sistem modelinde ifade edilen hesaplama araçlarının kapasitelerini içermektedir. Veri seti yaklaşık olarak %5 başarısız görevlerden oluşmaktadır. Başarısız görevleri bir diğer deyişle gönderilen paketlerin kuyrukta ya da iletim hattın herhangi bir yerinde kaybolması olarak ifade edilebilir. Veri setindeki bu dengesiz dağılımı eşitlemek için ortalama değer algoritması uygulanmıştır. Her bir görev aktarım aracı için (Edge, RSU, GSM) eşit sayıda başarılı ve başarısız görevler içerir. Oluşturulan bu veri setinde bulunan tekrarlı veriler çıkarıldıktan sonra yaklaşık olarak 1.707.480 eşsiz örnekten oluşan dengeli bir veri seti elde edildi.

Yukarıda ifade edilen görev aktarım aracının seçiminde kullanılacak KA modelinin belirlenmesi için veri setinde yer alan görev verisinin uzunluğu, WLAN-MAN-WAN-GSM gecikmeleri ve kenar kullanım oranı nitelikleri kullanılmıştır.

Sözde Kod: Karar Ağaç Tabanlı Araç Kenarı Hesaplama Algoritması.

1. Input: Train dataset $S=\{(x_1,y_1), (x_2,y_2), (x_3,y_3), \dots, (x_n,y_n)\}$.
Attribute set: $A=\{a_1,a_2, \dots a_m\}$.
Attribute selection method: a procedure to determine the criteria for best best entropy defined according to the Equation (1).
Initialization: Offloading decision $d=\{s \text{ (success)}, f \text{ (fail)}\}$, Attribute set $A=\{t \text{ (taskLength)}, w_u \text{ (wanUploadDelay)}, w_d \text{ (wanDownloadDelay)}, g_u \text{ (gsmUploadDelay)}, g_d \text{ (gsmDownloadDelay)}, w_{lu} \text{ (wlanUploadDelay)}, w_{ld} \text{ (wlanDownloadDelay)}, avg \text{ (avgEdgeUtilization)}, ot \text{ (offloadedTask)}\}$
 2. for all $t \in T$ do
 3. $i \leftarrow VehicleIndex \triangleright t$ generated by i^{th} vehicle estimation task t_i
 4. if $t_i \rightarrow \text{success}$ then do
 5. $LR_i = \text{argmin}(LR_{i,edge}^e, LR_{i,rsu}^c, LR_{i,gsm}^c)$
 6. else
 7. $r = \text{random}(e, c_{rsu}, c_{gsm})$
 8. end if
 9. end for
 - 10.
-

Şekil 12. KA-KBH algoritması sözde kodu.

Şekil 12’de ifade edilen sözde koda göre eğitilmiş veri seti ile algoritma çalıştırılırken elde edilen parametrelere göre hangi yolla aracın yük dengeleme yapılacağına karar verilmektedir. Eğer seçilen yük dengeleme aracı veri setine göre başarılı ise burada yeniden bir tahmin yapılarak yük dengeleme araçları içerisinde en hızlı cevap verebilecek araç seçilmektedir. Aksi durumunda ise seçim aracı rasgele olarak seçilmektedir.

$$Y_n = X_i \beta_i + e \quad (7)$$

$$S_{min} = \{Y_n | Y_n < Y(n_0) \forall n_0 \in R\} \quad (8)$$

İfade edilen sözde kodun detaylı aşamalarını incelemek için uygulanan formüllere bakmak yardımcı olacaktır. Bu açıdan bakıldığında Denklem (3)’de elde edilen kazanım sayesinde oluşturulan KA algoritması seçimden sonra eşitlik sağlanması durumunda LR algoritması her bir hesaplama aracına uygulandığında elde edilen sonuçlardan en düşüğü seçilir. Buna göre en düşük hata oranına sahip hesaplama aracı seçimi yapılır ki bu da Denklem (8)’de ifade edildiği gibidir.

5.3. DENEYSEL SONUÇLAR

Çizelge 7’de listelenen değerler elde edilen KA modeli WEKA aracında yer alan J48

algoritması ile çalıştırılarak Çizelge 8'deki sonuçlar elde edilmiştir. Tabloda ifade edilen değerler arasında kritik öneme sahip olan T_n değeri 0,942 olarak bulunmuştur.

Çizelge 7. KA tabanlı görev aktarım kararı için model parametreleri.

Parametreler	Değerler
Maksimum derinlik	20
Güven	0.25
Ortalama mutlak hata	0.0835
Ortalama karekök hatası	0.2356
Göreceli mutlak hata	16.7096 %
Köke bağlı karekök hatası	47.1277 %
Toplam örnek sayısı	72528
Doğruluk	93.55%

Buna göre veri dağılımının dengeli hale getirilmesi aktif olan görevlerin öğrenilmesine oldukça etki etmiştir. Çagatay ve ark. [164] T_n değerini 0,878 olarak bulmuştur. Bu açıdan bakıldığında elde edilen sonuçlar daha etkili bir görev aktarım aracının seçimine ve kaynak tahsisi konusunda daha etkili sonuçlar çıkmasına olanak sağlamaktadır.

Bir diğer önemli sonuç ise dengelenmiş veri seti ile orijinal veri seti arasındaki doğruluk oranları olarak karşımıza çıkmaktadır. Yine bu iki veri setinin karşılaştırılması Çizelge 8'de detaylı olarak ifade edilmiştir.

Çizelge 8. Veri setine göre ayrıntılı doğruluk oranları.

Veri Seti	Doğruluk oranı	T_p Oranı	F_p Oranı	Hassasiyet	Geri çağırma	F-Ölçüsü
Orijinal veri seti	0,999	0,999	0,017	0,877	0,905	0,863
Dengelenmiş veri seti	0,936	0,929	0,942	0,939	0,929	0,871

6. ÖNERİLEN YÖNTEMİN DEĞERLENDİRİLMESİ

6.1. KARŞILAŞTIRILAN ALGORİTMALAR

Önerilen KA-KBH algoritmasının etkinliğini analiz etmek için, literatürde geçen algoritmalar arasında bulunan SMA, multi armed bandit (MAB) ve multi layer perceptron (MLP) algoritmaları kullanıldı. Tüm algoritmalar, aynı parametrelerle çalıştırıldı ve hatalı görev gönderme oranları gibi performans metrikleri karşılaştırıldı. Karşılaştırma için kullanılan algoritmalar literatürde en çok geçen algoritmalar arasında bulunmaktadır. Aşağıda bu algoritmaların önemli özellikleri ifade edilmiştir.

SMA: Temel olarak kısa vadeli geçmiş verilere dayanarak gelecek analizi yaparak karar veren bir algoritmadır [165]. Bu algoritma, daha yakın tarihli kullanılan verilerin ağırlıklarını artırarak, seçimin ortalama değerini kullanılmayanlardan daha doğru bir şekilde belirler.

MAB: Öğrenme sürecinde reinforcement learning algoritmasını kullanarak görevin başarılı bir şekilde gönderileceği hedefin belirlenmesini sağlayan bir algoritmadır [158]. Sistem modelinde ifade edilen görev aktarım yöntemleri arasından en kazançlı olanının seçilmesi için gecikme, görev yoğunluğu ve bekleme alınan süre gibi özelliklerden yararlanır. Algoritma birçok seçenek arasında en iyi seçeneği seçmeye çalışan ve bu süreçte keşfetme (exploration) ve en iyi seçeneği seçme (exploitation) stratejilerini dengeleyen bir optimizasyon problemi için kullanılan bir algoritmadır.

MAB problemi, bir ajanın bir dizi "kol" veya seçenek arasında dengeli bir şekilde keşif yaparak ve en iyi seçeneği bulmaya çalıştığı bir optimizasyon yöntemidir. Her kolun bir ödül dağılımı vardır, ancak başlangıçta bu dağılımlar bilinmezdir. Her bir kolun, belirli bir stratejiye göre ödüllerin gerçekleşme olasılıklarını temsil eden bir olasılık dağılımı bulunmaktadır. Başlangıçta, ajan her bir kolun gerçek ödül dağılımını bilmez. Keşfetme stratejileri, ajanın henüz denenmemiş kolları sınavarak bilgisini güncellemesine yardımcı olur. Ajan, keşfetme ile elde ettiği bilgileri kullanarak en iyi performans gösteren kolları sıkça seçmeye yönelir. Exploitation stratejileri, ajanın daha önce iyi performans göstermiş kolları seçmesine odaklanmasını sağlar. Ajan, her kolu sınavdıktan sonra elde ettiği bilgileri kullanarak ödül dağılımlarını tahmin eder ve günceller. Güncellenen bilgiler, gelecekteki seçim stratejilerini etkiler. MAB algoritmaları farklı stratejilere dayanabilir.

Örneğin, epsilon-greedy stratejisi, epsilon olasılığı ile keşif yapma ve 1-epsilon olasılığı ile en iyi kolu seçme stratejisini kullanır. MAB algoritmalarının başarı ölçütleri, genellikle toplam elde edilen ödül, optimum seçim oranı veya benzeri performans metrikleri üzerinden değerlendirilir.

Naive Bayes: Olasılığa dayalı bir sınıflandırma ve tahminleme algoritmasıdır. Bu algoritma, özellikler arasındaki bağımlılık olasılığını basitleştirerek çalışır ve bu nedenle "naive" (saf, basit) olarak adlandırılır. Temel olarak, Bayes teoremine dayanır ve özelliklerin sınıf etiketlerini tahmin etmek için olasılık hesaplamalarını kullanır.

Naive Bayes algoritması, bu olasılıkları hesaplayarak, verilen bir özellik vektörüne dayanarak veri noktasını belirli bir sınıfa sınıflandırmaya çalışır. Eğitim aşamasında, model önceki veri örneklerini kullanarak bu olasılıkları öğrenir ve test aşamasında bu öğrenilen modeli kullanarak yeni örnekleri sınıflandırır. Naive Bayes algoritması, özellikle doğal dil işleme ve spam filtreleme gibi sınıflandırma problemlerinde yaygın olarak kullanılır.

MLP: YSA olarak adlandırılan ve giriş/çıkış katmanları arasında 2-3 gizli katmanı içeren bir algoritmadır [164]. Bu model, genellikle karmaşık ve doğrusal olmayan ilişkileri öğrenmek ve modeller için karmaşık görevleri gerçekleştirmek için kullanılır.

MLP'nin temel yapı taşları, giriş katmanı, bir veya daha fazla gizli katman ve çıkış katmanından oluşur. Gizli katmanlar, sigmoid veya benzeri aktivasyon fonksiyonlarını kullanarak çok sayıda nöron içerir. Bu nöronlar, öğrenme süreci boyunca ağırlıklarını güncelleyerek giriş verileri ile çıkış arasındaki karmaşık ilişkileri öğrenir.

Geri yayılım algoritması, öğrenme sürecinde kritik bir rol oynar. Bu algoritma, öğrenme sürecinin her bir aşamasında, nöronlar arasındaki bağlantılara gradyan iniş algoritması kullanarak ağırlıkları günceller. Bu, modelin öğrenme sırasında verilere daha iyi uymasını sağlar.

MLP'nin öğrenme sürecini kontrol eden önemli parametrelerden biri öğrenme hızıdır. Bu parametre, her iterasyonda izin verilen model değişikliği miktarını belirler. Ayrıca, momentum parametresi de kullanılır; bu parametre, algoritmanın yerel minimumlarda takılı kalmaktan kaçınmasına yardımcı olur.

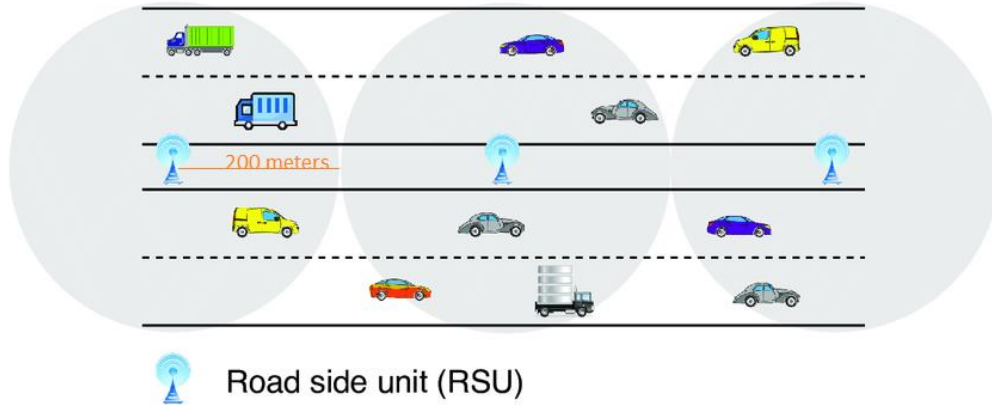
Bu açıklamaya göre, "kenar", "RSU aracılığıyla bulut" ve "hücrel aracılığıyla bulut" senaryoları için kullanılan MLP sinir ağı, modelin temel özellikleri açısından benzerdir. Model, tamamen bağlı tek bir gizli katmana sahiptir ve her bir kenar hesaplama aracı için

ayrı öğrenme yöntemleri kullanılarak özelleştirilebilir. Bu, farklı senaryolara uygun öğrenme stratejileri uygulamayı mümkün kılar.

6.2. BENZETİMİN OLUŞTURULMASI VE PARAMETRELER

Önerilen senaryoya göre, hareketli araçların bulunduğu bir rota üzerinde farklı hızlarda seyahat ederek hesaplama işlemlerinin görev aktarımı yöntemiyle gerçekleştirilmesi hedeflenmiştir. Senaryo kapsamında rota üzerinde belirlenen noktalarda yer alan RSU'lar kenar hesaplama aracı olarak kullanılıyor ve bu noktalarda görev aktarımı gerçekleştiriliyor. Senaryoda 60 KM/s, 40 KM/s ve 20 KM/s hızlarda Şekil 13'de gösterildiği gibi hareket eden araçlar yer alıyor ve düşük hızlı bölgelerde yoğunluk oluşturularak görev aktarım işleminin verimliliğinin iyileştirilmesi amaçlanıyor. Aynı zamanda yoğunluk oluşturularak gerçek hayatta yaşanacak yoğunluklar simüle edilmiştir.

Senaryo kapsamında, belirlenen rota üzerinde konumlandırılan araç sayısı 100 ila 1800 arasında değişiyor ve rota boyunca araç sayısı değiştirilmiyor. Bu senaryo üzerinde gerçekleştirilen benzetimler, belirlenen parametrelerin etkinliğini analiz etmek için kullanılmıştır.



Şekil 13. Araç hareket alanı ve RSU'ları konumlandırma.

Araçlar, belirli bir eşik değeri aştıktan sonra hesaplama aracı olarak kenar veya buluta görev aktarımı yapmaktadır. Görev aktarımı için kenar, RSU ile bulut veya GSM ile bulut seçenekleri bulunmaktadır.

Araç ile kenar arasındaki iletişim, 802.11p protokolü kullanılarak Access Point (AP) aracılığıyla sağlanmaktadır. 802.11p protokolü taşıma katmanı ihtiyacı için Kullanıcı Datagram Protokol'ü (UDP) kullanılmıştır. Bunun nedeni düşük araçlar için düşük

gecikme sağlamasıdır. WLAN olarak belirlenen bu ağ yapısı için veri transfer hızı 10 Mbps olarak belirlenmiştir. Benzetim aracında yer alan tüm RSU'lar, fiber ile ağa dahil edilecektir. Bulut hesaplama aracının kullanılabilmesi için MAN ve WAN ağ yapılarına ihtiyaç duyulmuştur. Ek olarak, bulut hesaplama aracına görev aktarımı yapabilmek için hücresel ağ yapısı seçeneği de eklenmiştir. Buna göre, 4G-LTE kullanılan bir GSM ağı için 20 Mbps'lik veri aktarım hızı öngörülmüştür. İfade edilen benzetim parametreleri Çizelge 9 ve 10'da yer almaktadır.

Çizelge 9. Benzetim parametreleri.

Parametreler	Değer
Benzetim süresi	60/3
Ağ gecikme modeli	MMPP/M/1 kuyruk modeli
Bulut/RSU başına VM sayısı	20/40
Bulut/RSU Kapasitesi	150/20 GPIS
RSU menzili	200 metre
WLAN/MAN Bant Genişliği	10/1000 Mbps
LTE Bant Genişliği	20 Mbps
LTE GSM/WAN için Yayılma Gecikmesi	150/160 ms

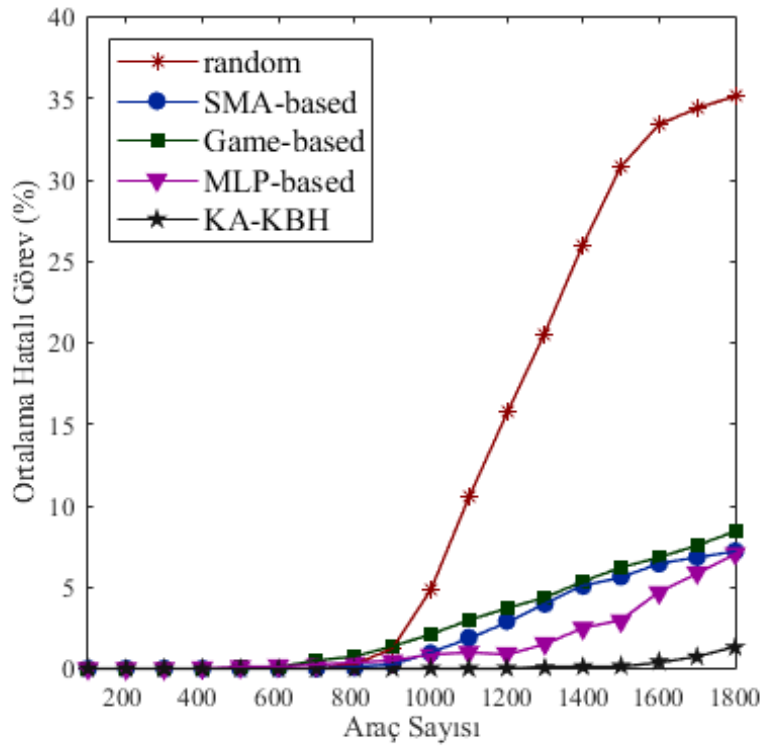
Çizelge 10. Uygulama özellikleri.

	Navigasyon Uygulamaları	Riskli Uygulamalar	Bilgilendirme Uygulamaları
Kullanım oranı (%)	30	35	35
Görev varış süreleri (saniye)	3	5	15
Max gecikme gereksinimi (s.)	0.5	1	1.5
Gecikme hassasiyeti	0.5	0.8	0.25
Veri Yükleme/indirme (KB)	20/20	40/20	20/80
Görev uzunluğu (GI)	3	10	20
RSU/bulut VM Kullanım (%)	6/1.2	20/4	40/8

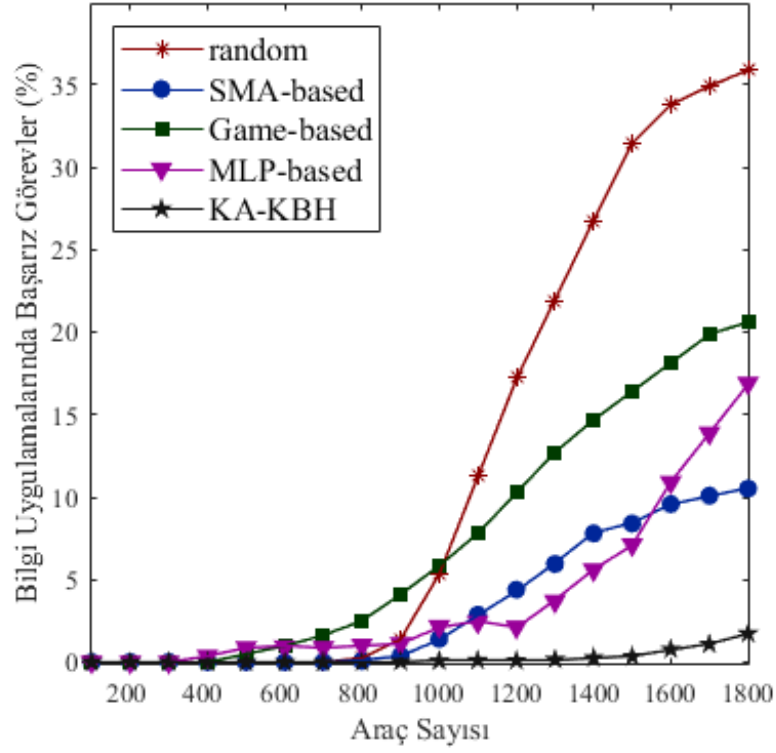
6.3. SİMÜLASYON SONUÇLARI VE TARTIŞMA

Kaynakların doğru kullanılması görevlerin başarı oranıyla doğru orantılıdır ve en etkili performans karşılaştırma kriteridir [139], [156], [164], [166]. Önemli bir başarı kriteri

olarak görülen görev başarımları oranı ağ araçlarının verimli kullanılmasına bağlıdır. edgeCloudSim benzetim aracına göre başarısız görevlerin belirlenmesinde en etkili araç, hatalı gönderilerin tespitinin doğru yapılmasına dayanmaktadır [164]. Hatalı gönderilerin sebepleri, hesaplama araçlarının kaynak oranı veya ağ kaynaklarının yetersiz olmasıdır. Bu iki önemli sebep daha detaylı ele alındığında, bir görevin başarısı kenar sunucusunun bağlı olduğu WLAN ağının bant genişliği, sunucunun hesaplama kapasitesi, kuyrukta bekleme süresi, görevin boyutu doğrudan ilişkilidir. Bu nitelikler doğru analiz edildiğinde daha etkili bir kaynak paylaşımı ve gönderilecek görevlere hangi hesaplama aracının tahsis edileceği doğru bir şekilde tespit edilebilir. Önerilen KA-KBH algoritması, Şekil 14'de gösterildiği gibi başarısız görev oranı açısından en etkili çözüm olduğu tespit edilmiştir. Özellikle 1400 araç sayısından sonra, önerilen yöntem diğer algoritmalarından oldukça avantajlı hale gelmektedir. Ayrıca görevler aktarılırken hatalı ya da bozuk nitelikli veriler dikkate alınmamıştır.

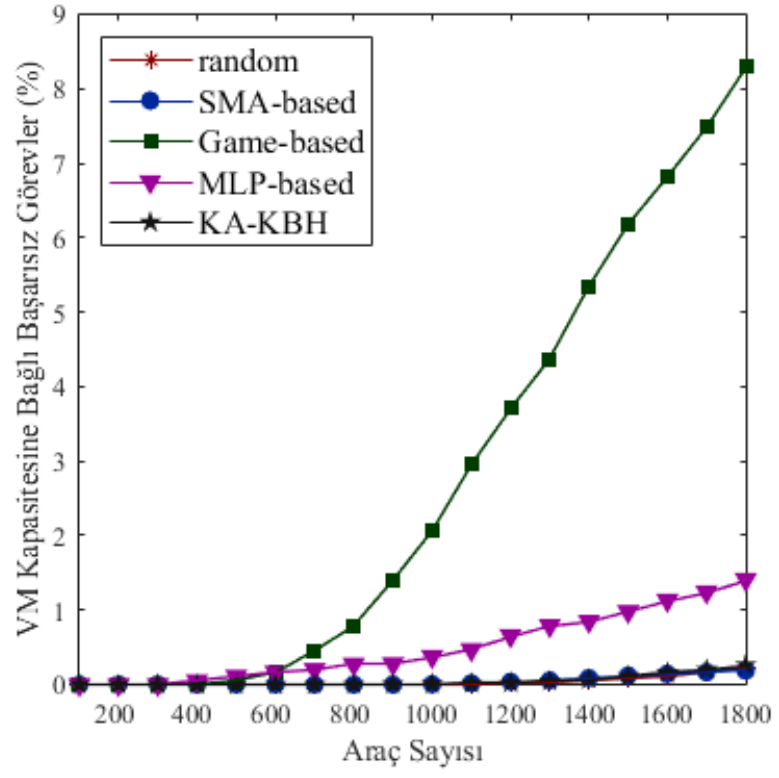


Şekil 14. Karşılaştırma algoritmalarının başarısız görev yüzdeleri.

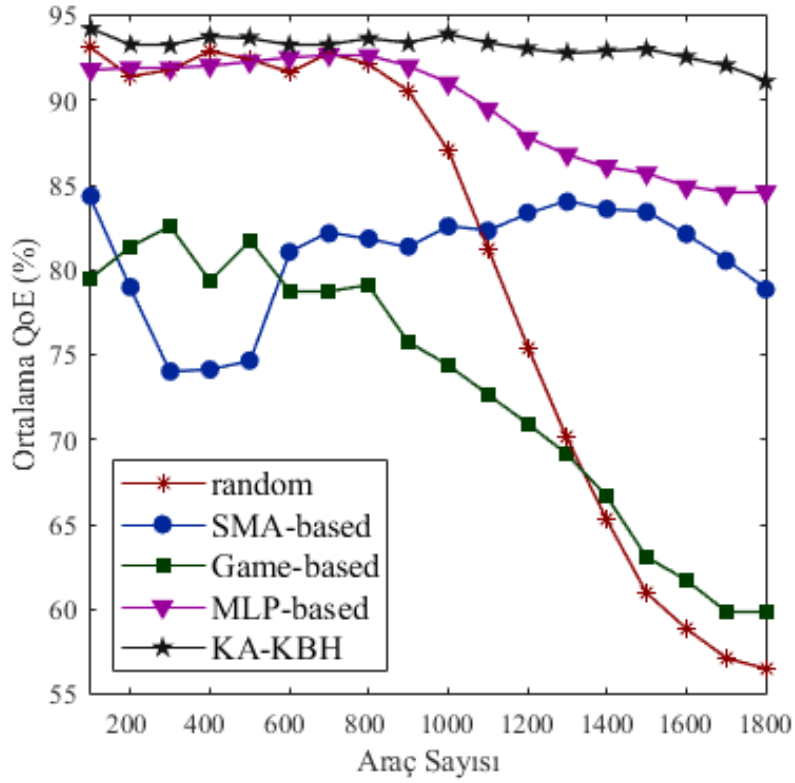


Şekil 15. Bilgi uygulaması tabanlı başarısız görevler.

Çizelge 10'de belirtilen uygulama türlerine göre başarısız görev oranları analiz edildiğinde, önerilen algoritmanın ortalama hata oranını gösteren Şekil 14 ile benzer özellikler taşıdığı görülmektedir. Ancak, Şekil 15'te bilgi uygulamaları türündeki görevler dikkate alındığında, SMA-tabanlı algoritmasının önerilen algoritmaya yakın sonuçlar verdiği görülmektedir. Bu sonuçlar, araç sayısının artması ve görev boyutuna bağlı olarak, SMA algoritmasının daha fazla bulut hesaplama kaynağı kullanabileceğini göstermektedir.



Şekil 16. VM kapasitesine göre başarısız görevler.



Şekil 17. Ortalama deneyim kalitesi (QoE).

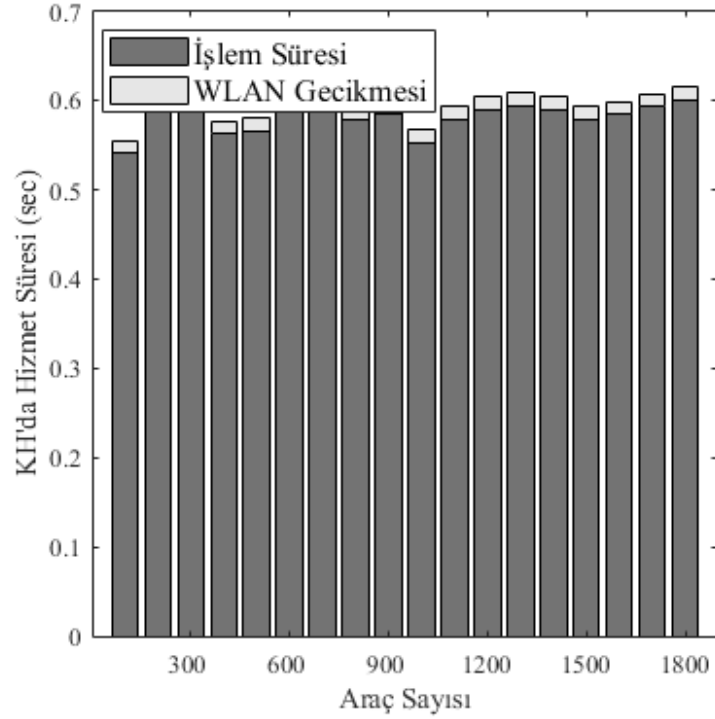
Bir diğer önemli özellik olarak, hatalı görev oranlarındaki kapasitedir. Bu kriter açısından bakıldığında, önerilen algoritmanın mevcut kapasiteyi en etkili kullanan algoritma olduğu, Şekil 16'da da gösterildiği gibi tespit edilmiştir.

Ortalama servis süresi tek başına bir başarı kriteri olarak ele alınması doğru bir yaklaşım olarak kabul edilmemektedir. Elde edilen sonuçlar, servis süresi ve hatalı görev sayısının birlikte değerlendirilmesi gerektiğini göstermektedir. Yani ortalama servis süresi düşük ancak hatalı görev oranı yüksek olan bir algoritma, hızlı sonuç alınmasına rağmen görev iletimi açısından başarılı sayılmamaktadır. Bu nedenle, sonucun doğru değerlendirilmesi için quality of experience (QoE) değerinin analiz edilmesi gerekmektedir. QoE hesaplaması Formül 9'da ifade edilen eşitliğe göre yapılmaktadır.

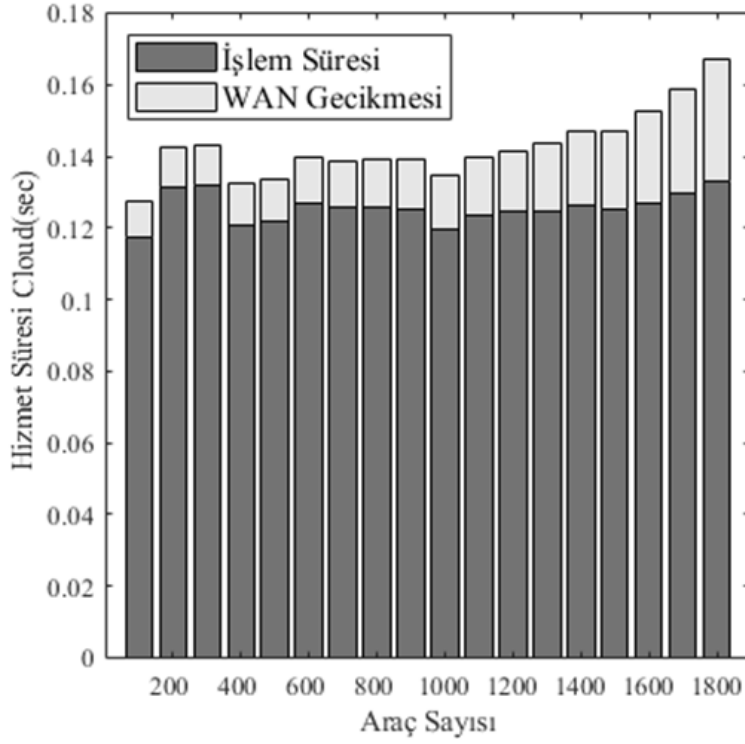
$$QoE_i = \begin{cases} 0, & \text{if } T_i \geq R_i \\ \left(1 - \frac{T_i - R_i}{R_i}\right) \cdot (1 - S_i), & \text{if } R_i < T_i < 2R_i \\ 1, & \text{if } T_i \leq R_i \end{cases} \quad (9)$$

Buna göre, i . görevin için T_i hizmet süresi, R_i gecikme gereksinimi ve S_i gecikme hassasiyetini ifade etmektedir. Hizmet süresi görev ile ilişkilendirilen maksimum servis

zamanıdır. Eğer QoE düşüyorsa bu durumda R_i görev i dan önce bitmiştir. İfade edilen bu hesaplamalar çerçevesinde elde edilen QoE sonuçları yüzde (%) olarak Şekil 17’de gösterilmiştir. Bu sonuçlar incelendiğinde 100-1800 arasındaki araç sayısı dikkate alındığında KA daki azalma oranı %8.62, SMA’DA %10.51 olarak elde edilmiştir. Her ne kadar algoritmanın çalışma zamanı QoE’ya etki etse de başarılı görev oranı açısından önerilen çalışma daha iyi sonuç verdiği görülmektedir.



Şekil 18. Kenar sunucuda hizmet süresi.



Şekil 19. Bulut sunucuda hizmet süresi.

Analiz edilen servis zamanı sonuçlarına göre, Edge ve Cloud arasındaki gecikmenin farklı nedenlere dayandığı görülmüştür. Bu farkın, hesaplama kapasitelerine bağlı olduğu anlaşılmaktadır. Edge sunucusunun kapasitesinin neredeyse tamamının kullanıldığı ve bu durumun hem gecikme hem de başarılı görev oranı üzerinde etkili olduğu Şekil 18 ve Şekil 19'da görülmektedir. Öte yandan, Cloud hesaplama süresi açısından oldukça etkilidir ancak WAN gecikmesinin önemli bir etkisi olduğu gözlemlenmektedir. Yani Cloud, hesaplama açısından etkili bir sonuç ortaya koyarken Edge, WLAN gecikmesi veya ağ iletim gecikmesi açısından daha etkilidir ve bu da mesafeye bağlıdır.

Tüm sonuçlar birlikte incelendiğinde, önerilen yaklaşımın hatalı görev oranı, QoE niteliği ve mevcut kaynakların etkili kullanımını açısından daha iyi sonuçlar verdiği görülmektedir.

7. SONUÇ VE ÖNERİLER

VEC sistemlerinde, dinamik ve heterojen bir ağ organizasyonu, sistemlerin karmaşıklığını artıran ve verimliliği etkileyen önemli bir sorun alanını temsil eder. Bu bağlamda, aktarılacak görevlerin doğru hesaplama aracına iletilmesi, etkili bir ağ organizasyonu ile mümkündür. Dinamik yapılarda ise ihtiyaç duyulan değişikliklerin hızla gelişmesi, bu alanda sürekli olarak iyileştirmeler ve adaptasyonlar yapma ihtiyacını beraberinde getirmektedir.

VEC sistemlerindeki dinamizm, genellikle araçların hızlı bir şekilde değişen koşullara adapte olmalarını gerektirir. Bu koşullar, ağ yapısı, trafik yoğunluğu, kullanıcı talepleri, araç sayısı ve çeşitli hava koşulları gibi bir dizi faktörü içerir. Dinamik bir ağ organizasyonu, bu faktörleri sürekli olarak izler, değerlendirir ve en uygun ağ konfigürasyonunu sağlamak için gerekli ayarlamaları yapar.

Ayrıca, dinamik yapılarda hızlı gelişen değişimler, VEC sistemlerinin performansını ve güvenilirliğini etkileyebilir. Bu nedenle, ağ organizasyonu sürekli olarak bu değişimlere uyum sağlamalı, aktarılacak görevleri doğru hesaplama aracına yönlendirmeli ve bu süreçte en etkili iletişim yollarını belirlemelidir. Bu adaptasyon yeteneği, VEC sistemlerinin hızla evrilen çevresel koşullara ve kullanıcı ihtiyaçlarına etkin bir şekilde yanıt vermesini sağlar.

Bu bağlamda, VEC mimarisinde araçlar arasındaki iletişimi daha etkili hale getirmeyi amaçlayan üç katmanlı bir yaklaşım benimsenmiştir. Bu yaklaşım, araçların birbirleriyle ve çevresindeki altyapı ile daha etkili bir şekilde iletişim kurabilmesi için tasarlanmıştır. Bu hedef doğrultusunda, VEC için üç önemli iletişim seçeneği tanımlanarak, bu seçenekler arasında en etkili olanın belirlenmesi sağlanmıştır.

İletişim seçeneklerinin belirlenmesi için, KA ve DR algoritmalarının birleştirilerek hibrid bir yaklaşım kullanılmıştır. Bu hibrid yaklaşım, her bir iletişim seçeneğinin belirli durumlarda ne kadar etkili olduğunu daha doğru bir şekilde değerlendirmeyi amaçlamaktadır. Bu seçim, daha etkili bir görev aktarım stratejisinin hayata geçirilmesine olanak tanımıştır.

Önerilen yöntemin etkinliği, edgeCloudSim simülatörü üzerinde gerçekleştirilen çeşitli senaryo testleri ve araç sayılarına yönelik benzetimlerle değerlendirilmiştir. Elde edilen

sonular, nerilen yntemin yakın gelecekteki VEC sistemleri iin avantajlar saėlama potansiyeline iřaret etmektedir.

Geliřtirilen yntemin analiz ve deėerlendirmesi iin Weka aracı kullanılmıřtır. Bu ok ynl ara, elde edilen verilerin doėru bir řekilde analiz edilmesini ve sonuların gvenilir bir řekilde yorumlanmasını saėlamıřtır.

Sonular, makine ėrenmesi yaklařımlarının bir araya getirilmesinin, etkili bir grev aktarım seimi iin daha iyi sonular alınmasına nemli katkılar saėladığını gstermektedir. Ayrıca, paket kayıp oranının kritik olduėu sistemlerde, nerilen yaklařımın olduka etkili bir zm sunabileceėi vurgulanmıřtır. Bu, VEC sistemlerinin performansını ve gvenilirliėini artırmak adına nemli bir adım olarak deėerlendirilebilir.

alıřmanın bir sonraki ařamasında, nerilen yaklařıma mimari erveden yenilikler eklenerek, daha yoėun aėlarda daha etkili zmler elde edilmesi hedeflenmektedir. Ayrıca sis biliřim teknolojisinden de yararlanılarak yeni mimari erve ile daha etkili sonular alınmasını hedeflenmektedir. Aynı zamanda yeni nesil aė teknoloji olan 5g'nin de sisteme dahil edilerek gecikme duyarlı sistemlerde daha etkili sonular alınması dřnlmektedir.



8. KAYNAKLAR

- [1] H. J. A. Van Os *et al.*, “The Internet of Things for Health Care: A Comprehensive Survey,” *International Journal of Stroke*, vol. 10. p. 217, 2015. [Online]. Available: <http://www.embase.com/search/results?subaction=viewrecord&from=export&id=L72034289%5Cnhttp://dx.doi.org/10.1111/ijs.12479>
- [2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1. pp. 22–32, 2014. doi: 10.1109/JIOT.2014.2306328.
- [3] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15. pp. 2787–2805, 2010. doi: 10.1016/j.comnet.2010.05.010.
- [4] J. A. Stankovic, “Research Directions for the Internet of Thing,” *Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.
- [5] H. Huang, J. Zhu, and L. Zhang, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *IET Conference Publications*, vol. 2014, no. CP639. pp. 175–179, 2014. doi: 10.1049/cp.2014.0680.
- [6] “IoT devices in use worldwide 2009-2020 | Statista.” Accessed: Sep. 30, 2020. [Online]. Available: <https://www.statista.com/statistics/764026/number-of-iot-devices-in-use-worldwide/>
- [7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
- [8] D. Mishra, A. Gunasekaran, S. J. Childe, T. Papadopoulos, R. Dubey, and S. Wamba, “Vision, applications and future challenges of Internet of Things: A bibliometric study of the recent literature,” *Industrial Management and Data Systems*, vol. 116, no. 7. Emerald Group Publishing Ltd., pp. 1331–1355, Aug. 08, 2016. doi: 10.1108/IMDS-11-2015-0478.
- [9] A. Gupta and R. K. Jha, “A Survey of 5G Network: Architecture and Emerging Technologies,” *IEEE Access*, vol. 3, no. Yok, pp. 1206–1232, 2015, doi: 10.1109/ACCESS.2015.2461602.
- [10] J. G. Andrews *et al.*, “What will 5G be?,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6. pp. 1065–1082, 2014. doi: 10.1109/JSAC.2014.2328098.
- [11] A. P. Alessio Botta, Walter de Donato, Valerio Persico, “Integration of Cloud

- Computing and Internet of Things: a Survey Alessio,” *Future Generation Computer Systems*, vol. 56, no. 2016, pp. 684–700, 2015.
- [12] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5. pp. 637–646, 2016. doi: 10.1109/JIOT.2016.2579198.
- [13] K. Georgiou, S. Xavier-De-Souza, and K. Eder, “The IoT Energy Challenge: A Software Perspective,” *IEEE Embedded Systems Letters*, vol. 10, no. 3. Institute of Electrical and Electronics Engineers Inc., pp. 53–56, Sep. 01, 2018. doi: 10.1109/LES.2017.2741419.
- [14] F. Jalali, O. J. Smith, T. Lynar, and F. Suits, “Cognitive IoT gateways: Automatic task sharing and switching between cloud and Edge/Fog computing,” in *SIGCOMM Posters and Demos 2017 - Proceedings of the 2017 SIGCOMM Posters and Demos, Part of SIGCOMM 2017*, Association for Computing Machinery, Inc, Aug. 2017, pp. 121–123. doi: 10.1145/3123878.3132008.
- [15] M. Aazam and E. N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT,” *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, vol. 2015-April. pp. 687–694, 2015. doi: 10.1109/AINA.2015.254.
- [16] L. Tong, Y. Li, and W. Gao, “A hierarchical edge cloud architecture for mobile computing,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, IEEE, Apr. 2016, pp. 1–9. doi: 10.1109/INFOCOM.2016.7524340.
- [17] S. M. Nirav and A. Buchade, “Priority Based Resource Allocation in Cloud Computing,” in *2012 1st IEEE International Conference on Cloud Networking, CLOUDNET 2012 - Proceedings*, 2014, pp. 63–67. doi: 10.1109/CloudNet.2012.6483656.
- [18] P. Pierleoni, R. Concetti, A. Belli, and L. Palma, “Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison,” *IEEE Access*, vol. 8, pp. 5455–5470, 2020, doi: 10.1109/ACCESS.2019.2961511.
- [19] Y. Zhang, Y. Zhang, X. Lan, J. Ren, J. Ren, and L. Cai, “Efficient computing resource sharing for mobile edge-cloud computing networks,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1227–1240, 2020, doi: 10.1109/TNET.2020.2979807.
- [20] H. El-Sayed *et al.*, “Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment,” *IEEE Access*, vol. 6, pp. 1706–1717, 2017, doi: 10.1109/ACCESS.2017.2780087.
- [21] M. De Donno, K. Tange, and N. Dragoni, “Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog,” *IEEE Access*, vol. 7, pp. 150936–150948, 2019, doi: 10.1109/ACCESS.2019.2947652.

- [22] S. Deng, H. Zhao, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *ArXiv*, vol. 7, no. 8, pp. 7457–7469, 2019.
- [23] X. Shen, "A hierarchical task scheduling strategy in mobile edge computing," *Internet Technology Letters*, vol. 4, no. 5, pp. 1–6, Sep. 2021, doi: 10.1002/itl2.224.
- [24] J. M. Khurpade, "A SURVEY ON IOT AND 5G NETWORK," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, pp. 1–3, 2018.
- [25] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 567–619, Jul. 2015, doi: 10.1007/s10922-014-9307-7.
- [26] B. Hayes, "Cloud Computing," *Commun ACM*, vol. 51, no. 7, pp. 9–11, May 2008, doi: 10.1145/1364782.1364786.
- [27] S. Li, L. Da Xu, and S. Zhao, "Journal of Industrial Information Integration 5G Internet of Things: A survey," vol. 10, no. January, pp. 1–9, 2018, doi: 10.1016/j.jii.2018.01.005.
- [28] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4. IEEE Computer Society, pp. 2233–2243, Nov. 01, 2014. doi: 10.1109/TII.2014.2300753.
- [29] S. O’Dea, "IoT devices in use worldwide 2009-2020," Number of IoT devices in use worldwide from 2009 to 2020. [Online]. Available: <https://www.statista.com/statistics/764026/number-of-iot-devices-in-use-worldwide/>
- [30] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2. pp. 243–259, 2015. doi: 10.1007/s10796-014-9492-7.
- [31] W. Xu *et al.*, "Internet of vehicles in big data era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018, doi: 10.1109/JAS.2017.7510736.
- [32] Arne Holst, "Connected IoTs." Accessed: Oct. 13, 2021. [Online]. Available: <https://www.statista.com/statistics/1194677/iot-connected-devices-regionally/>
- [33] P. Guillemin, H. Sundmaeker, M. Eisenhauer, and K. Moessner, "Internet of Things Strategic Research and Innovation Agenda," no. January, River Publishers, 2013.
- [34] Y. P. Yulei Wu, Haojun Huang, Cheng-Xiang Wang, *5G-Enabled Internet of Things*, vol. 66. 2012.
- [35] A. Guillen-Perez and M. Cano, "Intelligent IoT systems for traffic management: A practical application," *IET Intelligent Transport Systems*, vol. 15, no. 2, pp. 273–

285, Feb. 2021, doi: 10.1049/itr2.12021.

- [36] M. TAY and A. ŞENTÜRK, “Kenar, Sis ve Bulut Bilişimin IoT Açısından İncelenmesi,” *European Journal of Science and Technology*, Jan. 2022, doi: 10.31590/ejosat.1040133.
- [37] M. Tay and A. Senturk, “A research on resource allocation algorithms in content of edge, fog and cloud,” *Mater Today Proc*, Dec. 2022, doi: 10.1016/j.matpr.2022.11.232.
- [38] R. L. Grossman, “Cloud Computing,” 2009. [Online]. Available: <http://hadoop.apache.org/core>
- [39] E. Knorr and G. Gruman, “What Cloud Computing Really Means.” [Online]. Available: www.infoworld.com
- [40] A. Chinthas, D. Rani, and R. K. Ranjan, “A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing,” 2014. [Online]. Available: www.ijarcsse.com
- [41] S. S. Manvi and G. Krishna Shyam, “Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey,” *Journal of Network and Computer Applications*, vol. 41, pp. 424–440, May 2014, doi: 10.1016/j.jnca.2013.10.004.
- [42] D. Beimborn, T. Miletzki, and S. Wenzel, “Platform as a Service (PaaS),” *WIRTSCHAFTSINFORMATIK*, vol. 53, no. 6, pp. 371–375, Dec. 2011, doi: 10.1007/s11576-011-0294-y.
- [43] H. Shah-Mansouri and V. W. S. Wong, “Hierarchical fog-cloud computing for IoT systems: A computation offloading game,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018. doi: 10.1109/JIOT.2018.2838022.
- [44] W. Tsai, X. Bai, and Y. Huang, “Software-as-a-service (SaaS): perspectives and challenges,” *Science China Information Sciences*, vol. 57, no. 5, pp. 1–15, May 2014, doi: 10.1007/s11432-013-5050-z.
- [45] E. Bisong, “An Overview of Google Cloud Platform Services,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berkeley, CA: Apress, 2019, pp. 7–10. doi: 10.1007/978-1-4842-4470-8_2.
- [46] S. Goyal, “Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review,” *International Journal of Computer Network and Information Security*, vol. 6, no. 3, pp. 20–29, Feb. 2014, doi: 10.5815/ijcnis.2014.03.03.
- [47] A. Li, X. Yang, S. Kandula, and M. Zhang, “CloudCmp,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA: ACM, Nov. 2010, pp. 1–14. doi: 10.1145/1879141.1879143.
- [48] G. Aryotejo, D. Y. Kristiyanto, and Mufadhol, “Hybrid cloud: bridging of private and public cloud computing,” *J Phys Conf Ser*, vol. 1025, p. 012091, May 2018, doi: 10.1088/1742-6596/1025/1/012091.

- [49] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, “Amazon S3 for science grids,” in *Proceedings of the 2008 international workshop on Data-aware distributed computing*, New York, NY, USA: ACM, Jun. 2008, pp. 55–64. doi: 10.1145/1383519.1383526.
- [50] H. Hacigumus, B. Iyer, and S. Mehrotra, “Providing database as a service,” in *Proceedings 18th International Conference on Data Engineering*, IEEE Comput. Soc, 2002, pp. 29–38. doi: 10.1109/ICDE.2002.994695.
- [51] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, “The rise of ‘big data’ on cloud computing: Review and open research issues,” *Inf Syst*, vol. 47, pp. 98–115, Jan. 2015, doi: 10.1016/j.is.2014.07.006.
- [52] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, “Computation offloading in Edge Computing environments using Artificial Intelligence techniques,” *Eng Appl Artif Intell*, vol. 95, p. 103840, Oct. 2020, doi: 10.1016/j.engappai.2020.103840.
- [53] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, “A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective,” *Computer Networks*, vol. 182, p. 107496, Dec. 2020, doi: 10.1016/j.comnet.2020.107496.
- [54] J. Wang, Z. Feng, S. George, R. Iyengar, P. Pillai, and M. Satyanarayanan, “Towards scalable edge-native applications,” in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, New York, NY, USA: ACM, Nov. 2019, pp. 152–165. doi: 10.1145/3318216.3363308.
- [55] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. YANG, and W. Wang, “A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications,” *IEEE Access*, vol. 5, pp. 6757–6779, 2017, doi: 10.1109/ACCESS.2017.2685434.
- [56] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, “Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review,” *Journal of Systems Architecture*, vol. 122. Elsevier B.V., Jan. 01, 2022. doi: 10.1016/j.sysarc.2021.102362.
- [57] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, “Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review,” *Journal of Systems Architecture*, vol. 122. Elsevier B.V., Jan. 01, 2022. doi: 10.1016/j.sysarc.2021.102362.
- [58] “Moving the Cloud to the Edge with Edge Computing | PubNub.” Accessed: Apr. 05, 2022. [Online]. Available: <https://www.pubnub.com/blog/moving-the-cloud-to-the-edge-computing/>
- [59] H. Xiao, Q. Pei, X. Song, and W. Shi, “Authentication Security Level and Resource Optimization of Computation Offloading in Edge Computing Systems,” *IEEE Internet Things J*, vol. 9, no. 15, pp. 13010–13023, Aug. 2022, doi:

10.1109/JIOT.2021.3139222.

- [60] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-Efficient Edge AI: Algorithms and Systems," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2167–2191, Oct. 2020, doi: 10.1109/COMST.2020.3007787.
- [61] M. Marjanovic, A. Antonic, and I. P. Zarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10662–10674, 2018, doi: 10.1109/ACCESS.2018.2799707.
- [62] Z. Qu, Y. Wang, L. Sun, D. Peng, and Z. Li, "Study QoS optimization and energy saving techniques in cloud, Fog, EDge, and IoT," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/8964165.
- [63] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification," *Sensors*, vol. 21, no. 5. MDPI AG, pp. 1–23, Mar. 01, 2021. doi: 10.3390/s21051832.
- [64] M. Asim, Y. Wang, K. Wang, and P. Q. Huang, "A Review on Computational Intelligence Techniques in Cloud and Edge Computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 6. Institute of Electrical and Electronics Engineers Inc., pp. 742–763, Dec. 01, 2020. doi: 10.1109/TETCI.2020.3007905.
- [65] Q. Qi and F. Tao, "A Smart Manufacturing Service System Based on Edge Computing , Fog Computing , and Cloud Computing," *IEEE Access*, vol. 7, pp. 86769–86777, 2019, doi: 10.1109/ACCESS.2019.2923610.
- [66] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification," *Sensors*, vol. 21, no. 5. MDPI AG, pp. 1–23, Mar. 01, 2021. doi: 10.3390/s21051832.
- [67] J. Tang, T. Qin, Y. Xiang, Z. Zhou, and J. Gu, "Optimization Search Strategy for Task Offloading from Collaborative Edge Computing," *IEEE Trans Serv Comput*, pp. 1–14, 2022, doi: 10.1109/TSC.2022.3203700.
- [68] X. Fan, H. Zheng, R. Jiang, and J. Zhang, "Optimal design of hierarchical cloud-fog&edge computing networks with caching," *Sensors (Switzerland)*, vol. 20, no. 6, pp. 1–16, Mar. 2020, doi: 10.3390/s20061582.
- [69] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3832–3840, Jun. 2021, doi: 10.1109/TITS.2020.3048844.
- [70] X. Li and L. Da Xu, "A Review of Internet of Things - Resource Allocation," *IEEE Internet of Things Journal*, vol. 8, no. 11. Institute of Electrical and Electronics Engineers Inc., pp. 8657–8666, Jun. 01, 2021. doi: 10.1109/JIOT.2020.3035542.

- [71] J. Ren, G. Yu, Y. He, and G. Y. Li, “Collaborative Cloud and Edge Computing for Latency Minimization,” *IEEE Trans Veh Technol*, vol. 68, no. 5, pp. 5031–5044, 2019, doi: 10.1109/TVT.2019.2904244.
- [72] J. Zhang and K. B. Letaief, “Mobile Edge Intelligence and Computing for the Internet of Vehicles,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020, doi: 10.1109/JPROC.2019.2947490.
- [73] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, “BEGIN: Big Data Enabled Energy-Efficient Vehicular Edge Computing,” *IEEE Communications Magazine*, vol. 56, no. 12, pp. 82–89, Dec. 2018, doi: 10.1109/MCOM.2018.1700910.
- [74] C. Tang, C. Zhu, H. Wu, Q. Li, and J. J. P. C. Rodrigues, “Toward Response Time Minimization Considering Energy Consumption in Caching-Assisted Vehicular Edge Computing,” *IEEE Internet Things J*, vol. 9, no. 7, pp. 5051–5064, Apr. 2022, doi: 10.1109/JIOT.2021.3108902.
- [75] M. A. Javed and S. Zeadally, “AI-Empowered Content Caching in Vehicular Edge Computing: Opportunities and Challenges,” *IEEE Netw*, vol. 35, no. 3, pp. 109–115, May 2021, doi: 10.1109/MNET.011.2000561.
- [76] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, “Collaborative Vehicular Edge Computing Networks: Architecture Design and Research Challenges,” *IEEE Access*, vol. 7, pp. 178942–178952, 2019, doi: 10.1109/ACCESS.2019.2957749.
- [77] G. Mohindru, K. Mondal, and H. Banka, *Internet of Things and data analytics: A current review*, Second., vol. 10. New Jersey: Willey, 2020.
- [78] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016. doi: 10.1109/ACCESS.2016.2566339.
- [79] Y. C. and S. C. Y. Liu, M. Peng, G. Shou, “Toward Edge Intelligence: Multiaccess Edge Computing for 5G and Internet of Things,” *Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.
- [80] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, “Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks,” *IEEE Access*, vol. 7, pp. 26652–26664, 2019, doi: 10.1109/ACCESS.2019.2900530.
- [81] W. Yu *et al.*, “A Survey on the Edge Computing for the Internet of Things,” *IEEE Access*, vol. 6. Institute of Electrical and Electronics Engineers Inc., pp. 6900–6919, Nov. 28, 2017. doi: 10.1109/ACCESS.2017.2778504.
- [82] M. A. Hoque and R. Hasan, “Towards an Analysis of the Architecture, Security, and Privacy Issues in Vehicular Fog Computing,” in *2019 SoutheastCon*, IEEE, Apr. 2019, pp. 1–8. doi: 10.1109/SoutheastCon42311.2019.9020476.

- [83] X. Deng, Z. Sun, D. Li, J. Luo, and S. Wan, "User-Centric Computation Offloading for Edge Computing," *IEEE Internet Things J*, vol. 8, no. 16, pp. 12559–12568, Aug. 2021, doi: 10.1109/JIOT.2021.3057694.
- [84] G. Orsini, D. Bade, and W. Lamersdorf, "CloudAware: Empowering context-aware self-adaptation for mobile applications," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, Apr. 2018, doi: 10.1002/ett.3210.
- [85] J. W. Xiangwang Hou , Zhiyuan Ren, "Reliable Computation Offloading for Edge," vol. 7, no. March, pp. 7097–7111, 2020.
- [86] W. Shi, J. Zhang, and R. Zhang, "Share-Based Edge Computing Paradigm with Mobile-to-Wired Offloading Computing," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1953–1957, 2019, doi: 10.1109/LCOMM.2019.2934411.
- [87] Z. Wang and X. Su, "Dynamically hierarchical resource-allocation algorithm in cloud computing environment," *Journal of Supercomputing*, vol. 71, no. 7, pp. 2748–2766, Jul. 2015, doi: 10.1007/s11227-015-1416-x.
- [88] A. A. R. Mostafa Ghobaei-Arani, Alireza Souri, "Resource Management Approaches in Fog Computing: a Comprehensive Review Mostafa," *Grid Computing*, 2019.
- [89] D. N. and C. S. H. S. F. Abedin, M. G. R. Alam, S. M. A. Kazmi, N. H. Tran, "Resource Allocation for Ultra-reliable and Enhanced Mobile Broadband IoT Applications in Fog Network," *Transactions on Communications*, vol. 67, no. 1, pp. 489–502, 2019.
- [90] Q. W. and K. S. W. Wang, "Multimedia Sensing as a Service (MSaaS): Exploring Resource Saving Potentials of at Cloud-Edge IoTs and Fogs," *Internet of Things Journal*, vol. 4, no. 7, pp. 487–495, 2016.
- [91] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge Mesh: A New Paradigm to Enable Distributed Intelligence in Internet of Things," *IEEE Access*, vol. 5, pp. 16441–16458, 2017. doi: 10.1109/ACCESS.2017.2739804.
- [92] X. Xu *et al.*, "Dynamic Resource Allocation for Load Balancing in Fog Environment," *Wireless Communications and Mobile Computing*, vol. 2018. 2018. doi: 10.1155/2018/6421607.
- [93] A. A. Alsaffar, H. P. Pham, C. S. Hong, E. N. Huh, and M. Aazam, "An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing," *Mobile Information Systems*, vol. 2016. 2016. doi: 10.1155/2016/6123234.
- [94] R. bUYYA S. S. Gill, P. Garraghan, "ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices," *J Syst Softw*, vol. 154, pp. 125–138 Contents, 2019.

- [95] R. F. and B. P. F. Chiti, "A Matching Theory Framework for Tasks Offloading in Fog Computing for IoT Systems," *Internet of Things Journal*, vol. 5, no. 6, pp. 5089–5096, 2018.
- [96] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: Opportunities for 5G networks," *Applied Sciences (Switzerland)*, vol. 10, no. 14. 2020. doi: 10.3390/app10144735.
- [97] L. H. and I. H. M. Chen, Y. Miao, H. Gharavi, "Intelligent Traffic Adaptive Resource Allocation for Edge Computing-based 5G Networks," *Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 499–508, 2020.
- [98] M. M. Hesham El-Sayed, Sharmi Sankar, Mukesh Prasad, Deepak Puthal, Akshansh Gupta, "Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.
- [99] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M. T. Zhou, "MEETS: Maximal Energy Efficient Task Scheduling in Homogeneous Fog Networks," *IEEE Internet of Things Journal*, vol. 5, no. 5. pp. 4076–4087, 2018. doi: 10.1109/JIOT.2018.2846644.
- [100] L. S. and Z. H. Y. Gu, Z. Chang, M. Pan, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7475–7484, 2018.
- [101] O. M. Tiscornia *et al.*, "LA Inervación Autónoma Gastro-Entero-Bilio-Pancreática: El concepto de 'pista' plexual entérica," *Prensa Medica Argentina*, vol. 86, no. 2. pp. 129–139, 1999.
- [102] and Z. H. Nguyen Cong Luong, Yutao Jiao, Ping Wang, Dusit Niyato, Dong In Kim and To, "A Machine-Learning-Based Auction for Resource Trading in Fog Computing," *Communications Magazine*, vol. 58, no. 3, pp. 82–88, 2020.
- [103] N. Mostafa, "Resource Selection Service Based on Neural Network in Fog Environment," *Technology and Engineering Systems Journal*, pp. 408–417, 2020.
- [104] J. Du, C. Jiang, A. Benslimane, S. Guo, and Y. Ren, "Stackelberg Differential Game Based Resource Sharing in Hierarchical Fog-Cloud Computing," 2019 IEEE Global Communications Conference (GLOBECOM), 2019.
- [105] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Distributed Deep Neural Networks with System Cost Minimization in Fog Networks," Aug. 2017, [Online]. Available: <http://arxiv.org/abs/1708.03074>
- [106] W. R. , X. M.-B. , E. M.-T. , G. R. , G. T. V.B.C.Souza, *Handling Service Allocation in Combined Fog-Cloud Scenarios*. 2016.
- [107] M. M. Shurman, S. H. Alnabelsi, M. Aljarah, and M. Shurman, "Cooperative-Hierarchical Based Edge-Computing Approach for Resources Allocation of

- Distributed Mobile and IoT Applications,” *Article in International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, 2020, doi: 10.11591/ijece.v10i1.pp%25p.
- [108] P. Kochovski, R. Sakellariou, M. Bajec, P. Drobintsev, and V. Stankovski, “An architecture and stochastic method for database container placement in the edge-fog-cloud continuum,” in *Proceedings - 2019 IEEE 33rd International Parallel and Distributed Processing Symposium, IPDPS 2019*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 396–405. doi: 10.1109/IPDPS.2019.00050.
- [109] J. X. Liao and X. W. Wu, “Resource Allocation and Task Scheduling Scheme in Priority-Based Hierarchical Edge Computing System,” in *Proceedings - 2020 19th Distributed Computing and Applications for Business Engineering and Science, DCABES 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 46–49. doi: 10.1109/DCABES50732.2020.00021.
- [110] H. Cao and M. Wachowicz, “An Edge-Fog-Cloud Architecture of Streaming Analytics for Internet of Things Applications,” *Sensors*, vol. 19, no. 16, p. 3594, Aug. 2019, doi: 10.3390/s19163594.
- [111] Z. M. Nayeri, T. Ghafarian, and B. Javadi, “Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey,” *Journal of Network and Computer Applications*, vol. 185. Academic Press, Jul. 01, 2021. doi: 10.1016/j.jnca.2021.103078.
- [112] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, “Multitier Fog Computing With Large-Scale IoT Data Analytics for Smart Cities,” *IEEE Internet Things J*, vol. 5, no. 2, pp. 677–686, Apr. 2018, doi: 10.1109/JIOT.2017.2724845.
- [113] H. Bashir, S. Lee, and K. H. Kim, “Resource allocation through logistic regression and multicriteria decision making method in IoT fog computing,” *Transactions on Emerging Telecommunications Technologies*, 2019, doi: 10.1002/ett.3824.
- [114] M. Asim, Y. Wang, K. Wang, and P. Q. Huang, “A Review on Computational Intelligence Techniques in Cloud and Edge Computing,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 6. Institute of Electrical and Electronics Engineers Inc., pp. 742–763, Dec. 01, 2020. doi: 10.1109/TETCI.2020.3007905.
- [115] A. A. Alsaffar, H. P. Pham, C. S. Hong, E. N. Huh, and M. Aazam, “An Architecture of IoT Service Delegation and Resource Allocation Based on Collaboration between Fog and Cloud Computing,” *Mobile Information Systems*, vol. 2016, 2016, doi: 10.1155/2016/6123234.
- [116] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, “Resource provisioning for IoT services in the fog computing environment: An autonomic approach,” *Comput Commun*, vol. 161, pp. 109–131, Sep. 2020, doi: 10.1016/j.comcom.2020.07.028.
- [117] S. K. Battula, S. Garg, J. Montgomery, and B. Kang, “An Efficient Resource Monitoring Service for Fog Computing Environments,” *IEEE Trans Serv Comput*,

vol. 13, no. 4, pp. 709–722, Jul. 2020, doi: 10.1109/TSC.2019.2962682.

- [118] X. Deng, J. Li, E. Liu, and H. Zhang, “Task allocation algorithm and optimization model on edge collaboration,” *Journal of Systems Architecture*, vol. 110, Nov. 2020, doi: 10.1016/j.sysarc.2020.101778.
- [119] Y. K. Lin and C. S. Chong, “Fast GA-based project scheduling for computing resources allocation in a cloud manufacturing system,” *J Intell Manuf*, vol. 28, no. 5, pp. 1189–1201, Jun. 2017, doi: 10.1007/s10845-015-1074-0.
- [120] J. C. Guevara, R. da S. Torres, and N. L. S. da Fonseca, “On the classification of fog computing applications: A machine learning perspective,” *Journal of Network and Computer Applications*, vol. 159, Jun. 2020, doi: 10.1016/j.jnca.2020.102596.
- [121] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, “A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment,” *J Ambient Intell Humaniz Comput*, vol. 11, no. 11, pp. 4951–4966, Nov. 2020, doi: 10.1007/s12652-020-01768-8.
- [122] C. Li, J. Bai, and Y. Luo, “Efficient resource scaling based on load fluctuation in edge-cloud computing environment,” *Journal of Supercomputing*, vol. 76, no. 9, pp. 6994–7025, Sep. 2020, doi: 10.1007/s11227-019-03134-8.
- [123] Z. Al-Makhadmeh and A. Tolba, “SRAF: Scalable Resource Allocation Framework using Machine Learning in user-Centric Internet of Things,” *Peer Peer Netw Appl*, 2020, doi: 10.1007/s12083-020-00924-3.
- [124] X. Xiong, K. Zheng, L. Lei, and L. Hou, “Resource allocation based on deep reinforcement learning in IoT edge computing,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, Jun. 2020, doi: 10.1109/JSAC.2020.2986615.
- [125] Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, “Migration Modeling and Learning Algorithms for Containers in Fog Computing,” *IEEE Trans Serv Comput*, vol. 12, no. 5, pp. 712–725, Sep. 2019, doi: 10.1109/TSC.2018.2827070.
- [126] S. S. Lee and S. Lee, “Resource Allocation for Vehicular Fog Computing Using Reinforcement Learning Combined with Heuristic Information,” *IEEE Internet Things J*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020, doi: 10.1109/JIOT.2020.2996213.
- [127] T. Alfakih, M. M. Hassan, A. Gumaiei, C. Savaglio, and G. Fortino, “Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA,” *IEEE Access*, vol. 8, pp. 54074–54084, 2020, doi: 10.1109/ACCESS.2020.2981434.
- [128] Y. G. Kim and C. J. Wu, “Autoscale: Energy efficiency optimization for stochastic edge inference using reinforcement learning,” in *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, IEEE Computer Society, Oct. 2020, pp. 1082–1096. doi: 10.1109/MICRO50266.2020.00090.

- [129] Tianyu Yang, Yulin Hu, M. Cenk Gursoy, Anke Schmeink, and Rudolf Mathar, “Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks,” *Ultra-Reliable, Low-Latency and Low-Power Transmissions in the Era of Internet-of-Things*, pp. 1–5, 2018.
- [130] J. Wang, L. Zhao, J. Liu, and N. Kato, “Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach,” *IEEE Trans Emerg Top Comput*, vol. 9, no. 3, pp. 1529–1541, 2021, doi: 10.1109/TETC.2019.2902661.
- [131] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, “Optimal Auction For Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach,” Nov. 2017, [Online]. Available: <http://arxiv.org/abs/1711.02844>
- [132] S. Vimal, M. Khari, N. Dey, R. G. Crespo, and Y. Harold Robinson, “Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT,” *Comput Commun*, vol. 151, pp. 355–364, Feb. 2020, doi: 10.1016/j.comcom.2020.01.018.
- [133] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1657–1681, Jul. 2017, doi: 10.1109/COMST.2017.2705720.
- [134] Z. Wang, G. Xue, S. Qian, and M. Li, “CampEdge: Distributed Computation Offloading Strategy Under Large-Scale AP-Based Edge Computing System for IoT Applications,” *IEEE Internet Things J*, vol. 8, no. 8, pp. 6733–6745, Apr. 2021, doi: 10.1109/JIOT.2020.3026862.
- [135] Y. Zhang, X. Zhou, Y. Teng, J. Fang, and W. Zheng, “Resource Allocation for Multi-User MEC System: Machine Learning Approaches,” in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Dec. 2018, pp. 794–799. doi: 10.1109/CSCI46756.2018.00159.
- [136] Y. Wang *et al.*, “A Game-Based Computation Offloading Method in Vehicular Multiaccess Edge Computing Networks,” *IEEE Internet Things J*, vol. 7, no. 6, pp. 4987–4996, Jun. 2020, doi: 10.1109/JIOT.2020.2972061.
- [137] Y. Wang *et al.*, “A Game-Based Computation Offloading Method in Vehicular Multiaccess Edge Computing Networks,” *IEEE Internet Things J*, vol. 7, no. 6, pp. 4987–4996, Jun. 2020, doi: 10.1109/JIOT.2020.2972061.
- [138] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, “Machine Learning-Based Workload Orchestrator for Vehicular Edge Computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2239–2251, Apr. 2021, doi: 10.1109/TITS.2020.3024233.
- [139] F. Zeng, C. Liu, J. Tangjiang, and W. Li, “Deep Learning-Based Task Offloading

- for Vehicular Edge Computing,” 2021, pp. 291–298. doi: 10.1007/978-3-030-86137-7_32.
- [140] K. Lin, S. Pankaj, and D. Wang, “Task offloading and resource allocation for edge-of-things computing on smart healthcare systems,” *Computers & Electrical Engineering*, vol. 72, pp. 348–360, Nov. 2018, doi: 10.1016/j.compeleceng.2018.10.003.
- [141] H. Ning, Y. Li, F. Shi, and L. T. Yang, “Heterogeneous edge computing open platforms and tools for internet of things,” *Future Generation Computer Systems*, vol. 106, pp. 67–76, May 2020, doi: 10.1016/j.future.2019.12.036.
- [142] M. De Donno, K. Tange, and N. Dragoni, “Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog,” *IEEE Access*, vol. 7, pp. 150936–150948, 2019, doi: 10.1109/ACCESS.2019.2947652.
- [143] A. Shahidinejad and M. Ghobaei-Arani, “Joint computation offloading and resource provisioning for edge-cloud computing environment: A machine learning-based approach,” *Softw Pract Exp*, vol. 50, no. 12, pp. 2212–2230, Dec. 2020, doi: 10.1002/spe.2888.
- [144] M. Aazam, M. St-Hilaire, C. H. Lung, and I. Lambadaris, “PRE-Fog: IoT trace based probabilistic resource estimation at Fog,” *2016 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016*. pp. 12–17, 2016. doi: 10.1109/CCNC.2016.7444724.
- [145] F. Y. Okay, “Nesnelerin İnternetinde Sis Hesaplama Tabanlı Veri Kümeleme ve Yönlendirme Modelleri,” 2019.
- [146] X. Li and L. Da Xu, “A Review of Internet of Things - Resource Allocation,” *IEEE Internet of Things Journal*, vol. 8, no. 11. Institute of Electrical and Electronics Engineers Inc., pp. 8657–8666, Jun. 01, 2021. doi: 10.1109/JIOT.2020.3035542.
- [147] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, “Resource Management Approaches in Fog Computing: a Comprehensive Review,” *Journal of Grid Computing*, vol. 18, no. 1. Springer, Mar. 01, 2020. doi: 10.1007/s10723-019-09491-1.
- [148] R. Mahmud, K. Ramamohanarao, and R. Buyya, “Application Management in Fog Computing Environments,” *ACM Comput Surv*, vol. 53, no. 4, pp. 1–43, Jul. 2021, doi: 10.1145/3403955.
- [149] B. Kitchenham, “A procedure for analyzing unbalanced datasets,” *IEEE Transactions on Software Engineering*, vol. 24, no. 4, pp. 278–301, Apr. 1998, doi: 10.1109/32.677185.
- [150] D. Singh and B. Singh, “Investigating the impact of data normalization on classification performance,” *Appl Soft Comput*, vol. 97, p. 105524, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.

- [151] W. Z. Liu and A. P. White, “The Importance of Attribute Selection Measures in Decision Tree Induction,” *Mach Learn*, vol. 15, no. 1, pp. 25–41, 1994, doi: 10.1023/A:1022609119415.
- [152] Witten Ian H, Frank Eibe, and Trigg Len, “Weka: Practicle Machine Learning Tools and Techniques with Java Implementations,” 1999.
- [153] J. R. Quinlan, “Induction of Decision Trees,” 1986.
- [154] IBM, “What is a Decision Tree?” Accessed: Nov. 20, 2022. [Online]. Available: <https://www.ibm.com/topics/decision-trees>
- [155] G. K. Uyanik and N. Güler, “A Study on Multiple Linear Regression Analysis,” *Procedia Soc Behav Sci*, vol. 106, pp. 234–240, Dec. 2013, doi: 10.1016/j.sbspro.2013.12.027.
- [156] C. Sonmez, A. Ozigovde, and C. Ersoy, “EdgeCloudSim: An environment for performance evaluation of edge computing systems,” *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, Nov. 2018, doi: 10.1002/ett.3493.
- [157] W. Junior, E. Oliveira, A. Santos, and K. Dias, “A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment,” *Future Generation Computer Systems*, vol. 90, pp. 503–520, Jan. 2019, doi: 10.1016/j.future.2018.08.026.
- [158] Y. Sun *et al.*, “Adaptive Learning-Based Task Offloading for Vehicular Edge Computing Systems,” *IEEE Trans Veh Technol*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019, doi: 10.1109/TVT.2019.2895593.
- [159] R. Shahzadi¹ and S. M. A. , Ambreen Niaz¹ , Mudassar Ali¹, *, Muhammad Naeem² , Joel J.P.C. Rodrigues³, 4, 5, 6 , Farhan Qamar¹, “Three tier fog networks: Enabling IoT/5G for latency sensitive applications,” *China Communications*, vol. 16, no. 3. pp. 1–11, 2019. doi: 10.12676/j.cc.2019.03.001.
- [160] H. Santos *et al.*, “A multi-tier fog content orchestrator mechanism with quality of experience support,” *Computer Networks*, vol. 177, p. 107288, Aug. 2020, doi: 10.1016/j.comnet.2020.107288.
- [161] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Trans Syst Man Cybern*, vol. 21, no. 3, pp. 660–674, 1991, doi: 10.1109/21.97458.
- [162] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
- [163] M. Jammal and M. AbuSharkh, “Machine Learning for Edge-Aware Resource Orchestration for IoT Applications,” in *2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, IEEE, Dec. 2021, pp. 37–44. doi:

10.1109/GCAIoT53516.2021.9692940.

- [164] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, “Machine Learning-Based Workload Orchestrator for Vehicular Edge Computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2239–2251, Apr. 2021, doi: 10.1109/TITS.2020.3024233.
- [165] F. R. Johnston, J. E. Boyland, M. Meadows, and E. Shale, “Some properties of a simple moving average when applied to forecasting a time series,” *Journal of the Operational Research Society*, vol. 50, no. 12, pp. 1267–1271, Dec. 1999, doi: 10.1057/palgrave.jors.2600823.
- [166] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, “Recent Advances in Collaborative Scheduling of Computing Tasks in an Edge Computing Paradigm,” *Sensors*, vol. 21, no. 3, p. 779, Jan. 2021, doi: 10.3390/s21030779.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Muhammet TAY

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Bilgisayar Müh.	Düzce Üniversitesi	2024
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2018
Lisans	Bilgisayar Müh. (İng.)	İstanbul Aydın Üniversitesi	2016
Lise		Adana Anadolu İHL	2007

TEZ YAYINLARI

- [1] Tay M.,& Şentürk A. "Kenar, Sis ve Bulut Bilişimin IoT Açısından İncelenmesi," *Avrupa Bilim Ve Teknoloji Dergisi* (32), 68-75, 2021, doi: <https://doi.org/10.31590/ejosat.1040133>.
- [2] Tay M.,& Şentürk A. "A research on resource allocation algorithms in content of edge, fog and cloud," *Materials Today*, Volume 81, Part 1,2023, Pages 26-34, <https://doi.org/10.1016/j.matpr.2022.11.232>.
- [3] Tay M.,& Şentürk A. "Decision tree-based task offloading in vehicle edge computing," *Concurrency and Computation: Practice and Experience: Proceedings*, <https://doi.org/10.1002/cpe.8026>.