



**T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**IoT TABANLI AKILLI ŞEHİRLERDE DERİN ÖĞRENME VE
MOBİL TABANLI AKILLI PARK SİSTEMİ YAKLAŞIMI**

HİKMET CANLI

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
Doç. Dr. Sinan TOKLU**

DÜZCE, 2022

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

IoT TABANLI AKILLI ŞEHİRLERDE DERİN ÖĞRENME VE
MOBİL TABANLI AKILLI PARK SİSTEMİ YAKLAŞIMI

Hikmet CANLI 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. Sinan TOKLU

Gazi Üniversitesi

Jüri Üyeleri

Doç. Dr. Sinan TOKLU

Gazi Üniversitesi

Prof. Dr. Pakize ERDOĞMUŞ

Düzce Üniversitesi

Doç. Dr. İbrahim Alper DOĞRU

Gazi Üniversitesi

Prof. Dr. O. Ayhan ERDEM

Gazi Üniversitesi

Doç. Dr. A. TALHA KABAKUŞ

Düzce Üniversitesi

Tez Savunma Tarihi: 01/04/2022

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.

01 Nisan 22

Hikmet CANLI



TEŐEKKÖR

Doktora öđrenimimde ve bu tezin hazırlanmasında gösterdiđi her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Sinan TOKLU'ya en içten dileklerle teşekkür ederim.

Tez çalışmam boyunca değerli katkılarını esirgemeyen TİK üyelerim Prof. Dr. Pakize ERDOđMUŐ ve Doç. Dr. İbrahim Alper DOđRU'ya şükranlarımı sunarım.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili eşime, aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

“Sürdürülebilir ve Akıllı Kentler” kapsamı altında bu çalışmayı destekleyen YÜKSEKÖđRETİM KURULUNA teşekkürlerimi sunarım.

01 Nisan 2022

HİKMET CANLI

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	vii
KISALTMALAR.....	viii
SİMGELER	ix
ÖZET	x
ABSTRACT	xi
EXTENDED ABSTRACT	xii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI.....	4
3. MATERYAL VE YÖNTEM	10
3.1. ISTPARK VERİ SETİ.....	10
3.2. DERİN ÖĞRENME TABANLI OTO PARK DOLULUK TAHMİNİ	14
3.2.1. RNN Derin Öğrenme Mimarisi.....	15
3.2.2. LSTM Derin Öğrenme Mimarisi	16
3.2.3. GRU Derin Öğrenme Mimarisi	19
3.3. MAKİNE ÖĞRENME TABANLI OTO PARK DOLULUK TAHMİNİ	22
3.3.1. Rastgele Orman Makine Öğrenme Yöntemi (RF)	23
3.3.2. Destek Vektör Makineleri Öğrenme Yöntemi (SVM)	24
3.3.3. Bütünleşik Otoregresif Hareketli Ortalama Makine Öğrenme Yöntemi (ARIMA)	25
3.4. AKILLI PARK SİSTEMİ TASARIMI	25
3.4.1. Önerilen Model.....	25
3.4.2. Model Eğitimi ve Değerlendirme	31
3.4.2.1. LSTM, RF, SVM ve ARIMA Modellerinin Değerlendirilmesi	35
3.4.2.2. GRU, LSTM ve RNN Modellerinin Değerlendirilmesi	40
4. BULGULAR VE TARTIŞMA	47
5. SONUÇ VE ÖNERİLER	49
6. KAYNAKLAR	50
ÖZGEÇMİŞ	56

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 3.1. İstanbul'daki Otoparkların Konumları	11
Şekil 3.2 Algoritma 1: Veri Seti Optimizasyonu	12
Şekil 3.3 Otopark Doluluk Dağılım Grafiği	14
Şekil 3.4 Günlük Ortalama Otopark Doluluk Grafiği.....	14
Şekil 3.5 RNN Mimari.....	16
Şekil 3.6 LSTM Tekrarlayan Ağ	17
Şekil 3.7 LSTM Mimari.....	18
Şekil 3.8 GRU Mimari.....	20
Şekil 3.9 Makine Öğrenmesi Yöntemleri	22
Şekil 3.10 Rastgele Orman Algoritması Çalışması	23
Şekil 3.11 Destek Vektör Makineleri Sınıflandırma gösterimi	24
Şekil 3.12 ISTPARK Mobil Tasarımı.....	26
Şekil 3.13 Model Tasarımı.....	28
Şekil 3.14 Önerilen Yöntemin Akış Diyagramı.....	29
Şekil 3.15 Algoritma 2: Önerilen Yöntem.....	30
Şekil 3.16 Rastgele Orman Otopark Doluluk Tahmini.....	36
Şekil 3.17 Destek Vektör Makinesi Otopark Doluluk Tahmini	36
Şekil 3.18 ARIMA Otopark Doluluk Tahmini	37
Şekil 3.19 LSTM Otopark Doluluk Tahmini.....	37
Şekil 3.20 Doğruluk Kutu Grafiği	39
Şekil 3.21 Derin Öğrenme Yoluyla Otopark Doluluk Tahmininin Akış Şeması	41
Şekil 3.22 LSTM, GRU, RNN Veri Modeli	42
Şekil 3.23 RNN ile Otopark Doluluk Tahmini	42
Şekil 3.24 LSTM ile Otopark Doluluk Tahmini.....	43
Şekil 3.25 GRU ile Otopark Doluluk Tahmini	44
Şekil 3.26 Epoch – Loss Grafiği	46

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 2.1. Akıllı Park Sistemleri ile İlgili Başlıca Çalışmalar	8
Çizelge 3.1. ISTPARK Veri Seti Parametreleri ve Özellikleri	11
Çizelge 3.2 Model Parametreleri	31
Çizelge 3.3 Model Eğitiminde Kullanılan Parametreler	33
Çizelge 3.4 Farklı Özellikler için Performans Karşılaştırması	38
Çizelge 3.5 Modellerin Hata Değerlendirmesi	40
Çizelge 3.6 Modellerin Hata Değerlendirmesi	44
Çizelge 4.1 Model Karşılaştırılması	48



KISALTMALAR

ACC	Accuracy (Doğruluk)
AKOM	Afet Koordinasyon Merkezi
ARIMA	Autoregressive Integrated Moving Average (Bütünleşik Otoregresif Hareketli Ortalama)
AUC	Area Under Curve (Eğri Altındaki Alan)
BİT	Bilgi ve İletişim Teknolojileri
BST	Binary Search Tree (İkili Arama Ağacı)
BT	Bilgi teknolojileri
CB-SPS	Cloud based – Smart Parking Server (Bulut tabanlı – Akıllı Park Sunucusu)
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
EKF	Extended Kalman Filtre (Genişletilmiş Kalman Filtresi)
GPS	Global Positioning System (Global Konumlandırma Sistemi)
GRU	Gated Recurrent Unit (Geçitli Tekrarlayan Ünite)
İBB	İstanbul Büyükşehir Belediyesi
INS	Inertial Navigation System (Ataletsel Navigasyon Sistemi)
IoT	Internet of Things (Nesnelerin İnterneti)
İHA	İnsansız Hava Aracı
JSON	JavaScript Object Notation (JavaScript Nesnesi Gösterimi)
LSTM	Long short-term memory (Uzun Kısa Süreli Bellek)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
MAPE	Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hata)
MSE	Mean Squared Error (Ortalama Kare Hatası)
NLP	Natural Language Processing (Doğal Dil İşleme)
NN	Neural Network (Sinir Ağı)
ODP	İBB Open Data Portal (İBB Açık Veri Portalı)
RF	Random Forest (Rastgele Orman)
RFID	Radio Frequency Identification (Radyo Frekansı ile Tanımlama)
RFR	Random Forest Regression (Rastgele Orman Regresyonu)
RMSE	Root Mean Square Error (Kök Ortalama Kare Hatası)
RNN	Recurrent Neural Network (Yinelemeli Sinir Ağı)
SOA	Service Oriented Architecture (Hizmet Odaklı Mimari)
SPARK	VANET tabanlı akıllı park planı
SVM	Support Vector Machines (Destek Vektör Makinesi)
SVPS	Bulut tabanlı akıllı araç park sistemi
TWSVM	Twist Support Vector Machines (İkiz Destek Vektör Makinesi)
VANET	Vehicular ad-hoc network (Araçlı Geçici Ağlar)
WSN	Wireless Sensor Network (Kablosuz Sensör Ağları)
YSA	Yapay Sinir Ağı

SİMGELER

d	Bütünleşme Mertebesi
e	Bitiş Tarihi
ED	Bitiş Veri Seti
f	Zaman Frekansı
p	Oto korelasyonun Mertebesi
P	Park Listesi
q	Hareketli Ortalama Mertebesi
s	Başlangıç Tarihi
$s(t)$	Gizli Katman Sayısı
SD	Başlangıç Veri Seti
T	Boş Zaman Serisi
$\tanh$	Aktivasyon Fonksiyonu
x_t	Mevcut Giriş
h_t	Akım Çıkışı
C_t	Mevcut Durum
f_t	Unut Kapısı Aktivasyon Fonksiyonu
i_t	Çıkış Kapısı Aktivasyon Fonksiyonu
$\hat{C}_t$	Hücre Giriş Aktivasyon
W	Ağırlık Matrisi
z_t	Güncelleme Kapısı Aktivasyon Fonksiyonu
r_t	Reset Kapısı Aktivasyon Fonksiyonu

ÖZET

IoT TABANLI AKILLI ŞEHİRLERDE DERİN ÖĞRENME VE MOBİL TABANLI AKILLI PARK SİSTEMİ YAKLAŞIMI

Hikmet CANLI

Düzce Üniversitesi

Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Doç. Dr. Sinan TOKLU

Nisan 2022, 55 sayfa

Nesnelerin İnterneti (IoT) ve akıllı şehir ekosistemleri çağında, daha sürdürülebilir şehirler için yenilikçi akıllı park sistemlerine ihtiyaç duyulmaktadır. Her geçen gün artan araç sayısı ile akıllı park sistemleri akıllı şehirlerde önemli konular arasında yer almaktadır. Bunun nedeni yetersiz olan park yeri arayışlarının ciddi bir maliyet, hava kirliliği ve stres sorunlarını beraberinde getirmesidir. Akıllı park sistemleri ile ilgili yapılan araştırmalar bulunan çözümler artık yeterli gelmemektedir. Bu tez çalışmasında sürücülerin park yeri arama sorununu en aza indirmek için derin öğrenme ve bulut tabanlı yeni bir mobil akıllı park uygulama tasarımı geliştirilmiştir. Uygulama içerisinde, park yerini tahmin etmek için LSTM ile derin öğrenmeye dayalı bir hizmet geliştirilmiştir. Burada kullanıcının mobil cihazı üzerinden daha önce oluşturulan LSTM tabanlı modele dinamik erişim sağlanmakta ve ilgili parametreler girilerek parkların istenilen yerdeki doluluk oranlarının mobil cihaz üzerinde görüntülenmesi işlemi gerçekleştirilmektedir. Bu sayede hem enerji hem de zaman tasarrufu sağlanmıştır. Türkiye'de İstanbul ilinde gerçek zamanlı toplanan otopark verileri ile %99,57 doğrulukta sonuç elde edilmiştir. Önerilen modelin etkinliğini göstermek için makine öğrenmesi modelleri olan, SVM, RF ve ARIMA yöntemleri ile karşılaştırılmıştır. Tez çalışmasının devamında otopark kullanımında önemli etkisi olan hava durumu verileri AKOM'dan alınarak otopark verileri ile birleştirilmiştir. Zaman serileri tahmini problemlerinde yaygın olarak kullanılan ve başarısını ispatlamış olan GRU, LSTM ve RNN yöntemlerinden oluşturulan 27 farklı model ile bu veri seti üzerinden deneyler yapılmıştır. Elde edilen sonuçlarda GRU derin öğrenme modeli %99,11 doğruluk ve 0,90 MAE, 2,35 MSE ve 1,53 RMSE metrik değerleri ile en iyi sonucu vermiştir. Çeşitli hiper parametrelerle elde edilen deneysel sonuçlar, GRU derin öğrenme modelinin park doluluk oranlarını tahmin etmedeki başarısını açıkça göstermektedir. Elde edilen sonuçlar incelendiğinde derin öğrenme modellerinin park yeri tahminindeki yüksek doğruluk oranı, daha sürdürülebilir akıllı şehirler için akıllı park sistemlerinde kullanılabileceğini göstermiştir.

Anahtar sözcükler: Akıllı Şehirler, Derin Öğrenme, Makine Öğrenmesi, Nesnelerin İnterneti

ABSTRACT

DEEP LEARNING AND MOBILE-BASED SMART PARKING SYSTEM APPROACH IN IoT-BASED SMART CITIES

Hikmet CANLI

Düzce University

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

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Sinan TOKLU

April 2022, 55 pages

In the era of Internet of Things (IoT) and smart city ecosystems, innovative smart parking systems are needed for more sustainable cities. With the increasing number of vehicles, smart parking systems are among the important issues in smart cities. The reason for this is that the search for insufficient parking places brings with it serious cost, air pollution and stress problems. The solutions that have been researched on smart parking systems are no longer sufficient. In this thesis, a new deep learning and cloud-based mobile smart parking application design has been developed to minimize the problem of drivers searching for parking spaces. Within the application, a service based on deep learning with LSTM has been developed to prediction the parking location. Here, dynamic access to the previously created LSTM-based model is provided via the user's mobile device, and the related parameters are entered and the occupancy rates of the parks are displayed on the mobile device. In this way, both energy and time savings were achieved. With real-time parking data collected in Istanbul, Turkey, results with an accuracy of 99.57% were obtained. To demonstrate the effectiveness of the proposed model, it was compared with machine learning models, SVM, RF and ARIMA methods. In addition, the weather data, which has a significant impact on the use of the parking lot, was taken from AKOM and combined with the parking data. Experiments were carried out on this data set with 27 different models created from RNN, LSTM and GRU methods, which are widely used in time series prediction problems and have proven their success. GRU deep learning model gave the best results with 99.11% accuracy and 0.90 MAE, 2.35 MSE and 1.53 RMSE metric values. Experimental results with various hyper parameters clearly demonstrate the success of the GRU deep learning model in predicting park occupancy rates. When the results obtained are examined, it has been shown that deep learning models can be used in smart parking systems for more sustainable smart cities, with high accuracy in parking IoT prediction.

Keywords: Deep Learning, Internet of Things, Machine Learning, Smart Cities

EXTENDED ABSTRACT

DEEP LEARNING AND MOBILE-BASED SMART PARKING SYSTEM APPROACH IN IoT-BASED SMART CITIES

Hikmet CANLI

Düzce University

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

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Sinan TOKLU

April 2022, 55 pages

1. INTRODUCTION

Today, the number of people in big cities is increasing rapidly. In parallel with this speed, problems such as traffic density, insufficient parking capacity, lack of parking and infrastructure problems arise in cities. In addition, technology is progressing day by day, internet of things (IoT), smart devices, machines, software, systems are working to use city resources more efficiently and to become smarter. In the thesis study, deep learning and mobile-based smart parking system architecture was developed to solve the parking problem in big cities. In the created architecture, end users will make a real-time request to the deep learning service that prediction the parking occupancy rate through the designed mobile application and will be able to view the prediction results. The ISTPARK dataset collected from the parking lots and the weather data from AKOM were used to predict the parking occupancy rate. The models for predicting parking occupancy rate were evaluated in two different scenarios. In the first scenario, the LSTM deep learning model and the SVM, RF and ARIMA machine learning methods, which have proven themselves in the time series problem, are run on the ISTPARK dataset. When the experimental results obtained were examined, the LSTM model gave the best result with an accuracy rate of 99.57%. In the experiments, ARIMA gave 92.90%, RF 89.34% and SVM 87.01% accuracy rates in the models with the best performance. In the second scenario, experiments were carried out by running the LSTM model on GRU and RNN deep learning models, on ISTPARK and weather data. In the results obtained, while GRU and LSTM gave close results, RNN lagged behind in performance.

2. MATERIAL AND METHODS

With the development of deep learning methods in smart cities, previously used parking solutions have started to become obsolete. As resources become more and more limited, the need for innovative solutions becomes constant. Istanbul city of Turkey was used as the pilot region in the study. It is important that Istanbul has the most populous population and the heaviest traffic in the country in the selection of the pilot region. The ISTPARK dataset was created with 731 parking lots and weather data in Istanbul to be used in parking occupancy predictions. Parking lot data is provided by Istanbul Metropolitan Municipality parking information service, and weather data is provided by Istanbul Disaster Coordination Center. The created data set consists of 19 parameters and contains 4.5 million records. Data is stored on SQL Server 2014 installed on the cloud. In the study, experiments were carried out by establishing different models with deep learning and machine learning methods to predict the parking occupancy rate on the dataset. In the study, in addition to deep learning methods LSTM, GRU and RNN, SVM, RF and ARIMA machine learning methods, which have proven themselves in time series problems, were used. Experimental results were evaluated with MAE, MSE, RMSE and Accuracy metrics. Keras, a neural network library, was used for modeling in the thesis study. Jupiter Notebook was chosen as the compiler program. Model experiments were performed on a computer with Intel (R) UHD Graphics with Intel (R) Core (TM) i7-8650U processor (maximum speed: 2.11 GHz), 16 GB ram and 620 graphics card.

3. RESULTS AND DISCUSSIONS

In this thesis, a new cloud and deep learning-based architecture and a new mobile application are proposed for parking occupancy prediction. Two different scenarios and datasets were used to predict the parking occupancy rate. In the first scenario, deep learning-based LSTM model and ISTPARK dataset are used, which model multivariate and large datasets almost seamlessly. In order to show the effectiveness of the model, models were created with different parameters and different scenarios and compared with SVM, Random Forest and ARIMA models. In the best combination of experiments, LSTM showed the best performance with an accuracy rate of 99.57%. RF gave 89.34%, SVM 87.01% and ARIMA 92.90% accuracy values. In the second scenario, the LSTM model, which is superior to machine learning methods, is compared with the GRU and RNN deep learning models. In addition, the weather data, which significantly affects the parking usage, were combined with the ISTPARK dataset and experiments were carried

out. As a result of these experiments, the GRU deep learning model gave the best performance with an accuracy rate of 99.11%. When the simulation results obtained from two different scenarios are examined, it is clearly seen that the architecture and application we have created give successful results.

4. CONCLUSION AND OUTLOOK

Developed smart cities and IoT devices constantly generate data and enable us to access this data more easily. Deep learning techniques have become a frequently used method in the big data era we live in. With the increasing population of smart cities, traffic density has increased and parking areas have become insufficient. In this study, a new mobile-based smart parking system architecture based on deep learning has been developed for smart cities. When the experiments and the findings are examined, LSTM and GRU deep learning methods have provided results that will give confidence to the end user in the proposed smart parking system architecture. In addition, the data set created, the architecture of the smart parking system and the methods used have contributed to the literature in different fields. It is thought that the slower pace of vehicles while searching for a parking space in front of the parking lot causes an increase in traffic density. In future studies, the layout of the parking lots can be realized by using deep learning and optimization methods in smart city planning.

1. GİRİŞ

Dünyadaki kentsel nüfus 1950'de 751 milyondan 2018'de 4,2 milyara hızla yükseldi. Son on yılda dünya nüfusunun yarısından fazlası (%55) şehirlerde yaşamaktadır [1]. Şehirlerde artan nüfus yoğunluğu ve azalan kaynaklar şehirlerin akıllandırılması ve verimliliğinin artırılmasını gerektirmektedir. Akıllı şehirler kavramı ilk olarak 1990'lı yıllarda yayınlanmış ve şehirlerde BİT (Bilgi ve İletişim Teknolojileri) ve modern altyapıların kullanımına atıfta bulunulmuştur [2]. Günümüzde nesnelerin internetin (IoT) kullanımının artmasıyla her alanda bu kavram genişlemektedir. Şehirlerin ulaşımında, havaalanlarında, hastane, alışveriş merkezi vb. birçok alanda IoT uygulamaları hızla artmaktadır. 2021 yılı itibarıyla IoT 'ye bağlı cihaz sayısı 27 milyon cihazı geçmiştir ve bu Bilgi teknolojileri (BT) dünyasında muazzam bir rakam olarak karşımıza çıkmaktadır [3]. Bu teknolojinin kullanılmasıyla beraber günlük yaşam son on yıla göre daha kolay bir hale gelmiştir [4]. Hızla artan nüfus, altyapı eksikliğinden kaynaklanan yoğun trafik yükleri, ulaşım ve otopark sorunları her zaman en önemli sorunlar olmuştur. Bu sorunların bir sonucu olarak zaman kaybedilmekte, maliyetler artmakta ve insanlar trafikte yorulmakta ve stres yükleri artmaktadır [5]. IoT tabanlı akıllı şehirlerin önemli konu başlıkları arasında akıllı otopark sistemleri yer almaktadır. Akıllı otopark sistemlerinin önemli hale gelmesinde şehirlerde artan araç sayısı bulunmaktadır. Her gün trafiğe yeni katılan araç sayısı ile birlikte oluşan park sorunları son yıllarda daha fazla dikkat çekmiştir [6,7].

İstanbul, 15,5 milyonluk nüfusuyla Türkiye'nin en büyük metropol bölgesidir ve ülkenin ekonomik, kültürel ve tarihi merkezi olarak kabul edilmektedir [8]. Burada yaşayan insanların en büyük problemi özellikle işe gidiş ve işten dönüş saatlerinde yaşadıkları trafik yoğunluğu ve park yeri bulamama sorunudur [5]. Yapılan araştırmalar gösteriyor ki; Çok sayıda aracın bulunduğu şehirlerde insanlar park yeri bulmak için 3,5 ila 14 dakika harcıyor [9][10]. Şehir sakinlerini endişeyle park yeri aramaktan rahatsız edici çok az şey olabilir. Literatürde otopark ve park yeri arama sorunları üzerinde farklı çözüm yöntemleri sunulmuştur. Bu yöntemler kısa vadede iyi sonuç verse de trafiğe katılan araç sayısının her geçen yılda daha fazla artmasıyla yeni çözüm yöntemlerine ihtiyaç duyulmaktadır.

Bu tez çalışmasında Türkiye'nin İstanbul kentindeki park yeri arama problemini çözmek için derin öğrenme ve bulut tabanlı çalışan yeni bir yöntem ve mobil uygulama tasarımı önerilmektedir. Kullanıcılar mobil uygulama ile derin öğrenme yöntemi kullanılarak geliştirilen modele dinamik olarak erişebilmekte ve gerçek zamanlı olarak ileriye dönük sonuçları hızlı şekilde görebilmektedir. Yapılan tez çalışması iki aşamalıdır. Birinci aşamada İstanbul'da bulunan 731 otoparktan alınan kapasite, boş kapasite, enlem, boylam vb. bilgilerin anlık olarak alınması ve bulutta depolanması işlemi gerçekleştirilmektedir. İkinci aşamada mobil uygulamadan gelen park yeri öneri isteği için derin öğrenme tabanlı LSTM (Long short-term memory) servisi devreye girmektedir. Bu servis uygulamadan gelen isteğe göre ilgili otoparkın doluluk oranını kısa ve orta vadede tahmin edip cevap vermektedir.

Oluşturulan tahmin modelin etkinliğinin gösterilmesi için makine öğrenmesi yöntemleri olan SVM (Support Vector Machines), RF (Random Forest) ve zaman serilerinin analizinde sıklıkla kullanılan ARIMA (Autoregressive Integrated Moving Average) yöntemlerinden oluşturulan modeller kullanılmıştır. Veri setinde bulunan parametreler 23 farklı grupta incelenerek deneyler yapılmıştır. Elde edilen deneysel sonuçlar incelendiğinde %99,57 doğruluk oranı ile LSTM modeli diğer makine öğrenmesi modellerine göre en iyi sonucu vermiştir. Yapılan deneylerde, en iyi performansın elde edildiği modellerde ARIMA %92,90, RF %89,34 ve SVM %87,01 doğruluk sonucu alınmıştır. Burada otopark doluluk tahmini için önerilen derin öğrenme modelinin, zaman serilerinin tahmininde sıkça kullanılan makine öğrenmesi yöntemlerine göre başarısı açıkça görülmektedir. Tabi ki literatürde de zaman serilerinin tahmininde kullanılan birçok derin öğrenme modeli bulunmaktadır. Yapılan tez çalışmanın devamında otoparklardan alınan veriler ile otopark kullanımını önemli ölçüde etkileyen hava durumu verileri AKOM (Afet Koordinasyon Merkezi) istasyonlarından alınarak birleştirilmiştir. Hava durumu verilerinin çalışmaya eklenmesinde, yağışlı ve karlı havalarda trafik yoğunluğunun artması, kazaların meydana gelmesi vb. durumların otopark kullanımını etkilemesi önemli rol almaktadır. Bunun yanında kötü hava koşullarında sahil, mesire alanı vb. bölgelere insanların daha az gittiği ve bu bölgedeki otoparkların daha az kullanıldığı gözlemlenmiştir. Oluşturulan veri seti üzerinde GRU (Gated Recurrent Unit), LSTM ve RNN derin öğrenme yöntemleri ile oluşturulan modellerle deneyler yapılmıştır. Çalışmada kullanılan veriler sırasıyla eğitim-test %70-30, %80-20 ve %90-10 olarak üç farklı gruba ayrılmıştır. Her bir veri setine 10, 20 ve 30 zaman adımlarında 3 farklı derin

öğrenme yöntemi uygulanarak 27 farklı model oluşturulmuştur. Test veri setleri üzerinde geliştirilen modellerin tahmin başarısı MAE (Mean Absolute Error), MSE (Mean Squared Error) ve RMSE (Root Mean Square Error) metrik değerlerinin hesaplanmasından sonra incelenmiştir. Düşük RMSE, MAE ve MSE değerleri, bu modellerin otoparkların tahmininde daha iyi bir seçim olacağını göstermiştir. Yapılan çalışmada GRU, en düşük MAE değeri 0,90 ve en düşük RMSE değeri 1,53 ile en iyi performans gösteren yöntem olduğu görülmüştür. Aynı zamanda %99,11 doğruluk oranı ile en iyi otopark tahminini yaptığı görülmüştür.

Özetle bu tez çalışmasının katkıları aşağıdaki gibidir:

- a) IoT tabanlı akıllı şehirlerde otopark doluluk tespiti için yeni bir mimari ve derin öğrenme tabanlı yeni bir uygulama geliştirilmiştir.
- b) LSTM derin öğrenme modeli ile çok değişkenli ve çok zaman serili otopark veri seti ile otopark doluluk tespitinde SVM, RF ve ARIMA'dan daha iyi sonuçlar elde edilmiştir.
- c) GRU, LSTM ve RNN derin öğrenme yöntemleri ile farklı modeller kurularak deneyler yapılmış, elde edilen sonuçlar incelenmiş ve GRU en iyi performansı göstermiştir.
- d) Bir yıl boyunca değişik parametrelerde toplanan otopark verileri ve AKOM'dan alınan hava durumu verileri ile literatürde başka çalışmalara ilham olacak ve kullanılabilecek büyük veri seti oluşturulmuştur.

Bu çalışmanın ikinci bölümünde literatürde akıllı park sistemleri üzerine yapılan çalışmalar detaylı olarak incelenmiştir. Daha sonra dört ana başlık çerçevesinde materyal ve yöntemler incelenmiştir. Birinci başlıkta ISTPARK veri setinin toplanması, özellikleri, depolanma süreci ve görselleştirilmesi ele alınmıştır. İkinci ve üçüncü başlıkta otopark doluluk tahmini için derin öğrenme ve makine öğrenmesi tabanlı yöntemler anlatılmıştır. Son olarak önerilen model, yapılan deneyler ve sonuçları anlatılmıştır. Daha sonra bulgular ve tartışma bölümü oluşturulmuş çalışmada elde edilen bulgular bu bölümde incelenmiş ve tartışılmıştır. Son bölümde çalışmadan elde edilen sonuçlar tartışılmış, sonraki çalışmalar için önerilerde bulunulmuştur.

2. LİTERATÜR TARAMASI

Şehirlerde her geçen gün nüfus artmakta kaynaklar ise sınırlı kalmaktadır. Bu sınırlı kaynakları yönetmek için sürdürülebilir yenilikçi çözümlere ihtiyaç bulunmaktadır. [11]. Şehirler de artan araç sayısına, nesnelerin interneti ile beraber yenilikçi çözümler gerekmektedir [12][13]. Park yeri probleminin çözümü ile ilgili literatürde birçok çalışma bulunmaktadır. Bu çalışmalar genel olarak üç ana başlık altında incelenebilir. Birinci başlık rezervasyon ve algoritma tabanlı yaklaşımlar, ikinci başlık otoparklardaki boş park yerlerinin tespiti, üçüncü başlıkta ise orta vadede mevcut boş park kapasitesinin tahminine yönelik çalışmaları kapsamaktadır. Bu bölümde literatürde akıllı park sistemleri ile ilgili yapılan çalışmalar detaylı bir şekilde incelenmiştir.

[15]'de yapılan çalışmada sürücüler kentsel alanlardaki otoparklara yönlendirmenin yolları araştırılmıştır. Statik ve dinamik yönlendirmeden bahsedilmiş ve bu yöntemlerin avantajları ve dezavantajları tartışılmıştır. [16]'da akıllı park rehberlik sistemi tanıtılmıştır. Park seçiminde etkili olan faktörleri dikkate alıp akıllı park rehberlik algoritması önerilmiştir. Önerilen algoritma ve yöntem ile araç sürücülerinin en uygun otoparkı bulması sağlanmıştır. [17]'de akıllı park yeri rehberlik çalışması Boston üniversitenin garajında pilot bir bölgede yapılmıştır.

[18]'de yapılan çalışmada park yeri hizmeti sağlayanlar ile sürücüler bir araya getirmek için bulut tabanlı bir mimari oluşturulmuştur. Park yeri hizmeti sağlayanlar bu sayede reklamlarını yapabilmektedirler. [13]'de park yeri arama işlemi yapılırken kullanılan algoritmaların performansı enerji ve süre yönünden incelenmiştir. Hiyerarşik tabanlı BST algoritması kullanılmış ve hiyerarşik olmayan yönlerden daha verimli olduğu gösterilmiştir.

[19]'da hafta sonları yoğunluk yaşanan alışveriş merkezlerinin otoparklarındaki problemin çözümü için çalışma yapılmıştır. Bir aracın park yerine yerleştirilmesi için genetik yaklaşım kullanılmıştır. Sürücülerin en uygun park yerlerine park etmesi ile kuyrukta bekleme sürelerini azaltmışlardır.

[20]'de Daha sağlam park çözümleri sağlamak için, her yerde bulunan VANET'ler üzerinden yeni bir bulut tabanlı akıllı araç park sistemi (SVPS) önerilmiştir. Önerilen SVPS mimarisi, araçları etkili, gerçek zamanlı ve kesin bir şekilde kolaylaştırmak için

rezervasyon ve öneri seçeneklerinin yanı sıra uygun bir boş park yeri bilgisi sağlayan benzersiz bir algoritma sunmaktadır. Çalışmada yapılan simülasyonlar önerilen mimarinin park kaynaklarını en iyi şekilde kullandığını göstermektedir. [21]'de büyük otoparklar için VANET tabanlı akıllı park planı (SPARK) önerilmiştir. SPARK şeması ile bir otoparkta bulunan bütün RSU'lar üzerinde otoparkı izlemişlerdir. Elde ettikleri veriler ile gerçek zamanlı park navigasyonu, akıllı hırsızlık koruması ve dostça park bilgilerinin yayılmasını sağlamışlardır.

[22]'de yapılan çalışmada, akıllı park entegre yönetimi için SOA (Hizmet Odaklı Mimari) tabanlı bir bulut platform mimarisi önerilmiştir. Akıllı parkların bilgi alt sistemlerindeki "veri adası", "güç adası" ve "kullanıcı deneyimi" sorunlarını çözmeye odaklandılar. Deneylerinde, birden fazla kullanıcı aynı anda istek başlattığında mimarinin sağlam yürütme verimliliği ve ölçeklenebilirlik sergilediğini kanıtlamışlar. SOA, aynı zamanda IoT uygulamalarında birlikte çalışabilirliği ele almak için en uygun mimari yollardan biridir. Yüksek düzeyde ölçeklenebilir bir altyapı sağlar [23][24]. Şehir altyapısının bileşenlerini destekleyen işlevleri geliştirme, dağıtma ve yönetme yeteneğine sahip standart bir çerçeve aracılığıyla şehir hizmetlerini entegre etmek ve yönetmek için kullanılır [25].

[26]'da konuya farklı bir yönden bakılmıştır. Park yerlerinin tespiti için akıllı telefonlar kullanılmıştır. Bir sürücü park ettiğinde veya park yerinden ayrıldığında ilgili park yeri tespit edilebiliyor. Kapalı otoparklarda bile yüksek doğruluk oranı vermektedir. [27]'de otoparklarda park yerlerinin dolu olup olmadığını kontrol eden sensörlerin performansları incelenmiştir. Bu sensörler ışık açısına, uzaklığa göre farklı performanslar göstermektedir. [28]'de otopark giriş çıkışlarındaki personel kaynaklı oluşan trafiğin azaltılması konu alınmıştır. Giriş ve çıkışlara yerleştirilen RFID tarayıcılar ile otomatik bariyer sistemi geliştirilmiştir. Personel maliyetini ve giriş çıkışlardaki yoğunluğu azaltmışlardır.

[29]'da rezervasyon tabanlı bir yöntem oluşturulmuştur. Otoparktaki her bir alanı IoT cihazları üzerinden izlemişler ve mobil uygulama üzerinden rezervasyon sistemi oluşturmuşlardır. Otoparklarda IoT cihazı kullanımı güvenlik problemlerini de beraberinde getirmektedir. [30]'da bu güvenlik problemleri vurgulanmış ve eliptik şifreleme yöntemi önerilmiştir. Diğer şifreleme yöntemleri ile karşılaştırılmış daha verimli olduğu gösterilmiştir.

[31]'de park altyapılarının akıllı hizmetler açısından sürücülerin ihtiyaçlarına yönelik bir anketi sunmaktadır. Hem sürülere hem de park operatörlerine fayda saylayacak otopark izleme, park rezervasyonu, fiyatlar ve son trendler tartışılmıştır. Ayrıca, hem sürücülere hem de otopark işletmecilerine fayda sağlamak için teknolojik olarak gelişmiş park altyapılarının entegrasyonu da analiz edilmiştir. [32]'de park yerleri için optimizasyon üzerinde çalışılmıştır. Wireless Sensor Networks (WSN) aracılığıyla araçları boyutlarına göre gruplandırarak uygun park yerlerine yerleştirip park alanlarının daha verimli kullanılmasını önermişlerdir.

IoT teknolojisinin gelişmesiyle beraber akıllı şehirlerde WSN'ler daha yaygın kullanılmaya başlandı. Kaynak yerleştirme ve hedef izleme, WSN'ler için en zorlu problemler arasındadır. Makine öğrenmesi yöntemleri kullanılarak bu problemlere çözüm üreten araştırmalar giderek artmakta. [35]'de yapılan çalışmada WSN'ler için kaynak yerleştirme ve hedef izleme maliyetini en aza indirmek için SVM ve ikiz SVM (TWSVM) yöntemlerini önermişlerdir. Yaptıkları deneylerle de olay algılama bölgesinin tespitinde önerdikleri yöntemin maliyet azaltma etkinliğini göstermişlerdir. Ancak yapılan deneyler tek olay bölgesi varsayılarak yapılmıştır. Çok kaynaklı çalışmalar için gelecekte k-ortalamalı kümeleme yöntemini uygulayacaklarını belirtmişlerdir. [33]'te yapılan çalışmada akıllı şehirlerde araç konumlarının bulma maliyetini düşürmek için daha düşük maliyetli sensörlerden makine öğrenmesi yöntemlerini kullanarak daha yüksek verim almaya çalışmışlardır. Global Konumlandırma Sistemi (GPS) yanında Ataletsel Navigasyon Sistemlerinden (INS) daha uygun maliyetli Genişletilmiş Kalman Filtresi (EKF) kullanmışlardır. Ancak EKF performansı aracın dinamik değişikliklerine bağlıdır ve çevresel değişiklikler nedeniyle hızla farklılaşabilir. EKF'nin dezavantajlarını sınırlandırarak araç konumunu güvenilir bir şekilde tahmin etmek için EKF ve SVM kullanarak sağlam ve düşük maliyetli bir yaklaşım önermişlerdir. Elde ettikleri sonuçlarda EKF'nin sonuçlarına kıyasla konum doğruluğunda %94'e yakın iyileştirme sağlamışlardır. Ayrıca elde ettikleri deneysel sonuçlar Random Forest Regression (RFR) ile yaptıklarından biraz daha iyidir. [34]'te yapılan çalışmada akıllı şehirlerde kamu güvenliği sağlamak için, bir nükleer radyoaktif kaynağın tespitinde, WSN kullanan dağıtılmış bir şema geliştirilmiştir.

Makine öğrenmesi ve derin öğrenme tekniklerini son yıllarda gelişmesiyle beraber park yerlerinin görüntü üzerinden tespit edilmesi yaygın çalışılan alanlar arasına girdi. Buradaki en büyük problem eğitimin gerçekleştirileceği büyük veri setinin olmamasıdır.

[36]'da yapılan çalışmada üç farklı kamera açısından kaydedilen 65.899 görüntüden PKlot olarak adlandırılan veri seti oluşturulmuş ve çalışmalar için paylaşılmıştır. [37] ve [38]'de park yerinin görsel tespiti için derin öğrenme tabanlı CNN modeli önerilmiştir. Önerilen modelin etkinliği göstermek için AlexNet ile karşılaştırılmıştır. [39]'da görüntü üzerinden park yeri tespitinin performansının artırılması için CNN ile beraber SVM kullanılmıştır. CNN ile çıkarılan özellikler SVM ile gruplandırılmıştır. Dış ortamda çalışılan bu çalışmada daha yüksek doğruluk oranı elde edilmiştir. [40]'da yapılan çalışmada park alanları geniş bir ölçekten zoom lensli ve motorlu kameralar ile izlenmektedir. Önerilen derin öğrenme tekniği ile otoparka giren araçlar park yerleri ile beraber takip edilecek plaka numarası ile eşleştirebilmektedir. [41] ve [42]'de de yapılan çalışmalarda park yerinin tespiti için derin öğrenme yöntemlerinden yararlanılmıştır. Park yerinin tespiti için kullanılan görüntüleri uydu ve İHA' kameralarından elde etmişlerdir. [43]'de otoparklardaki görüntü verileri CNN ile anlamlı verilere dönüştürerek bulut tabanlı sunuculara aktarılmıştır. Bu bilgilere erişilebilecek mobil bir uygulama geliştirmiş ve park yerleri takibi yapılmıştır. [14]'te, blok düzeyinde otopark doluluk oranını gerçek zamanlı olarak tahmin etmek için bir Graph-CNN modeli oluşturulmuştur. Modelin parkmetre işlemleri, trafik hızı ve hava koşulları gibi girdiler, heterojen olarak yapılandırılmış çoklu trafik veri kaynakları ile beslenmiştir.

[45]'de, San Francisco şehrinden toplanan otopark verileri üzerinden otomatik kümeleme ve anormallik algılama teknikleri kullanılarak potansiyel olarak ilginç eğilimler ve olaylar tespit edilmeye çalışılmıştır. Bir metropol alanında gerçek zamanlı otopark verileri üzerinde ilk kez kümeleme ve anomali tespiti için bir analiz gerçekleştirilmiştir. Bulgular, arazi kullanımının (ticari veya konut) dış faktörlerinin fiyatlandırılması, güvenlik, emniyet ve diğer ulaşım yöntemlerine uygunluk gibi park etme davranışını etkileyebilecek analizlere odaklanmaktadır. Ayrıca, aşırı duyarlı ölçek kümelemesinin, arızalı bir sensör olasılığı veya günün belirli saatlerinde dış etkenlerden etkilenen park yerleri gibi olası operasyonel sorunları nasıl tanımlayacağı belirtildi.

[46]'da yapılan çalışmada boş park yerlerinin tespiti sensörler aracılığıyla yapılmıştır. Park yerlerinin tahmini yapmak için yapay sinir ağı kullanmışlardır. [47]' de San Francisco, ABD ve Melbourne, Avustralya olmak üzere iki şehirden toplanan park doluluk verilerini analiz edilmiştir. Otopark doluluk oranı için çeşitli tahmin mekanizmaları kullanılmış ayrıca farklı makine öğrenmesi metotlarının görece olarak güçlü yönleri karşılaştırılmıştır. Ek olarak otopark doluluk oranına etki eden faktörler

tespit edilmiştir. [48]' de park alanlarını kısa vadede tahmin etmek için LSTM ve genetik algoritmandan oluşan hibrit bir yaklaşımdan bahsedilmiştir.

Çizelge 2.1. Akıllı Park Sistemleri ile İlgili Başlıca Çalışmalar

Problem	Yöntem	Elde Edilen Başarı
Arama algoritmalarının enerji tüketimi [13]	IoT, BST	50 otopark için %98'in üzerinde enerji tasarrufu.
Park Edilen Yeri Arama [26]	Mobil Telefon	En iyi deneyde, %98,00 doğruluk.
Personel Maliyeti Azaltma [28]	RFID	Personel kaynaklı maliyet ve sorunları ortadan kaldırma.
Boş Park Yeri Tespiti [37]	CNN, AlexNet	En iyi deneyde, %90,13 doğruluk.
Boş Park Yeri Tespiti [38]	mLeNet, mAlexNet	En iyi deneyde, %99,60 doğruluk.
Boş Park Yeri Tespiti [39]	CNN + SVM	En iyi deneyde, %99,70 doğruluk.
Park Yeri izleme ve Tespiti [40]	AlexNet, mAlexNet	En iyi deneyde, %98,12 doğruluk.
Nesne Algılama [41]	FCN	En iyi deneyde, %85,00 doğruluk.
Araç Algılama [42]	Mean-Shift Algorithm + CNN + SVM	En iyi deneyde, %93,60 doğruluk.
Boş park yeri tespiti ve yayınlama [43]	CNN, IoT, Bulut	%8,13 oranında AUC iyileştirmesi.
Akıllı park yeri ve trafik doluluk tespiti [46]	DELM	En iyi deneyde, %91,25 doğruluk.

Çizelge 2.1.(devam) Akıllı Park Sistemleri ile İlgili Başlıca Çalışmalar

Akıllı park yeri ve trafik doluluk tespiti [14]	GCNN, LSTM, FNN	GCNN ile en iyi deneyde, %10,6 MAPE değeri.
Otopark doluluk tespiti [47]	RT, NN, SVR	RT ile en iyi deneyde, 0,032 MAE değeri.
Otopark doluluk oranı tahmini [48]	BP, LSTM	En iyi deneyde, %90,00 doğruluk.

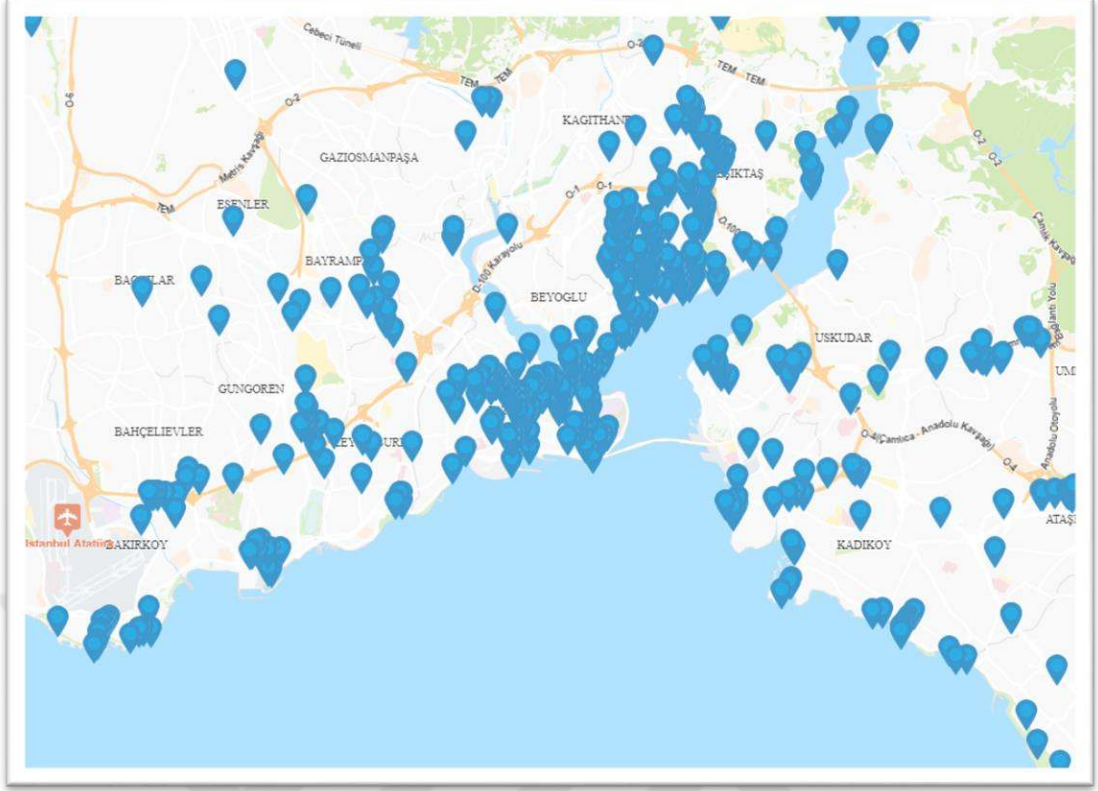
Çizelge 2.1’de akıllı park sistemleri üzerine yapılan çalışmaların ana hatları karşılaştırmalı olarak verilmiştir. Yapılan çalışmalar incelendiğinde park yeri problemi çözümünde IoT cihazlarının kullanılmaya başlanması ve derin öğrenme yöntemleri etkisini göstermektedir. Özellikle boş park yerinin görüntü üzerinden derin öğrenme yöntemleri ile tespitinde önemli çalışmalar mevcuttur.

3. MATERYAL VE YÖNTEM

Bu bölümde ISTPARK veri setinin oluşturulması, derin öğrenme ve makine öğrenmesi yöntemleri kullanılarak otopark doluluk tahmini yapılması, önerilen akıllı park sistemi mimarisi, yapılan deneyler ve elde edilen sonuçlar detaylı olarak anlatılmıştır.

3.1. ISTPARK Veri Seti

Türkiye'nin İstanbul şehri coğrafi konumu, mimari yapısı, yoğun nüfusu gibi özellikleriyle metropol alanıdır. Ayrıca ülkenin ekonomik, kültürel ve tarihi merkezi olarak kabul edilir [8]. Bu kadar kalabalık nüfusa sahip şehirler için akıllı şehir kavramı daha çok önem kazanıyor. İstanbul Büyükşehir Belediyesi, akıllı şehirlere, şirketlere ve araştırmacılara katkı sağlamak amacıyla yaşam, çevre, insan, ekonomi, afet yönetimi, trafik, enerji vb. konularda xml, xlsx, csv, API vb. formatlarda açık veri sağlamaktadır. Bu tez çalışmasında kullanılan otopark bilgi hizmeti ve Meteorolojik Gözlem İstasyonu Verileri, İstanbul Büyükşehir Belediyesi Açık Veri Lisansı ile lisanslanmıştır. Ayrıca 4.0 International (CC BY 4.0) kapsamında lisanslanmış kamu sektörü bilgilerinden oluşmaktadır. Şekil 3.1'de otopark bilgi hizmeti sağlanan otoparkların harita üzerindeki konumları işaretlenmiştir. İç bölgelerde ve sahil kenarlarında otopark yoğunluğu fazlayken diğer bölgelerde yoğunluk daha azdır.



Şekil 3.1. İstanbul'daki Otoparkların Konumları

ISTPARK veri seti, İstanbul Afet Koordinasyon Merkezi tarafından sağlanan hava durumu verileri ve İstanbul Büyükşehir Belediyesi otopark bilgi hizmeti servisi üzerinden alınan 731 otoparkın verisi toplanarak oluşturulmuştur. Her bir otopark için belirli periyotlarda alınan verilerden oluşan veri seti, 19 parametreden oluşmakta ve 4,5 milyon kayıt içermektedir. Veriler bir SQL Server 2014 üzerinde depolanmaktadır. Veri seti ile ilgili detaylı bilgiler Çizelge 3.1'de verilmiştir.

Çizelge 3.1. ISTPARK Veri Seti Parametreleri ve Özellikleri

Parametre	Veri Tipi	Açıklama	Kapsamı
CarparkID	Tam Sayı	Otopark kimlik bilgilerini içerir, Benzersiz bir alandır	Otopark
CarparkName	Metin	Otopark adı bilgilerini içerir	Otopark
Latitude	Ondalık	Otoparkın enlem bilgisi	Otopark
Longitude	Ondalık	Otoparkın boylam bilgisi	Otopark
Capacity	Tam Sayı	Toplam otopark kapasitesi	Otopark
EmptyCapacity	Tam Sayı	Boş otopark kapasitesi	Otopark

Çizelge 3.1.(devam) ISTPARK Veri Seti Parametreleri ve Özellikleri

CarparkType	Metin	Otopark Tipi (Yol üstü, Kapalı, Açık)	Otopark
County	Metin	Otoparkının bulunduğu İlçe	Otopark
DateofUpdate	Tarih-Saat	Bilgi güncelleme tarihi ve saati	Otopark
BusinessHours	Saat Aralık	Otoparkın çalışma saatleri	Otopark
FreeParkingMin	Tam Sayı	Ücretsiz park etme süresi (dakika)	Otopark
MonthlySubscriptionFee	Ondalık	Aylık Abonelik Ücreti	Otopark
Address	String	Otoparkın tam adresi	Otopark
AreaPolygon	Dizi[Ondalık] [Ondalık]	Otoparkın çokgen bilgisi	Otopark
Distance	Ondalık	Otoparka uzaklık	Otopark
Tariff	Liste	Tarife (Metin) & Ücret Bilgileri (Ondalık)	Otopark
Location	Metin	Otopark yeri bilgisidir. Bir veya daha fazla otopark bir yerde olabilir	Otopark
AirTemperature	Ondalık	Gün boyunca ortalama sıcaklık	Hava
Precipitation	Ondalık	Gün boyunca toplam yağış miktarı	Hava

Her otopark için veri güncelleme süresi farklıdır. Bu süre 10 ile 147 dakika arasında değişmektedir. Farklı periyotlara ait zaman serilerini aynı frekansa getirmek için SQL Server üzerinde bir optimizasyon işlemi gerçekleştirilmiştir. Öncelikle veri setinin başlangıç tarihinden bitiş tarihine kadar 10 dakikalık bir frekans zaman serisi oluşturulmuştur. Daha sonra bu zaman serisi her bir otoparkın verileriyle karşılaştırılarak en uygun veri dizisi yeni seriye eklenmektedir. Bu işlemin ardından her bir otopark için 10 dakikalık bir frekans zaman serisi oluşturulur. Şekil 3.2’de veri optimizasyon sürecinin algoritması sunulmuştur.

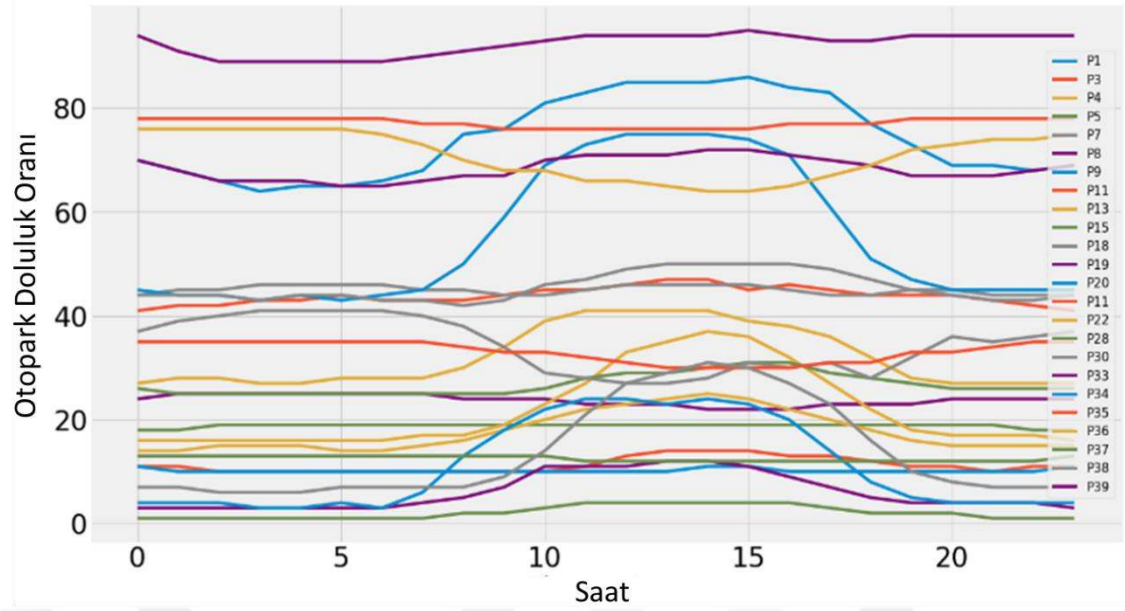
1:	function CreateTimeSeries(<i>s,e,f</i>)
2:	Input: <i>s</i> : Start date, <i>e</i> : End date, <i>f</i> : Time frequency (minute)
3:	Output: <i>T</i> : Time series
4:	while $s \leq e$
5:	$T \leftarrow \text{add}(s)$
6:	$s \leftarrow s + f$
7:	end while
8:	return <i>T</i>

Şekil 3.2 Algoritma 1: Veri Seti Optimizasyonu

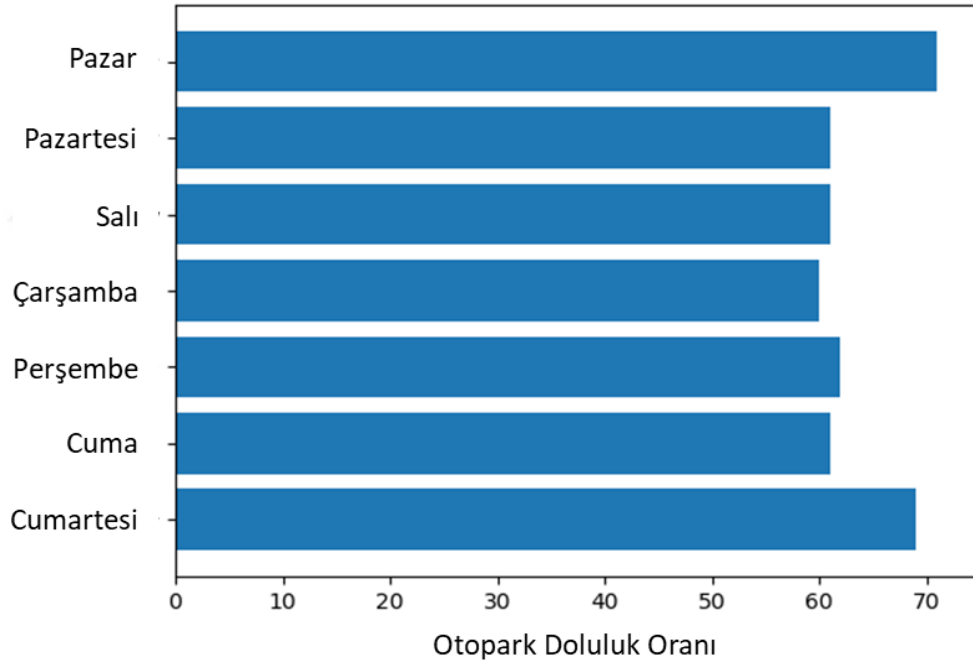
9:	end function
10:	main:
11:	Input: s: Dataset start date, e: Dataset end date, f: Time frequency (10
13:	minute), T: Empty Time series, P: Parking List, SD: Start Dataset
14:	Output: ED: End dataset
15:	$T \leftarrow \text{CreateTimeSeries}(s, e, f)$
16:	for $k=0$ to $T.length-1$ step 1
17:	for $i=0$ to $P.length-1$ step 1
18:	for $j=0$ to $SD.length-1$ step 1
19:	if $SD[j][ParkID] == P[i][ParkID]$ then
20:	if $SD[j][Date] < T[k]$ then
21:	$ED \leftarrow \text{add}(T[k-1], SD[j][value], SD[j][...])$
22:	//First value before this date
23:	end if
24:	end if
25:	end for
26:	end for
27:	end for
28:	return ED

Şekil 3.2.(devam) Algoritma 1: Veri Seti Optimizasyonu

Şekil 3.3'de rastgele seçilen 40 otoparkın gün içindeki ortalama doluluk dağılım grafiği sunulmaktadır. Doluluk oranının 09.00'dan sonra arttığı, 16.00'dan sonra azaldığı açıkça görülmektedir. Bunun nedeni Türkiye'de işyerlerinin ve okulların başlangıç ve bitiş saatlerinin sırasıyla 08.30 ve 17.30 olmasıdır [5]. Ayrıca alışveriş vb. nedenlerle hafta sonu bireysel kullanımlı araçlar daha yüksektir. Şekil 3.4'de bu durum teyit edilmektedir. Otoparklar hafta içi günlerden daha çok hafta sonları kullanılmaktadır.



Şekil 3.3 Otopark Doluluk Dağılım Grafiği



Şekil 3.4 Günlük Ortalama Otopark Doluluk Grafiği

3.2. Derin Öğrenme Tabanlı Otopark Doluluk Tahmini

Büyük metropollerde araç popülasyonunun fazla olmasından ve yeterli sayıda otopark olmamasından dolayı otoparka ihtiyacı olan sürücülerin otoparkta yer aramak için harcadıkları zaman, enerji ve maliyet her geçen gün artmaktadır. Bu durumun aynı

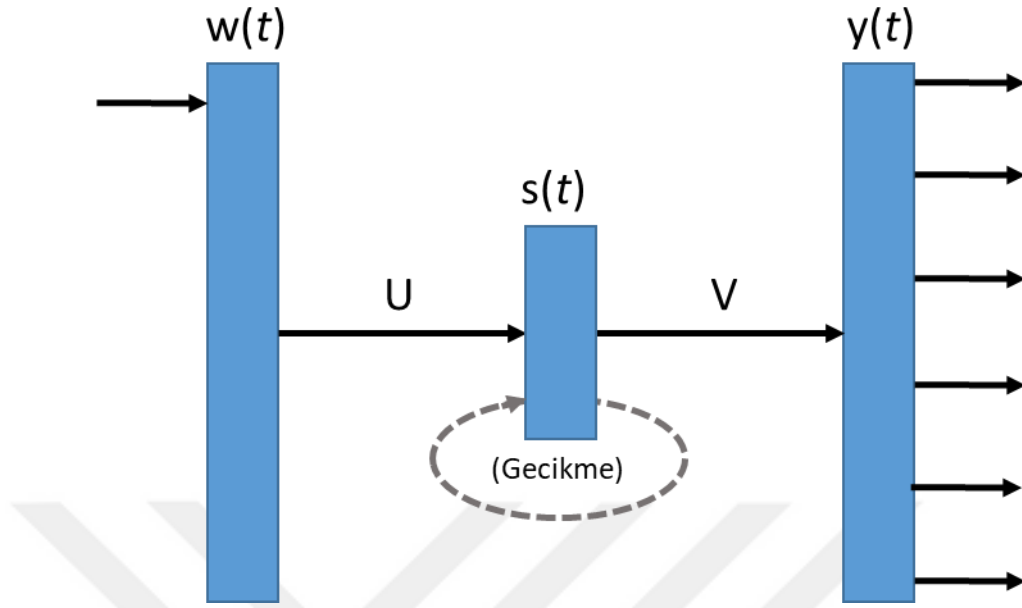
zamanda trafik yoğunluğunu arttırdığı ön görülmektedir. Bu problem için önerilen mimari de dinamik veri toplama işlemi yapılmakta ve derin öğrenme yöntemleri kullanılarak geleceğe dönük tahmin yapılmaktadır. Ayrıca oluşturulan modelin çift yönlü olarak mobil cihazlar ile iletişimi IoT tabanlı olarak yapılmaktadır. Bu sayede kullanıcılar bu mimariyi kullanarak mobil cihazlar üzerinden hangi otoparkın hangi saatte müsait olduğunu görecekler ve akıllı şehir konsepti çerçevesinde planlarını yapabileceklerdir. Aslında bu mimari benzerlik açısından haftalık hava tahminine çok benzemektedir. Çoğu kullanıcılar hava tahminine bakarak haftalık planları oluşturmaktadırlar. Bu mimari sayesinde kullanıcılar kolaylıkla haftalık otopark planlarını yapabileceklerdir. Dolayısıyla zaman, enerji ve maliyet verimliliği üst seviyeye çıkacaktır.

İlk olarak Igor Aizenberg ve meslektaşları tarafından kullanılan "derin öğrenme", makine öğreniminin bir alt alanıdır ve Yapay Sinir Ağlarına (YSA) dayanmaktadır. Derin öğrenmenin YSA'dan farkı, yapısındaki gizli katmanlardır. Ardışık katmanlar, önceki katmanın çıktısını girdi olarak alır ve yapısı verilerin temsilini öğrenmeye dayanır [49] [50] [51]. Derin öğrenme teknikleri çok büyük boyutlardaki verilerde kullanıldığında geleneksel veri işleme yöntemlerine göre daha iyi sonuçlar vermektedir. [49][50] [52][53]. Otopark doluluk tahmini problemlerinin yüksek doğruluk performansı verecek yöntemlere ihtiyacı vardır. Aksi takdirde yeterli güvenilirliğe ulaşmadığı için çok fazla tercih edilmemektedir. Derin öğrenme yöntemleri bu nedenle oluşturan büyük veriler için daha uygun bir seçenektir. Tez çalışmasında önerilen metodolojide derin öğrenme modeli için otoparklardan toplanan verilerden veri seti oluşturulmuştur. Bu veri seti üzerinde RNN, LSTM ve GRU derin öğrenme yöntemleri ile deneyler yapılmış ve sonuçları tartışılmıştır. Öğrenme algoritmaları, Python kütüphanelerinden KERAS Tensorflow'un desteğiyle uygulanmıştır [54].

3.2.1. RNN Derin Öğrenme Mimarisi

Tekrarlayan sinir ağı (RNN), aynı zamanda Elman ağı olarak da adlandırılır [55]. RNN sıralı veya zaman serisi verilerini kullanan bir tür yapay sinir ağıdır. Bu derin öğrenme algoritmaları, dil çevirisi, doğal dil işleme (nlp), konuşma tanıma ve resim yazısı gibi sıralı veya zamansal sorunlar için yaygın olarak kullanılır. RNN'ler öğrenmek için eğitim veri setini kullanır. Mevcut girdi ve çıktıyı etkilemek için önceki girdilerden bilgi aldıkları için "memory" ile ayırt edilirler. Geleneksel derin sinir ağları, girdilerin ve çıktılarının birbirinden bağımsız olduğunu varsayarken, tekrarlayan sinir ağlarının çıktısı, dizi

içindeki önceki ögelere bağlıdır. Şekil 3.5’ de basit bir RNN mimarisi verilmiştir.



Şekil 3.5 RNN Mimarisi

RNN ağı geri yayılım kullanılarak eğitilir. Giriş, gizli ve çıkış katmanlarını içerir. Bu katmanlardaki değerler hesaplamaları denklem 1,2 ve 3’te gösterilmektedir.

$$x(t) = [w(t)^T s(t-1)^T]^T \quad (1)$$

$$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right) \quad (2)$$

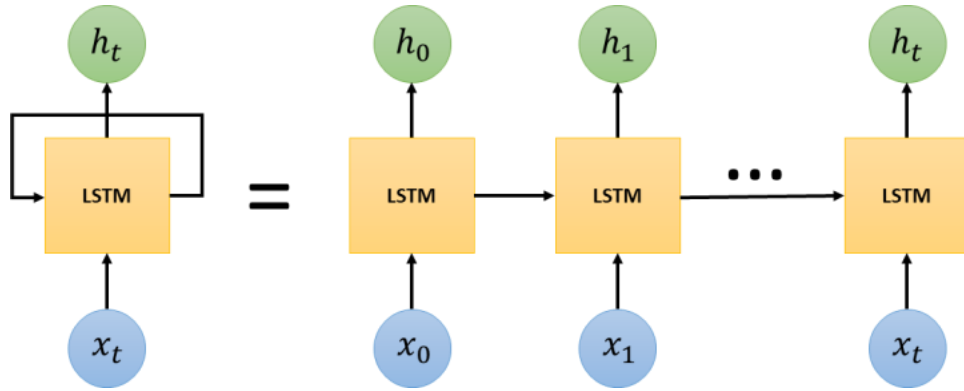
$$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right) \quad (3)$$

$x(t)$ vektörü, N kodlamanın 1’i kullanılırken geçerli kelimeyi temsil eden $w(t)$ vektörünün ve önceki zaman adımından gizli katmandaki çıktı değerlerini temsil eden $s(t-1)$ vektörünün birleştirilmesiyle oluşturulur. $s(t)$ gizli katman ve $y(t)$ çıkış vektörünü temsil eder.

3.2.2. LSTM Derin Öğrenme Mimarisi

Uzun kısa süreli bellek (LSTM) ağı, derin öğrenme alanında kullanılan tekrarlayan sinir ağının (RNN) bir uzantısıdır [56][57]. Standart ileri beslemelerin ağların aksine, LSTM’lerin geri bildirim bağlantıları vardır. Sepp Hochreiter ve Juergen Schmidhuber 1997 yılında RNN’de bulunan kaybolan gradyan problemini çözmek için LSTM’i geliştirdiler [58]. Daha sonra birçok kişinin katkısıyla düzenlenen ve popülerleşen LSTM

şu anda geniş bir kullanım alanına sahiptir [59].



Şekil 3.6 LSTM Tekrarlayan Ağ

LSTM ağları, zaman serisi verilerine dayanarak sınıflandırmak, işlemek ve tahminler yapmak için çok uygundur, çünkü bir zaman serisindeki önemli olaylar arasında bilinmeyen süre gecikmeleri olabilir.

LSTM hücresi giriş, çıkış ve unut kapısı olarak şu üç kapıyı içerir.

- Giriş kapısı, giriş aktivasyonlarının bellek hücresine akışını kontrol eder.
- Çıkış kapısı, hücre aktivasyonunun çıkış akışını kontrol eder.
- Unut kapısı, giriş ve önceki çıktındaki bilgileri filtreler ve hangisinin hatırlanacağına veya unutulup çıkarılacağına karar verir [60].

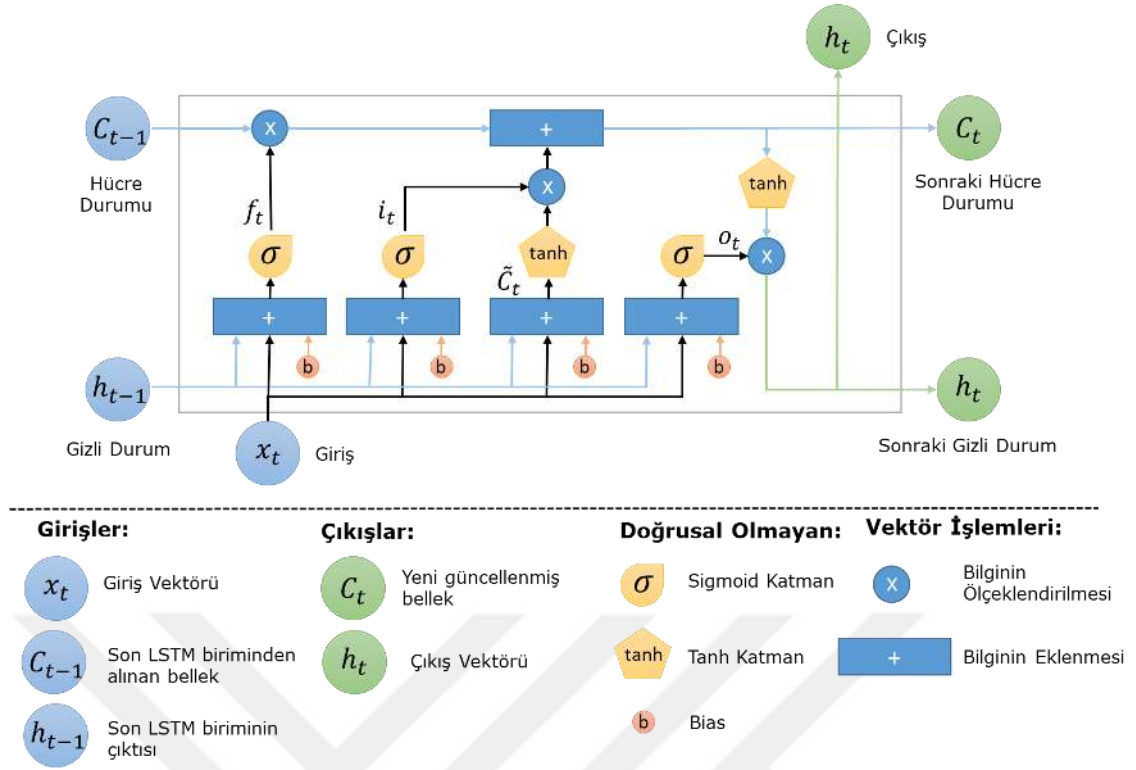
Üç kapının yanı sıra, LSTM hücresi, genellikle hücre durumunun bir parçası olan tanh katmanı olan hücre güncellemesi içerir.

Her bir LSTM hücresinde üç değişken hücreye girer.

- X_t Mevcut giriş,
- h_{t-1} Önceki çıktı,
- C_{t-1} Önceki hücre durumu

Diğer taraftan iki değişken hücreden çıkar.

- h_t Akım çıkışı
- C_t Mevcut hücre durumu



Şekil 3.7 LSTM Mimari

Şekil 3.7’de LSTM’min mimari yapısı verilmiştir. LSTM tekrarlayan ağını uygulamak için önce LSTM hücresi uygulanmalıdır.

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

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, X_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c[h_{t-1}, X_t] + b_c) \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Değişkenler:

- $X_t \in \mathbb{R}^d$: LSTM birimine giriş vektörü
- $f_t \in \mathbb{R}^d$: Unut kapısının aktivasyon vektörü
- $i_t \in \mathbb{R}^h$: Çıkış kapısının aktivasyon vektörü
- $h_t \in \mathbb{R}^h$: LSTM biriminin çıkış vektörü olarak da bilinen gizli durum vektörü
- $\tilde{C}_t \in \mathbb{R}^h$: Hücre girişi aktivasyon vektörü

- $C_t \in \mathbb{R}^h$: Hücre durumu vektörü
- $W \in \mathbb{R}^{h \times d}$: h ve d üst simgelerinin sırasıyla girdi özelliklerinin sayısını ve gizli birimlerin sayısını ifade ettiği eğitim sırasında öğrenilmesi gereken ağırlık matrisleri ve yanlılık vektörü parametreleri.

LSTM hücremizin ilk adımı, hangi bilginin hücre durumundan atılacağına karar vermektir. Bu karar, “kapıyı unut katmanı” f_t olarak adlandırılan bir sigmoid tabaka tarafından yapılır. X_t ve h_{t-1} bakar, C_{t-1} ‘ye hücre durumunu her sayı için 0 ile 1 arasında bir çıkış değeri verir. Çıkış değeri 1 ise, “bunu tamamen tut” temsil ederken, 0 ise “tamamen bundan kurtulun” anlamına gelir.

Bir sonraki adım, hücre durumuna hangi yeni bilgiyi depolayacağımıza karar vermektir. Bu iki bölümden oluşuyor. İlk olarak, “giriş katmanı” olarak adlandırılan bir sigmoid katman hangi değerleri güncelleyeceğimize karar verir. Sonra ki, bir tanh katmanı yeni aday değerler vektörü oluşturur, $\tilde{C}_t$, yeni duruma eklenebilir.

Bir sonraki adımda, durumu güncellemek i_t ve $\tilde{C}_t$ birleştirilir. Bu her bir değeri ne kadar güncellememize karar vermemizi ölçeklendirerek bir aday değeri oluşmasını sağlar. C_{t-1} ve f_t çarpılarak bir önceki kararın unutulması sağlanır.

Son olarak çıkışa ne gönderileceğine karar verilir. Bu çıkış bizim hücre durumumuza bağlıdır. Fakat filtre versiyonu olabilir. İlk olarak, hangi hücre durumunun hangi bölümlerine çıktı çıkartacağına karar veren bir sigmoid katman çalıştırılır. Sigmoidin çıktısı o_t tanh katmanı tarafından hücre durumundan C_t oluşturulan (+1) ile (-1) arasında oluşturan yeni değerle çarpılır.

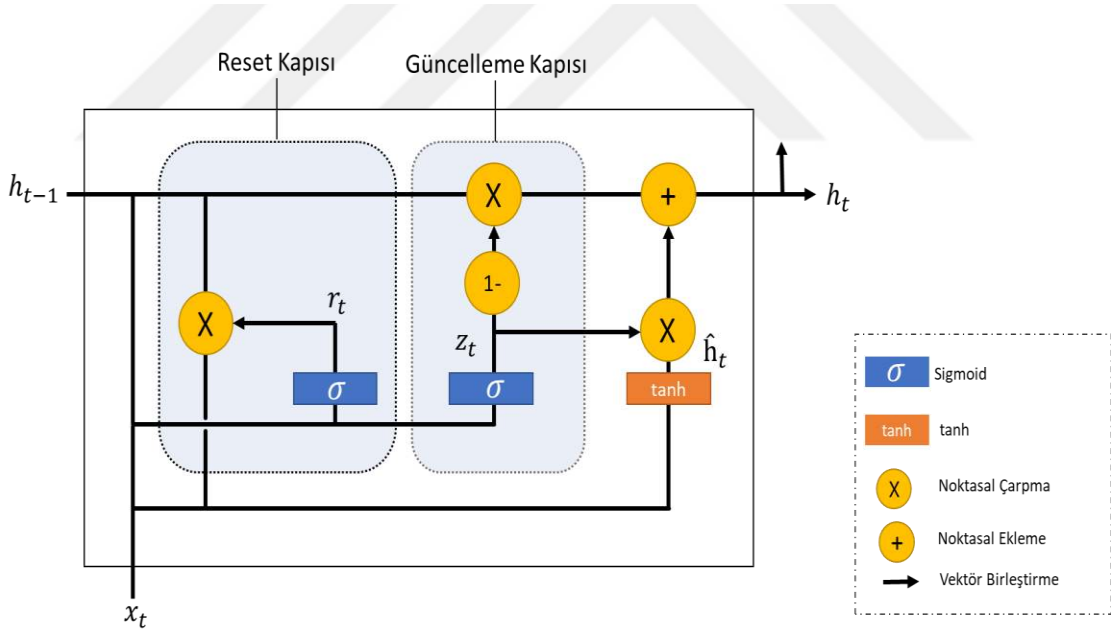
3.2.3. GRU Derin Öğrenme Mimarisi

Kapılı tekrarlayan hücre (GRU) ağırları, son zamanlarda derin öğrenme alanlarının ilgisini çeken bir derin öğrenme yöntemidir [61]. Adından da anlaşılacağı gibi, RNN mimarisinin bir çeşididir ve sinir ağındaki hücreler arasındaki bilgi akışını kontrol etmek ve yönetmek için ağ geçidi mekanizmalarını kullanır.

GRU'lar kısa süreli belleğe bir çözüm olarak oluşturuldu. Bilgi akışını düzenleyebilen kapılar adı verilen iç mekanizmalara sahiptirler [62]. Tekrarlayan Sinir Ağları (RNN), kısa süreli bellek nedeniyle dizi yeterince uzunsa, önceki zaman adımlarından sonraki aşamalara bilgi taşımada yavaşlar. Geri yayılım sırasında, RNN kaybolan gradyan sorunu yaşar. Kaybolan gradyan sorunu, gradyanın zamanda geriye doğru yayıldıkça

küçülmesidir. “Degrades”, bir sinir ağının ağırlıklarını güncellemek için kullanılan değerlerdir. Degrade değeri çok küçükse, öğrenmeye pek katkısı olmaz. Tekrarlayan sinir ağlarında, küçük bir degrade güncellemesi alan katmanlar öğrenmeyi durdurur. Bunlar genellikle önceki katmanlardır. Bu katmanlar öğrenmediğinden, RNN'ler daha uzun dizilerde gördüklerini unutmaya eğilimlidirler, dolayısıyla daha kısa süreli bir hafızaya sahiptirler. Standart bir RNN'nin kaybolan gradyan problemini çözmek için GRU güncelleme ve sıfırlama kapılarını kullanabilir.

GRU, temel bir yapıya sahip olan ve birçok uygulamada yaygın olarak kullanılan LSTM'nin güncellenmiş bir sürümü olarak düşünülebilir [63] [64]. LSTM ağ yapısı ile karşılaştırıldığında, GRU ağı, güncelleme kapısı ve sıfırlama kapısı dâhil olmak üzere uzun menzilli ve gecikmeli serilerin tahmin problemini çözebilen sadece iki kapı yapısına sahiptir [65]. Veri kümelerinin tüm modelleri için sabit sayıda parametre kullanan GRU, CPU yakınsama süresi ve parametre güncellemeleri ve genelleme açısından LSTM birimlerinden daha iyi performans gösterebilir [66]. GRU'lar daha az tensör işlemine sahip olduğundan, eğitim süresi LSTM'lere kıyasla daha hızlıdır.



Şekil 3.8 GRU Mimari

Şekil 3.8’de GRU mimari yapısı verilmiştir. Belirli bir t zaman aralığı için, mini-batch girişi $x_t \in r^{n \times d}$ olarak ifade edilir (n : numune sayısı, d : giriş sayısı) ve son adımın gizli durumu $h_{t-1} \in r^{n \times h}$ olarak ifade edilir (h : gizli durum sayısı). GRU, ünite içindeki bilgi akışını kontrol eden güncelleme ve sıfırlama kapılarına sahiptir. Güncelleme kapısı, hangi bilgilerin atılacağına ve hangi bilgilerin ekleneceğine karar verir. Ayrıca zaman

serilerinde uzun vadeli bağımlılıkları yakalamaya yardımcı olur.

$$z_t = \sigma(W_z h_{t-1} + W_z x_t + b_z) \quad (1)$$

Güncelleme Kapısının hesaplanması denklem (1)'de verilmiştir. Sıfırlama kapısı, geçmişin ne kadarının unutulacağına karar vermek için kullanılan kapıdır. Ayrıca zaman serilerine kısa vadeli bağımlılıkları yakalamaya da yardımcı olur.

$$r_t = \sigma(W_r h_{t-1} + W_r x_t + b_r) \quad (2)$$

Reset Kapısının hesabı denklem (2)'de verilmiştir. Değerin 0'a yakın olması bir önceki ana ait bilgilerin mevcut hafıza içeriğinde unutulduğunu gösterir. 1'e yakın olduğunda mevcut hafıza içeriğinde kalacağını gösterir [56].

$$\hat{h}_t = \tanh(W. [r_t * h_{t-1}] + W x_t) \quad (3)$$

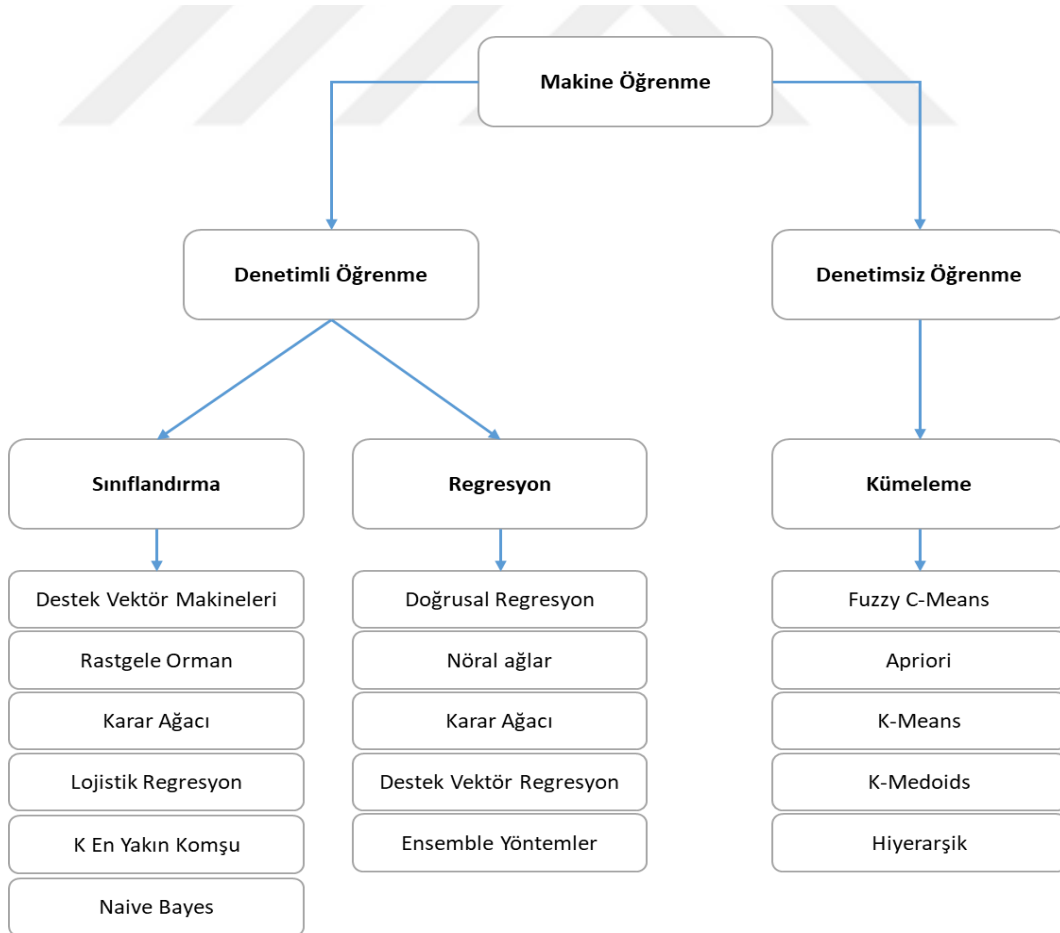
Denklem (3) ile t anındaki hafızanın içeriği belirlenir. Sıfırlama ve güncelleme kapıları aracılığıyla hafızanın ne kadarının unutulup saklanacağı belirlendikten sonra, “tanh” aktivasyon fonksiyonu ile iki kapının bilgisi -1 ile 1 arasında ölçeklendirilir.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \quad (4)$$

Denklem (4) ile gizli katman tarafından t zamanında depolanan bilgiler belirlenir.

3.3. Makine Öğrenme Tabanlı Otopark Doluluk Tahmini

Makine öğrenimi, çevredeki ortamdan öğrenerek insan zekâsını taklit etmek için tasarlanmış, gelişen bir hesaplama algoritması dalıdır. Makine öğrenimine dayalı teknikler, örüntü tanıma, bilgisayarla görme, uzay aracı mühendisliği, finans, eğlence ve hesaplamalı biyolojiden biyomedikal ve tıbbi uygulamalara kadar çeşitli alanlarda başarıyla uygulanmıştır. [67]. Şekil 3.9’da makine öğrenmesinde yaygın olarak kullanılan yöntemlerin hiyerarşik yapısı verilmiştir. Denetimli öğrenme, öngörülemez bir girdi örneği verildiğinde tahmin yapmak için oluşturulmuş bir öğrenme modelidir. Denetimli bir öğrenme modelinin gerçekleştirmesi gereken iki ana görevi vardır: sınıflandırma ve regresyon. Sınıflandırma nominal bir sınıf etiketini tahmin etmekle ilgilenirken, regresyon, sınıf etiketi için sayısal değeri tahmin etmekle ilgilenir. Denetimsiz öğrenmede amaç, belirli kalıpların diğerlerinden daha sık meydana geldiği şekilde girdideki düzenlilikleri bulmak ve genellikle neyin olup neyin olmadığını görmeyi öğrenmektir. Konuşma tanıma, belge kümeleme ve görüntü sıkıştırma örnekleri, denetimsiz öğrenmeyle iyi gider [68].

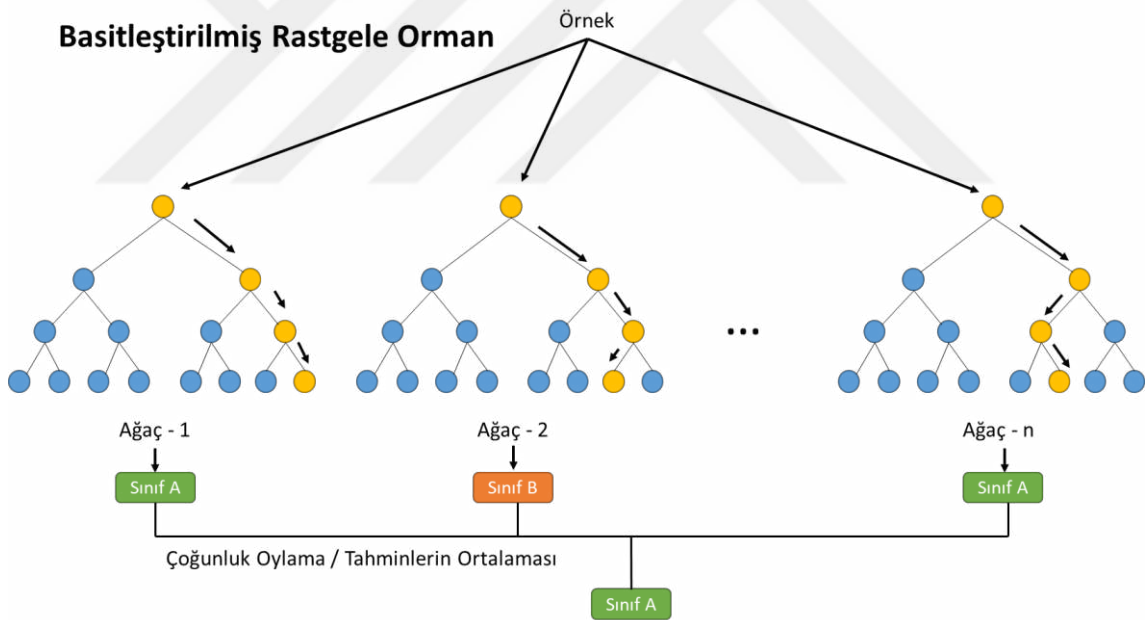


Şekil 3.9 Makine Öğrenmesi Yöntemleri

Yapılan tez çalışmasında ISTPARK veri seti üzerinde zaman serisi problemlerinin tahmininde literatürde başarılı örnekleri olan Rastgele Orman, Destek Vektör Makineleri ve Bütünleşik Otoregresif Hareketli Ortalama yöntemleri kullanılarak otopark doluluk oranı tahmini için simülasyon çalışmaları yapılmış, elde edilen sonuçlar tartışılmıştır.

3.3.1. Rastgele Orman Makine Öğrenme Yöntemi (RF)

2001 yılında L. Breiman tarafından önerilen Rastgele Orman algoritması, genel amaçlı bir sınıflandırma ve regresyon yöntemi olarak son derece başarılı olmuştur [69]. Rastgele orman denetimli bir öğrenme algoritmasıdır. Popüler makine öğrenimi yöntemleri arasında benzersiz bir tahmin doğruluğu ve model yorumlama kombinasyonu sağlar. Algoritmada kullanılan rastgele örnekleme ve topluluk stratejileri, daha iyi genellemelerin yanı sıra doğru tahminler elde etmesini sağlar [70]. Rastgele orman, hesaplama açısından verimli bir tekniktir ve büyük veri kümeleri üzerinde hızlı bir şekilde çalışabilir [71].



Şekil 3.10 Rastgele Orman Algoritması Çalışması

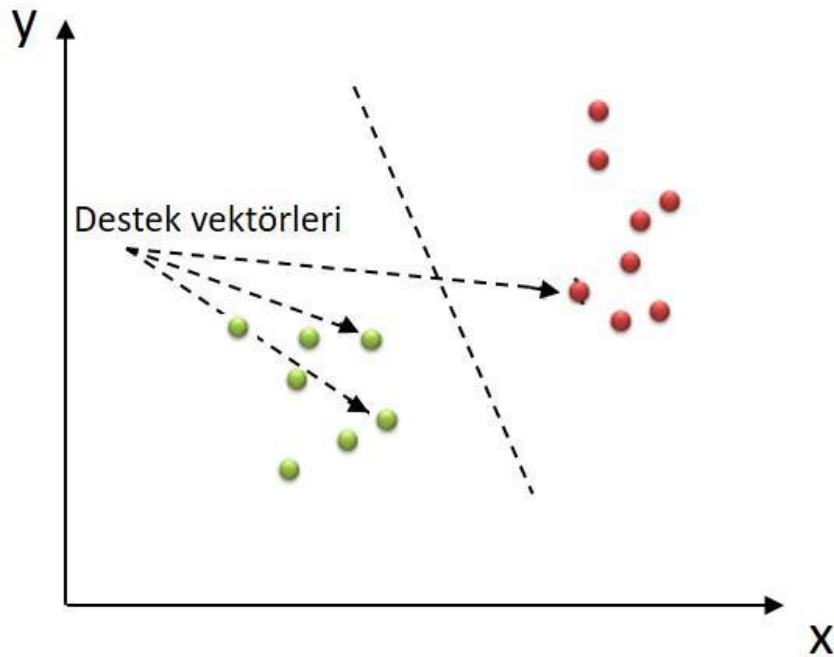
Rastgele orman öğrenme algoritması rastgele olacak şekilde hem veri setinden hem de öznitelik veri setinden birden fazla karar ağacı oluşturur, daha sonra daha doğru tahmin yapabilmek için bu karar ağaçlarını birleştirerek çalışır. Çözölmeye çalışılan problem bir sınıflandırma problemi ise çıktı olan tahminler arasında en çok oyu alanı seçer. Eğer çözölmeye çalışılan problem bir regresyon problemi ise çıktı olan tahminlerin ortalamasını alır. Şekil 3.10'da algoritmanın temel çalışma prensibi verilmiştir. Hem

sınıflandırma hem de regresyon problemlerinin çözümünde kullanılması ve iyi sonuçlar vermesi popülerliğini arttırmaktadır. Rastgele ormanın en büyük dezavantajı aşırı öğrenme veriyi ezberlemedir. Bu dezavantajdan kaçınmak için doğru hiperparametrelerle öğrenme modelinin en uygun şekilde oluşturulması gerekmektedir.

3.3.2. Destek Vektör Makineleri Öğrenme Yöntemi (SVM)

Destek Vektör Makinesi, hala büyük veri sınıflandırma problemlerinin çözülmesine yardımcı olabilecek klasik makine öğrenimi tekniklerinden biridir. Özellikle büyük veri ortamında çok alanlı uygulamalara yardımcı olabilir. Bununla birlikte, destek vektör makinesi matematiksel olarak karmaşık ve hesaplama açısından pahalıdır [72]. Makine öğrenmesinde, tıpkı rastgele orman gibi sınıflandırma ve regresyon analizi için destek vektör makineleri sıkça tercih edilen bir algoritmadır [73]. Denetimli öğrenme prensibine göre çalışan algoritmalar içerisinde yer alır.

Destek vektör makinesi (SVM), nesnelere etiket atamayı örnek olarak öğrenen bir bilgisayar algoritmasıdır [74]. SVM bir düzlem ve hiper düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Şekil 3.11’de SVM sınıflandırıcı ve genel sınıflandırma yapısı verilmiştir.



Şekil 3.11 Destek Vektör Makineleri Sınıflandırma gösterimi

3.3.3. Bütünleşik Otoregresif Hareketli Ortalama Makine Öğrenme Yöntemi (ARIMA)

ARIMA zaman serisi modeli, t zamanındaki değerin geçmiş değerler ile modellenmesidir. AutoRegressive(AR), Integrated(I) ve Moving Average(MA) kısımlarından oluşur. ARIMA, zaman serileri yardımıyla tahmin yapmak için en çok bilinen ve yaygın olarak kullanılan modeller arasında yer almaktadır. Seriyi oluşturan veriler arasında doğrusal bir ilişkinin olduğunu varsayan ve bu doğrusal ilişkiyi modelleyebilen ARIMA modelleri durağan ya da çeşitli istatistiksel yöntemlerle durağan hale getirilen zaman serilerine başarıyla uygulanabilmektedir. [75].

ARIMA Notasyonu:

AR(p): “p” değeri Otokorelasyonun mertebesidir. Geçmiş gözlemlerin ağırlıklı hareketli ortalamasını gösterir.

I(d): “d” değeri Bütünleşme mertebesidir. Doğrusal veya polinomial trendi gösterir.

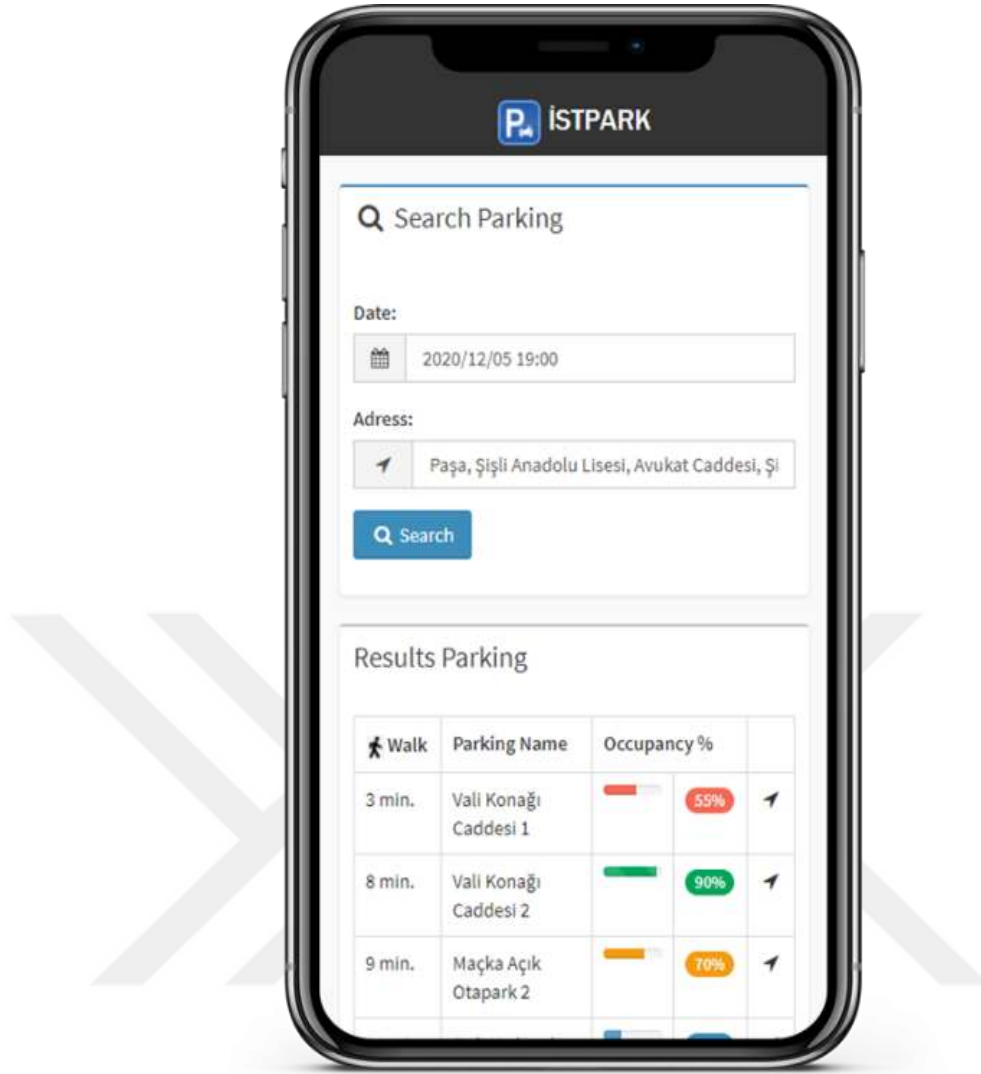
MA(q): “q” değeri Hareketli ortalamanın mertebesidir. Geçmiş hataların ağırlıklı hareketli ortalamasını gösterir.

3.4. Akıllı Park Sistemi Tasarımı

3.4.1. Önerilen Model

Akıllı şehirlerde derin öğrenme yöntemlerinin gelişmesi ile litetaturde daha başarılı akıllı park yeri çözümleri çalışmaları ortaya çıkmaya başlamıştır. Kaynakların giderek daha sınırlı hale gelmesiyle yenilikçi çözümlere ihtiyaç sürekli hale gelmektedir.

Bu tez çalışmasında park yeri arama probleminin çözümünde gerçek zamanlı ve ileriye yönelik tahmin sunan derin öğrenme ve bulut tabanlı yeni bir mimari geliştirilmiştir. Kullanıcıların en uygun otoparkı bulabilmeleri için web tabanlı aynı zamanda mobil uyumlu “ISTPARK” uygulama tasarımı geliştirilmiştir. Bu sayede kullanıcı gitmek istediği yerdeki parkların gün ve saat olarak doluluk oranlarının dinamik olarak tahmin edildiği bir uygulama ile enerji ve vakit tasarrufu yapmış olacaktır. Şekil 3.12’de uygulamanın ara yüz tasarımı gösterilmektedir.



Şekil 3.12 ISTPARK Mobil Tasarımı

Yapılan çalışmada pilot bölge olarak Türkiye'nin İstanbul şehri kullanılmıştır. İstanbul'un pilot bölge olarak seçilmesinde ülkenin en kalabalık nüfusuna ve en yoğun trafiğine sahip olması önemlidir. Her geçen yıl artan nüfusuyla ciddi park yeri problemi bulunmaktadır.

Uygulama ara yüzü kullanıcı dostu olarak tasarlanmıştır. Uygulama kullanılırken sadece tarih ve adres sorularak daha kullanışlı hale getirilmiştir. "Date" değeri, otopark doluluk bilgisini kontrol etmek istediği tarihi ifade etmektedir. "Adress" değeri, gitmek istediği konumun adresidir.

Kullanıcı "Search" butonuna bastığında sistem hedef konuma yakın otoparklar arasından en uygun otoparkı akıllı kriterler sırası ile önerir. Bu akıllı kriterler "Walk" ve "Occupancy"dır. "Walk" otopark ile hedef konum arasındaki yürüyüş süresidir.

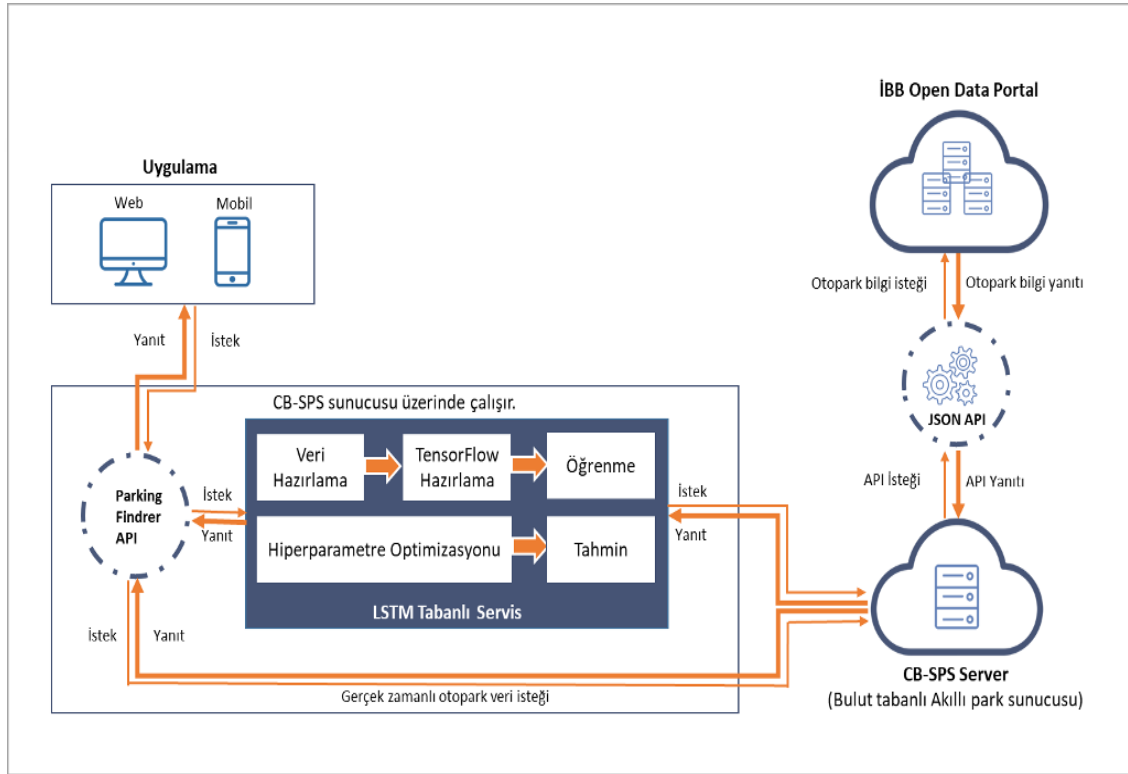
“Occupancy” hedef konumun yakınındaki otoparkların doluluk oranı veya gerçek durumudur. En az bir boş park yeri olan otoparkların listesi listelenir. Liste yürüyüş süresi artan şekilde sıralanır.

Kullanıcı ileriye yönelik bir tarih seçerse bu otoparkların doluluk oranı için derin öğrenme yöntemlerinden “Uzun kısa süreli bellek” (LSTM) modeli kullanılır. LSTM, geleneksel doğrusal yöntemlerin çok değişkenli veya çok girdili tahmin problemlerine uyarlanmasının zor olabileceği zaman serilerinin tahmininde olağanüstü avantaj sağlamaktadır. Birden çok giriş değişkeni olan problemleri neredeyse sorunsuz bir şekilde modelleyebilmektedir.

ISTPAK’tan toplanan 4,5 milyon veri üzerinden 8 farklı özellik kullanılarak LSTM derin öğrenme yöntemi kullanılmıştır. Bu sayede %99,57 oranında kısa ve orta vadede otoparkların gelecekte doluluk oranları tahmin edilmiştir. Gerçek zamanlı veriler otoparklardan toplandıkça ISTPAK veri tabanı dinamik bir şekilde büyümektedir. Derin öğrenme yöntemlerinin büyük veri üzerindeki başarısı dinamik bir şekilde karşılaştırılmıştır. Bu karşılaştırma sonucu en yüksek başarı oranını veren LSTM derin öğrenme yöntemi ile bir model oluşturulmaktadır. LSTM’in dinamik öğrenme modeli her 30 bin veride (yaklaşık haftada bir) güncellenmektedir. Bu sayede veri arttıkça elde edilen sonuçların %100’e daha çok yaklaşacağı ve daha kesin sonuçlar alınacağı ön görülmektedir. Ayrıca veriler arttıkça değişebilecek şartlara daha kolay adapte olacaktır. Buna ek olarak hata oranı sıfıra yaklaşacaktır. Yapılan çalışmada LSTM ile beraber üç farklı makine öğrenmesi algoritması kullanılmış bunlar içerisinde en stabil sonucu LSTM tarafından verildiğini görülmüştür. Verilerin resmi kurumdan direkt olarak alınması ile objektif bir veri seti oluşturulmuş bu sayede oluşturulan veri setinin derin öğrenme yöntemleri ile analiz edilmesi sonucunda akıllı şehir konusuna katkıda bulunulmuştur. Burada elde edilen model ile mobil cihazlar üzerinden JSON formatı ile veriler gerçek zamanlı olarak elde edilmekte ve son kullanıcıya hizmet olarak sunulmaktadır. Bu sayede İstanbul içinden veya dışından gelen kullanıcılar bu mimariyi kullanarak çok rahatlıkla haftalık plan yapabilecekler ve akıllı şehir konseptinden faydalanabileceklerdir. Ayrıca kullanıcılar hem zaman, enerji ve maliyet tasarrufu yapabileceklerdir.

Uygulama mimarisi “Parking Finder Api”, “LSTM Based Service”, “Cloud based – Smart Parking Server (CB-SPS)”, “IBB Open Data Portal (ODP)” üzerine kurulmuştur. “Parking Finder Api” uygulama ara yüzünden gelen istekleri alır ve en uygun otopark listesini geri döndürür. Hedef adresin 2 km yakınındaki bütün otoparklar listelenir. Her

bir otopark için tarihe göre direk olarak CB-SPS (Cloud based – Smart Parking Server) sunucusuna ya da LSTM tabanlı servise istek oluşturulur. Eğer otoparkın doluluk oranını isteği gerçek zamanlı ise CB-SPS sunucusundan her bir otoparkın verileri anlık olarak alınır. Eğer istek tarihi ileriye yönelik ise “LSTM Based Servise” e istek gönderilir. Bu servis ile LSTM derin öğrenme modeli ile otopark doluluk oranını tahmin edip sonucu geri döndürülür. IBB açık veri portalından veri toplanmasını ISTPARK veri seti oluşturma bölümünde detaylı olarak anlatılmıştır. Şekil 3.13’de önerilen yöntemin model tasarımı gösterilmektedir.



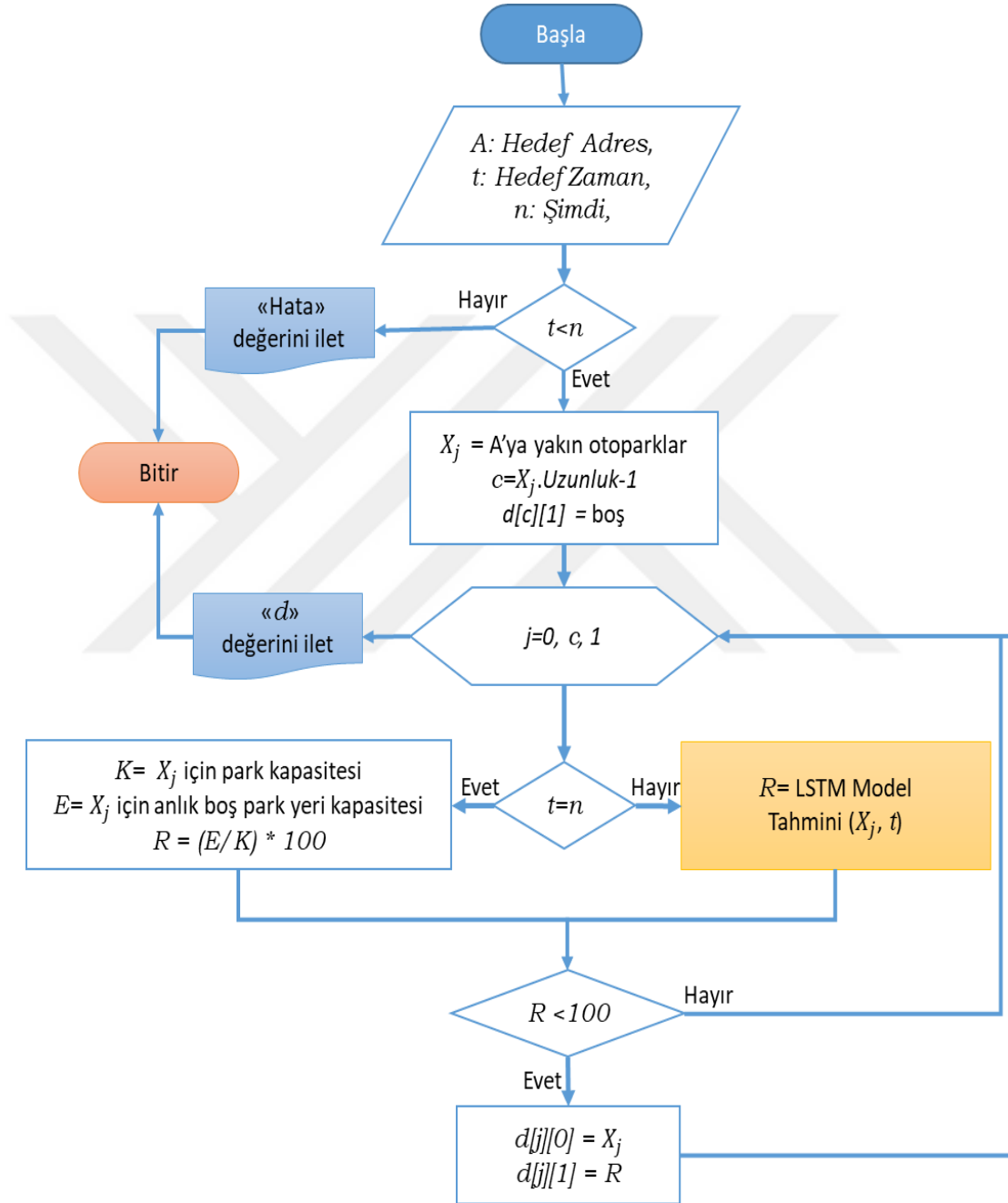
Şekil 3.13 Model Tasarımı

Önerilen yöntemin adımları şu şekildedir;

1. Kullanıcı uygulamaya gideceği adresi ve gitmek istediği tarih girişini yapar. Eğer girdiği tarih geçmiş bir tarih ise hata mesajı alır.
2. Hedef tarih şuna eşit veya ileriye yönelik ise gideceği adresin n metre çapındaki bütün otoparklar araştırılır.
3. Araştırılan her bir otopark için öncelikle hedef tarihe bakılarak işlem yapılır. Eğer hedef tarih şuna eşit ise gerçek zamanlı olarak en az bir boş park yeri olan otoparkların bilgileri alınır. Diğer durumda araştırılan otoparklar LSTM

modeline sokulur ve doluluk oranıyla ilgili tahmin oluşturulur. En az bir boş park yeri olan otopark bilgileri alınır.

4. Alınan otopark bilgileri otopark ile hedef adres arasındaki yürüyüş mesafesi ve doluluk oranına göre kullanıcıya önerilir.



Şekil 3.14 Önerilen Yöntemin Akış Diyagramı

Şekil 3.14’de önerilen yöntemin akış diyagramı verilmiştir. Burada “A” hedef adresi, “t” hedef zamanı ve “n” şu anki zamanı ifade etmektedir. X_j hedef adresin k metre yakınındaki otoparkların bilgileridir. “c”, hedef adrese yakın otoparkların sayısıdır. “K”

her bir otoparkın toplam kapasitesi “E” her bir otoparkın boş kapasitesi ve “R” doluluk oranını ifade etmektedir. “d” hedef adrese yakın en az bir boş park yeri bulunan otoparkların bilgisini tutmaktadır. LSTM Model ilgili otopark ve zaman için tahmin modelidir.

Algoritma 2:	#of Times
function GetParkingCapacity(<i>ParkID</i>) //Connect CB-SPS Server and Run SQL Statement return “select Capacity from tbl_Park where id= ParkID”	- - 1
function GetParkingInstantCapacity (<i>ParkID</i>) //Connect CB-SPS Server and Run SQL Statement return “select top 1 EmptyCapacity from tbl_Park_Detail where id= ParkID order by RecordDate desc ”	- - 1
main: //map: IBB Map API Input: A: Target Adress, t: Target time, n: Current time, key:”Car Park”–keyword, m: 2000 meters, K: Parking Capacity E: Instant empty parking capacity Output: d: List of available car parking A ← ‘Write Target address:’ t ← ‘Write Target time:’ If (t < n) Then X _j = map.Nearby(A, key,m) c=X _j .length() For j= 0 to c-1 R = 0 // Parking Occupancy If (t = n) Then K = GetParkingCapacity(X _j) E = GetParkingInstantCapacity(X _j) R = (E/K) * 100 Else R = LSTMPrediction(X _j) End if	- - - - - - - - 1 1 n-1 1 1 n 1 n-1 1 1 1 n-1 n -

Şekil 3.15 Algoritma 2: Önerilen Yöntem

If ($R < 100$) Then	n-1
$d.Add(X_j, R)$	1
End If	-
End For	-
return d	1
Else	n-1
return “error”	1
End if	-

Şekil 3.15.(devam) Algoritma 2: Önerilen Yöntem

Şekil 3.15’de önerilen yöntemin sözde kodu verilmiştir. Çalışmanın tasarımında IBB Map API’nin önemli rolü bulunmaktadır. Hedef adrese yakın otoparkların listesi bu api üzerinden elde edilmektedir. Daha sonra bulunan her bir otopark için kod çalışmaktadır. “GetParkingCapacity” ve “GetParkingInstantCapacity” fonksiyonları direk olarak CB-SPS üzerinden sorgulama yapmaktadır. “LSTMPrediction” fonksiyonu ilgili otopark için doluluk tahmini yapan ara bir katmandır. Gelen tarih bilgisine göre ilgili otoparkın geçmiş verilerine bakarak otopark doluluk oranını tahmin edip cevap olarak geri döndürmektedir.

LSTM'nin hesaplama karmaşıklığı $O(N_1(H^2 + H N_0))$, burada N_1 girişlerdeki özelliklerin sayısı, H gizli birimlerin sayısı ve N_0 çıkışların sayısıdır [51]. Jupiter Notebook'ta gerçekleştirilen işlemlerin hesaplama karmaşıklığı $O(n)$ olarak hesaplanmıştır.

Bir sonraki bölümde otopark doluluk oranının tahmini, doğruluk performansı ve hata oranı detaylı bir şekilde verildi. Ayrıca zaman serisi tahmininde kullanılan diğer başarılı modellerle karşılaştırılıp değerlendirildi.

3.4.2. Model Eğitimi ve Değerlendirme

Yapılan tez çalışmasında modelleme için bir sinir ağı kütüphanesi olan Keras kullanılmıştır. Derleyici program olarak Jupiter Notebook seçilmiştir. Model deneyleri Intel (R) Core (TM) i7-8650U işlemci (maksimum hız: 2.11 GHz), 16 GB ram ve 620 grafik kartına sahip Intel (R) UHD Graphics'e sahip bir bilgisayarda gerçekleştirilmiştir.

Çizelge 3.2 Model Parametreleri

<i>Parametre</i>	<i>Değer</i>
Eğitim Veri Seti Boyutu	%70, %80, %90
Test Veri Seti Boyutu	%30, %20, %10
Ölçekleme	MinMaxScaler

Çizelge 3.2.(devam) Model Parametreleri

Çalıştırılan Epochs Sayısı	100
Zaman Adımları	10, 20, 30
Aktivasyon Fonksiyonu	tanh
Optimizasyon Fonksiyonu	Adam
Kayıp Metriği	MSE
Değerlendirme Metrikleri	MAE, MSE, RMSE, Accuracy
Dense	1
Dropout	%20
Batch Boyutu	128

Çizelge 3.2’de yapılan çalışmada oluşturulan modellerde kullanılan önemli parametre ve değerleri verilmiştir.

Sayısal girdi değişkenlerinin standart bir aralığa ölçeklendiğinde daha iyi sonuç verdiği makine ve derin öğrenme yöntemlerinde bilinen bir gerçektir. MinMaxScaler yöntemini giriş değişkenlerini ölçeklemek ve normalleştirmek için kullanılan en yaygın fonksiyonlardandır. Özellikle veri dağılımının benzer olduğu uç verilerin olmadığı veri setleri üzerinde daha başarılı sonuçlar vermektedir. Sayısal giriş değişkenlerini 0 ile 1 arasında ölçeklendirilir. Veri setimizdeki girdi değişkenlerinin değerleri benzer olduğu için MinMaxScaler yöntemini tercih ederek verilerimizi 0 ile 1 arasında ölçeklendirdik.

Bir model eğitilirken verilerin tamamı aynı anda eğitime katılması mümkün değildir. Belirli parçalara bölünerek eğitime katılırlar. İlk parça eğitilir, modelin başarısı test edilir, başarı durumuna göre geriye yayılım (“backpropagation”) ile ağırlıklar güncellenir. Daha sonra yeni eğitim kümesi ile model tekrar eğitilip ağırlıklar tekrar güncellenir. Bu işlem her bir eğitim adımında tekrarlanarak model için en uygun ağırlık değerleri hesaplanmaya çalışılır. Bu eğitim adımlarının her birine “epoch” denilmektedir. Makine ve derin öğrenme problemi çözecek en uygun ağırlık değerleri adım adım hesaplandığı için ilk epoch’larda başarımlar düşük olacak, epoch sayısı arttıkça başarımlar da artacaktır. Bununla birlikte belli bir adımdan sonra modelimizin öğrenme durumu oldukça azalacaktır. Ayrıca epoch sayısı arttıkça modelin çalışma süresi saatler hatta günler alabilir. Yapılan çalışmada modellerde kullanılacak epoch değerini bulabilmek için Şekil 3.24’te verilen Epoch – Loss grafiğinin ’den yararlanılmıştır. Loss değeri sabitlendiğinde model artık stabil sonuçlar vermeye başlamıştır.

Aktivasyon fonksiyonu en basit tabirle, ağın verilerdeki karmaşık kalıpları öğrenmesine yardımcı olmak için bir yapay sinir ağına eklenen bir işlevdir. Aktivasyon fonksiyonları modele doğrusal olmama yeteneği katar. Bir önceki hücreden gelen çıkış sinyalini alır ve onu bir sonraki hücreye girdi olarak alınabilecek bir forma dönüştürür. Derin öğrenme yöntemleri modellerinde Tanh, Sigmoid, ReLU, Softmax aktivasyon fonksiyonları sıklıkla kullanılır. Farklı aktivasyon fonksiyonları denenerek yapılan deneylerde en iyi sonucu veren Tanh aktivasyon fonksiyonu önerilen modelde kullanılmıştır.

Dropout sinir ağlarında aşırı öğrenme yani veri setini ezberlemenin önüne geçmek için kullanılır. Dropout değerinin belirlenmesinde ele alınan probleme, veri setine, kurulan modelin yapısına ve elde edilen deneysel sonuçlar etkili olur. Yapılan çalışmada %20 dropout kullanılmıştır. Dropout boyutu arttırılarak yapılan deneylerde model performansının değişmediği, modelin çalışma süresinin uzadığı gözlemlenmiştir.

Batch sayısı bir model eğitilirken aynı anda kaç adet verinin işleneceği anlamına gelir. Batch boyutunun küçük veya büyük olması modelin performansını etkiler. Düşük olduğunda daha fazla optimizasyon işlemi yapılacağı için eğitim süresini uzatır. Büyük olduğunda ise eğitim süresini azaltsa da modelin performansını etkileyebilir. Yapılan çalışmada, 32, 64, 128 ve 256'lık batch boyutu ile deneyler yapılmıştır. En iyi doğruluk oranı 128'lik batch boyutu ile oluşturulan modelde elde edilmiştir. Önerilen modelde, 128'lik batch büyüklüğü ile 100 eğitim dönemi için kurulmuştur. 128 adet veri aynı anda eğitime girer, daha sonra "Adam" optimizasyon fonksiyonu ile MSE hata oranı hesaplanır ve sonraki 120'lik veriye geçilir. Bu işlem tüm veriler işleninceye kadar devam eder, daha sonra bir sonraki epoch adımına geçilerek işlem tekrarlanır.

Modellerin eğitilmesi için oluşturulan veri seti 17 farklı parametreden oluşmaktadır. Ancak her bir otopark için sabit değerleri parametreler model performanslarına etki etmemektedir. Model eğitimi için kullanılan parametre listesi Çizelge 3.3'te verilmiştir.

Çizelge 3.3 Model Eğitiminde Kullanılan Parametreler

<i>Parametre</i>	<i>Açıklama</i>
Datetime	Zaman serisi tarihidir. yyyy-aa-gg SS:dd formatındadır.
Capacity	Otoparkın toplam park yeri sayısıdır.
Empty Capacity	Otoparkın boş park yeri sayısıdır.
Location Density Rate	Aynı bölgedeki otoparkların ortalama doluluk oranıdır.

Çizelge 3.3.(devam) Model Eğitiminde Kullanılan Parametreler

Day/Time	Günün hangi saati olduğunu gösterir.
Week/Day	Haftanın hangi günü olduğunu gösterir.
Public Holiday	İlgili günün resmi tatil olup olmadığını gösterir.
Carpark occupancy	Otoparkın doluluk oranı bilgisidir.

Yapılan çalışmada, oluşturulan modellerin daha hızlı çalışabilmesi için ilk olarak veri seti üzerinde bir normalizasyon işlemi gerçekleştirilmiştir. Sayısal değerlere çevrilen veriler 0 ile 1 arasında ölçeklendirildi. Modeli eğitmek ve tahmin etmek için ölçeklenen veriler eğitim ve test veri setlerine ayrıldı. Daha sonra eğitim ve test verileri kaydırılarak 10 zaman adımı ve 1 çıktı olarak gruplandırılmıştır. Yürütülen çalışmada oluşturulan modeller, eğitim sayısı - test oranları ve zaman adımlarına göre belirlenmiştir.

Eğitilmiş bir sinir ağının (NN) başarısını ölçerken, tahmin edilen sonuçlar gerçek sonuçlarla karşılaştırılmalıdır [76]. Makine öğrenmesi veya derin öğrenme modelleri kullanılarak oluşturulan tahmin modellerinin başarısını değerlendiren metrikler vardır. Bu çalışmada Root Mean Square Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE) ve Accuracy metrikleri ile modellerin başarısı değerlendirilip en iyi sonuç veren model belirlenmiştir. RMSE metriği model tahminin sapma derecesi için kullanılır. 0 ile $+\infty$ arasında değer alır. 0'a ne kadar yakınsa model tahminin sapma derecesi o kadar düşüktür [77]. Denklem 7'de RMSE metriği ifade edilmektedir.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \quad (7)$$

MSE, bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu söyler. Genellikle regresyon problemlerinde, yani sürekli değerleri tahmin etmede kullanılır [78]. Denklem 8'de MSE metriği ifade edilmektedir.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (8)$$

MAE, bir dizi tahmindeki hataların ortalama büyüklüğünü ölçer. Doğru tahminler için MAE değeri düşüktür [79]. Denklem 9'da MAE metriği ifade edilmektedir.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (9)$$

Doğruluk (Accuracy), eğitim kümesi kullanılarak elde edilen modelin ne oranda test

kümesindeki verileri doğru sınıflandırdığını ifade etmektedir. Denklem 10’da doğruluk metriği ifade edilmektedir.

- DP (Doğru Pozitif) değeri, belirli bir sınıflandırıcı ile pozitif sınıfa ait olan verilerden kaçının doğru sınıflandırıldığını ifade etmektedir.
- DN (Doğru Negatif) değeri, belirli bir sınıflandırıcı ile negatif sınıfa ait olan verilerden kaçının doğru sınıflandırıldığını ifade etmektedir.
- YN (Yanlış Negatif) değeri, gerçekte pozitif sınıfta yer alan verinin belirli bir sınıflandırma sonucunda negatif sınıf olarak belirtilmesidir.
- YP (Yanlış Pozitif) değeri, gerçekte negatif sınıfta yer alan verinin belirli bir sınıflandırma sonucunda pozitif sınıf olarak belirtilmesidir.

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YN + YP} \quad (10)$$

3.4.2.1. LSTM, RF, SVM ve ARIMA Modellerinin Değerlendirilmesi

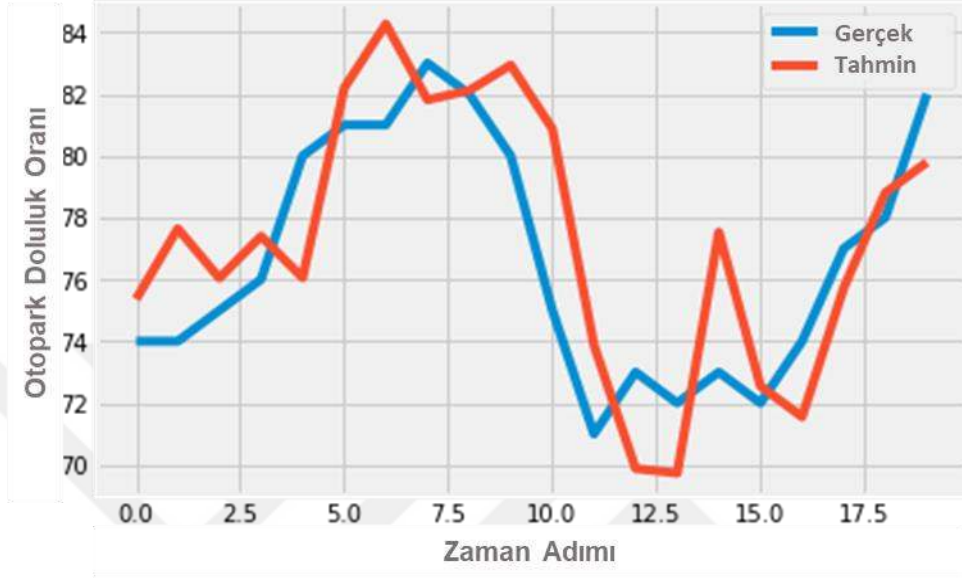
Bu bölümde önerilen LSTM modelinin etkinliğinin ve performansının gösterilebilmesi için RF, SVM ve ARIMA yöntemlerinin iki farklı simülasyondaki başarısı ile karşılaştırılmıştır. Karşılaştırma modeli olarak RF ve SVM’nin seçilmesinde yaygın kullanılan makine öğrenmesi yöntemi olması, ARIMA’nın seçilmesinde zaman serilerinin analizinde başarılı performans göstermesi önemli bir etkidir.

Birinci simülasyonda ISTPARK veri kümesi, LSTM, SVM, RF ve ARIMA makine öğrenme yöntemlerinin performansını karşılaştırmak için farklı özelliklerin kombinasyonları için ayrı ayrı 100 epoch değeri kadar koşturulmuştur. Kullanılan özelliklere göre değişen doğruluk oranları Çizelge 3.4’de sunulmuştur.

İkinci simülasyonda, LSTM, SVM, RF ve ARIMA modellerinin kalitesinin daha doğru ölçülebilmesi için her modelin eğitimi ve testi üç farklı senaryoda gerçekleştirilmiştir. Birinci senaryoda her model veri setinin %70’i ile eğitilmiş, %30’u test ile test edilmiştir. İkinci senaryoda modeller veri setinin %80’i ile eğitilmiş, %20’si ile test edilmiştir. Son senaryoda eğitim için veri setinin %90’ı test için %10 kullanılmıştır. Tüm modellerde 10 epoch değeri kadar koşulmuştur. Çizelge 3.5’te bu üç senaryoda elde edilen sonuçlar sunulmuştur. Modellerin performansları MAE, MSE ve RMSE kayıp metrikleri ile değerlendirilmiştir.

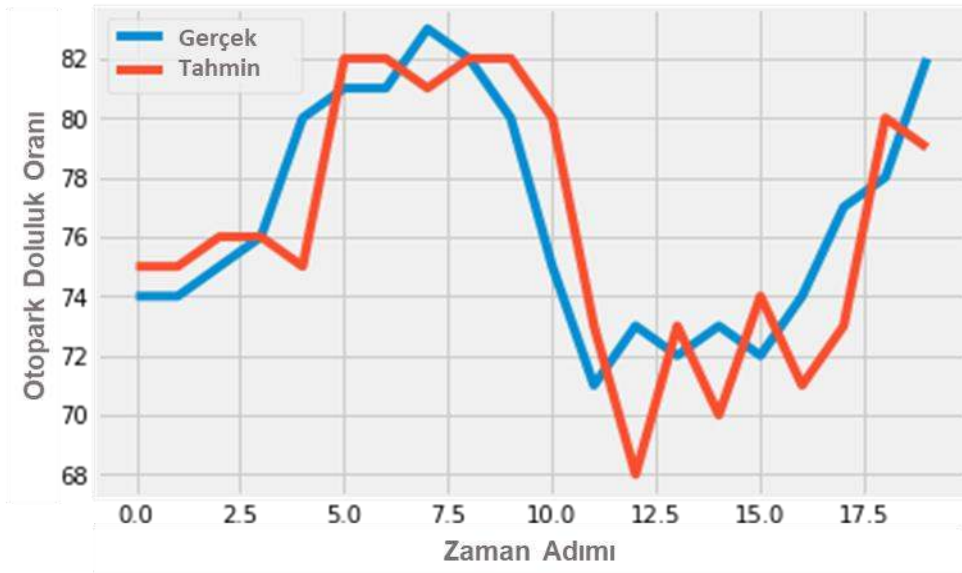
Şekil 3.16’da Rastgele orman modelimizin otopark doluluk oranı tahmin performansı

gösterilmektedir. Mavi seri gerçek değerleri kırmızı seri tahmin edilen değerleri ifade etmektedir. Rastgele orman doğrusal hareket eden seride iyi tahminler yaparken, dönüşlerde ve maksimum ve minimum noktalarında daha kötü tahmin yaptığı gözükmemektedir.



Şekil 3.16 Rastgele Orman Otopark Doluluk Tahmini

Şekil 3.17’de Destek Vektör Makinesi modelimizin otopark doluluk oranı tahmin performansı gösterilmektedir. SVM modelinin de Rastgele orman gibi doğrusal hareket eden seride iyi tahminler yaparken, dönüşlerde ve maksimum ve minimum noktalarında daha kötü tahmin yaptığı gözükmemektedir.



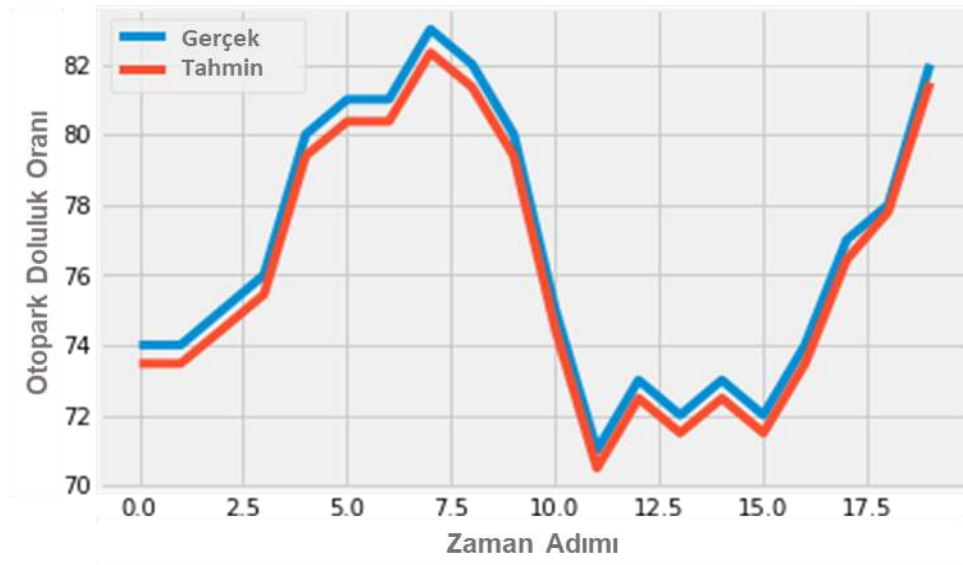
Şekil 3.17 Destek Vektör Makinesi Otopark Doluluk Tahmini

Şekil 3.18’de ARIMA modelimizin otopark doluluk oranı tahmin performansı gösterilmektedir. ARIMA zaten zaman serilerinin analizinde sıklıkla kullanılan bir yöntemdir. Bu çalışmada da ISTPARK veri seti üzerinde yapılan simülasyonlarda SVM ve RF modelinden göre daha iyi tahmin sonuçları vermiştir. Ancak LSTM modelinin çok gerisinde olduğu gözükmemektedir.



Şekil 3.18 ARIMA Otopark Doluluk Tahmini

Şekil 3.19’da oluşturduğumuz Uzun kısa süreli bellek derin öğrenme modelimizin otopark doluluk oranı tahmin performansı gösterilmektedir. Şekilde de görüldüğü gibi LSTM modeli otoparkın doluluk oranını tahmin etmede çok iyi sonuç vermiştir.



Şekil 3.19 LSTM Otopark Doluluk Tahmini

Modellerin tahmin performansları şekiller üzerinde incelendiğinde önerdiğimiz mimaride otopark doluluk oranı tahmini için kullandığımız LSTM’ modeli çok daha iyi sonuç verdiği gözükmemektedir.

Yapılan tez çalışmasında LSTM, RF, SVM ve ARIMA modelleri ISTPARK veri setinin farklı özelliklerinin kombinasyonu ile 100 epochs koşuldu. Elde edilen sonuçlar Çizelge 3.4’de sunuldu. Çizelge 3.4’de verilen “capacity”, otoparkın toplam park yeri kapasitesi ifade eder. “Density”, aynı bölgedeki otoparkların yoğunlukların ortalamasını ifade eder. “Time”, veri setindeki saati ifade eder. “Day”, veri setindeki günü ifade eder, “Holiday” ilgili gün resmi tatil günü ise 1 değil ise 0 değerinin alır. Her kombinasyon için yapılan deneyde LSTM diğer yöntemlerden daha iyi sonuç vermiştir.

Çizelge 3.4’de görülebileceği gibi, en yüksek doğruluk (%99,57) “Capacity”, “Density”, “Time”, “Day”, “Holiday” kombinasyonu kullanılarak elde edilmiştir. “Time” ve “Day” özelliklerinin tahmin performansında en önemli iki parametre olduğu gözükmemektedir. Tekli kombinasyonlarda “Density” özelliği ile en düşük doğruluk oranı alındığı gözlemlenmiştir. Burada en yüksek değer temel alınarak kullanılacak özellikler belirlenmiştir. Ancak çizelge 3.4’de görüldüğü gibi özellik sayısı ile doğruluk oranı azalmış olsa bile elde edilen tahminler, literatürde yapılan çalışmalar göz önüne alındığında kabul edilebilecek düzeydedir. Bu şekilde değişik kısıtlar göz önünde bulundurularak kullanıcıya özellik sayısının azaltılması olanağını da sunmaktadır.

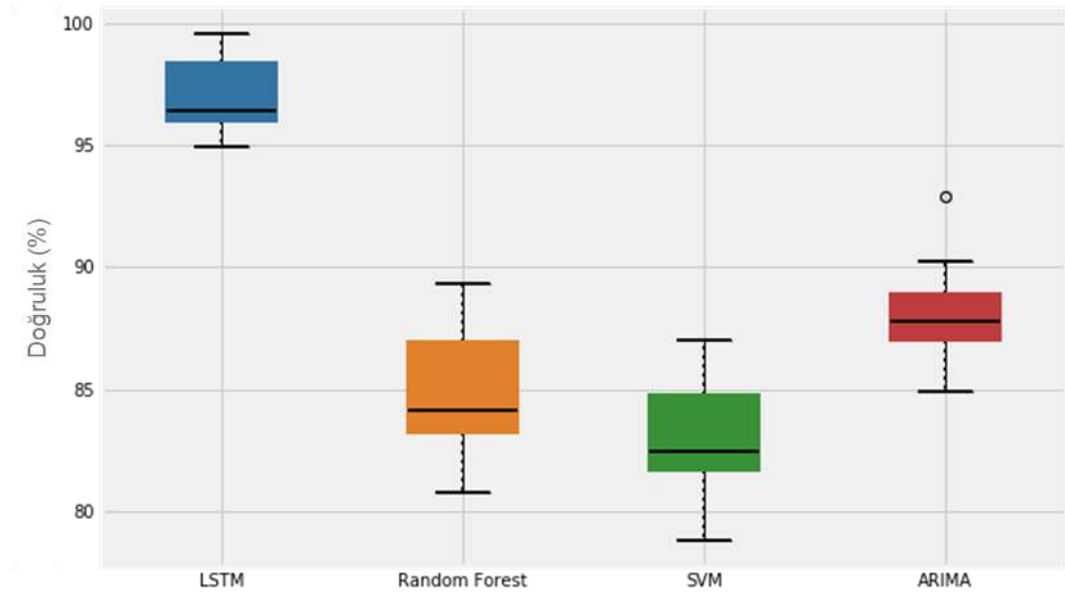
Çizelge 3.4 Farklı Özellikler için Performans Karşılaştırması

Özellikler	Doğruluk (%)			
	LSTM	RF	SVM	ARIMA
Capacity, Density, Time, Day, Holiday	99,57	89,34	87,01	90,22
Capacity, Density, Time, Holiday	96,91	83,22	81,95	88,70
Capacity, Density, Day, Holiday	96,06	82,90	81,47	88,81
Capacity, Time, Day, Holiday	98,90	87,43	86,60	90,12
Density, Time, Day, Holiday	99,02	88,91	86,90	89,01
Capacity, Density, Time, Day	98,70	87,01	86,30	88,80
Time, Day, Holiday	99,23	89,12	86,94	89,45
Capacity, Day, Holiday	96,01	82,77	80,98	87,12
Capacity, Density, Holiday	95,42	83,76	81,67	86,59
Capacity, Time, Holiday	95,01	81,18	79,01	85,90

Çizelge 3.4.(devam) Farklı Özellikler için Performans Karşılaştırması

Capacity, Density, Time	94,91	80,79	78,77	86,34
Density, Time, Day	98,91	87,40	86,70	92,90
Capacity, Time, Day	97,83	86,92	83,12	89,15
Day, Holiday	96,01	83,18	79,92	87,80
Capacity, Holiday	95,15	83,44	83,12	87,06
Capacity, Density	96,05	83,10	81,70	84,93
Density, Time	96,43	83,45	82,44	86,77
Time, Day	98,01	85,50	82,34	88,70
Time	97,67	84,90	82,01	88,12
Day	97,23	84,48	82,40	87,45
Holiday	96,12	84,12	83,25	86,90
Capacity	96,44	84,55	82,85	87,65
Density	95,90	83,82	83,01	87,20

Şekil 3.20’de karşılaştırılan modellerin “Doğruluk” değerlerinin kutu grafiği verilmiştir. LSTM modelinin diğer modellere göre daha stabil çalıştığı grafik üzerinden gözükmektedir.



Şekil 3.20 Doğruluk Kutu Grafiği

Çizelge 3.5’te modellerin hata değerlendirilmesi gösterilmektedir. Farklı eğitim - test veri setine ayrılmış üç farklı senaryoda da en iyi sonuç LSTM modelinden alınmıştır. En iyi

performans olan modelde 0,92 MAE, 2,54 MSE, 1,59 RMSE değerleri alınmıştır. Bu sonuçlar aynı zamanda modelin otopark doluluğunu doğru bir şekilde tahmin edebildiğini göstermektedir.

Çizelge 3.5 Modellerin Hata Değerlendirmesi

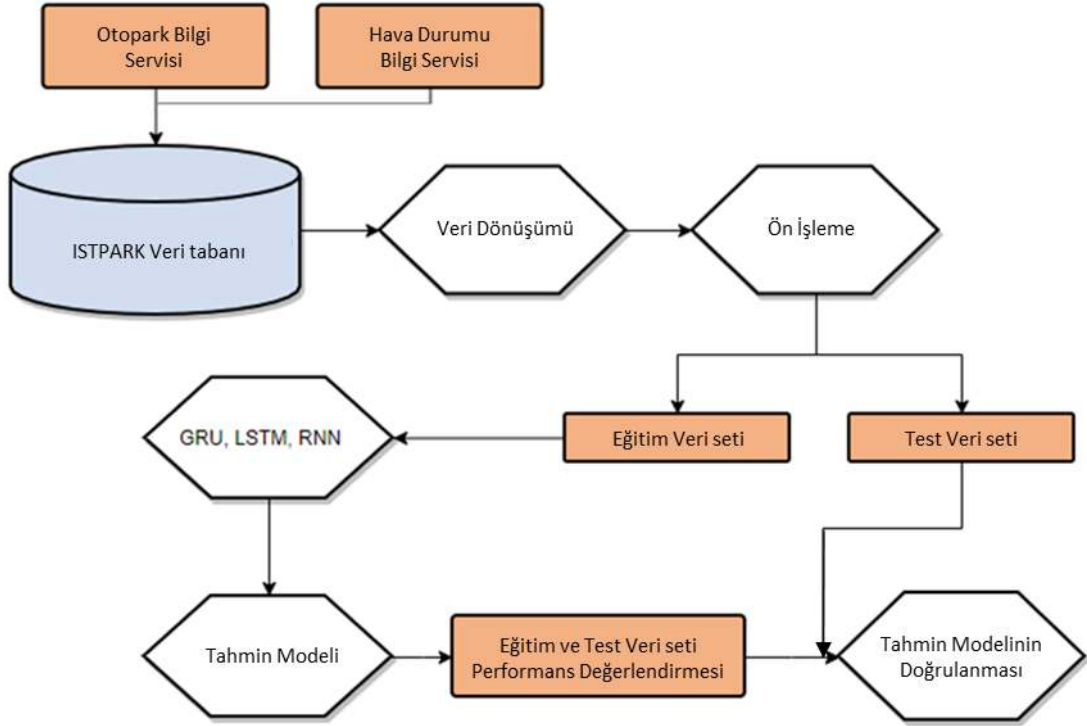
<i>Eğitim - Test</i>	<i>Model</i>	<i>MAE</i>	<i>MSE</i>	<i>RMSE</i>
%70 - 30	LSTM	0,92	2,54	1,59
	SVM	2,50	8,50	2,91
	RF	2,47	7,12	2,66
	ARIMA	1,87	7,29	2,70
%80 - 20	LSTM	1,02	3,12	1,77
	SVM	3,50	18,01	4,24
	RF	2,17	6,96	2,63
	ARIMA	1,76	5,08	2,25
%90 - 10	LSTM	1,11	3,18	1,78
	SVM	2,64	9,11	3,02
	RF	1,83	4,55	2,13
	ARIMA	1,43	4,06	2,01

3.4.2.2. GRU, LSTM ve RNN Modellerinin Değerlendirilmesi

Bir sinir ağı eğitilmeden önce, verilerin bir dizi önceki değerden öğrenebilecekleri şekilde modellenmesi gerekir. Denetimli öğrenme yaklaşımı olarak GRU, LSTM ve RNN, öğrenmek için hem özellikleri hem de etiketleri gerektirir [44].

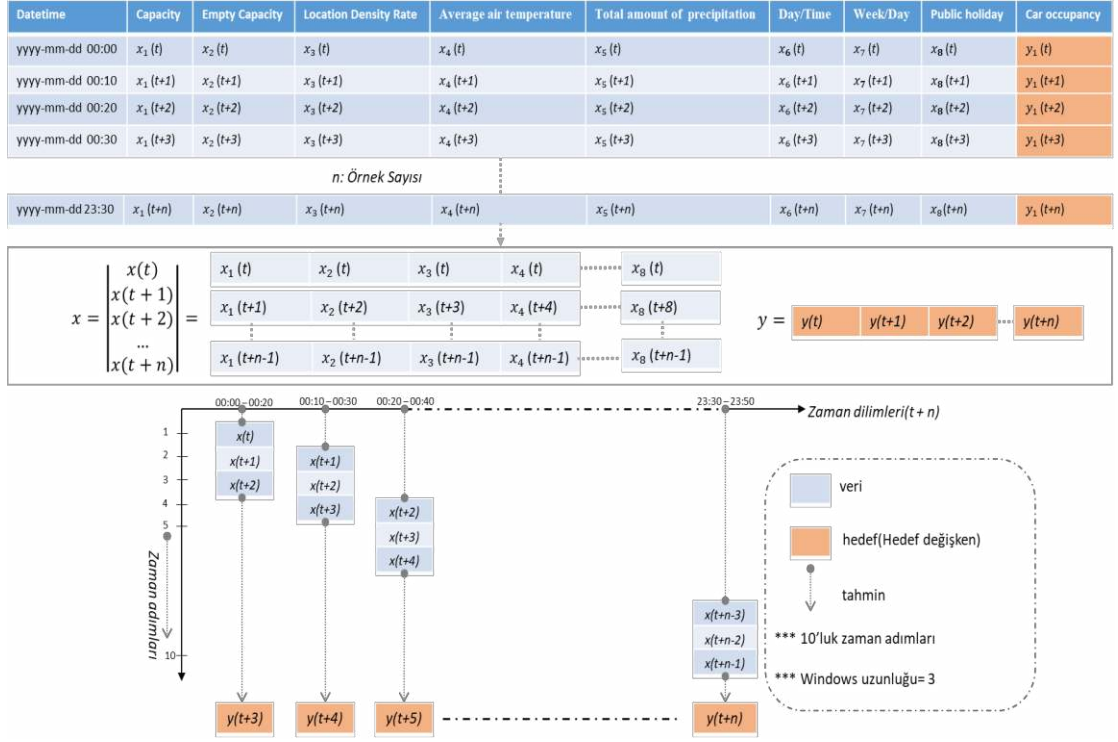
Tekrarlayan Sinir Ağları ve türevleri, çok girişli değişkenlerle ilgili sorunları neredeyse sorunsuz bir şekilde modelleyebilir. Bu, geleneksel doğrusal yöntemlerin çok değişkenli veya çok girdili tahmin problemlerine uyarlanmasının zor olabileceği zaman serilerinin tahmininde olağanüstü bir avantaj olarak gelir. Bu bölümde ISTPARK veri seti ile AKOM'dan alınan hava durumu verileri üzerinde GRU, LSTM ve RNN derin öğrenme modellerinin uygulanmasıyla yapılan deneyler ve elde edilen sonuçlar detaylı olarak incelenmiştir.

Şekil 3.21'de otopark doluluk oranı tahmininde kullanılan veri setinin oluşumu (ISTPARK + AKOM) ve üzerine uygulanan derin öğrenme modelinin işlem adımları gösterilmektedir.



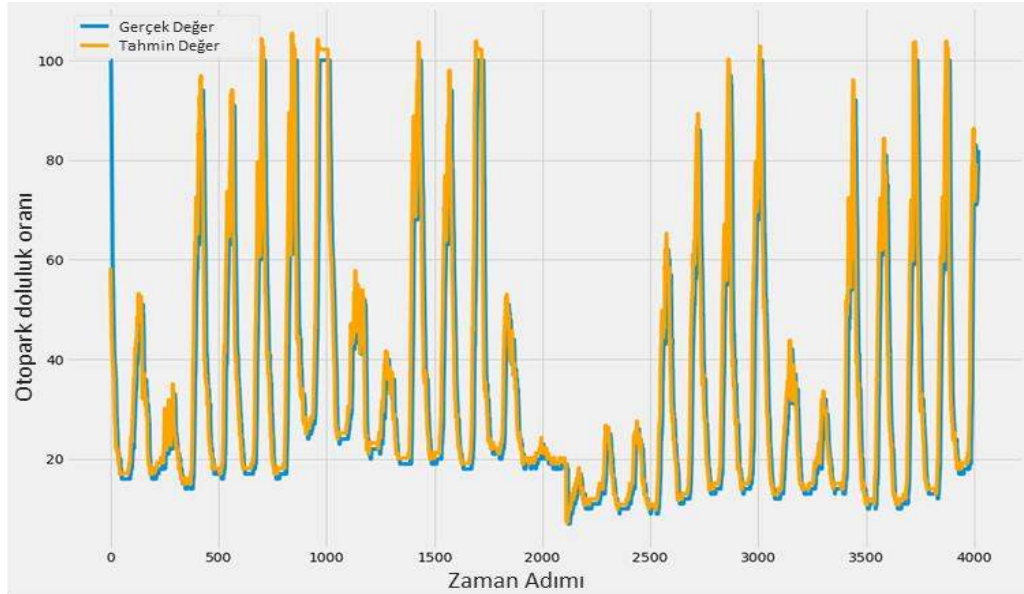
Şekil 3.21 Derin Öğrenme Yoluyla Otopark Doluluk Tahmininin Akış Şeması

GRU, LSTM ve RNN, veri girişlerinin üç boyutlu olarak yapılmasını bekler. Bu nedenle [örnekler, zaman adımları, özellikler] şeklinde gösterilen bir formatta olması gerekir. Öncelikle zaman serisi verileri iki boyuta dönüştürülür ve x'de toplanır. Burada girdi verileri, kümelerde istenen sayıda çakışan gecikmeli değerlerden oluşur. Her bir girdi özneteliği grubu için, tahmin edilmeye çalışılan etiketlerden veya gelecek değerlerden oluşan bir boyutlu bir y dizisi oluşturulur. Ağın etiketlerin önceki değerlerinden öğrenebilmesi için girdi verilerinin ayrıca gecikmeli 'y' değerleri içermesi gerekir. Yapılan çalışmanın ikinci kısmında veri seti, üç farklı derin öğrenme modeli için eğitim ve test verisi olarak sırasıyla %70-30, %80-20, %90-10'luk setlere ayrılmıştır. Her bir eğitim ve test verisi 10, 20 ve 30 zaman adımına bölünerek toplam 27 farklı model oluşturulmuştur. Şekil 3.22'de çok değişkenli zaman serisi veri seti için derin öğrenme modellerinde kullanılacak veri modeli detaylı olarak gösterilmektedir. Zaman serileri söz konusu olduğunda, çapraz doğrulama gibi klasik validasyon yöntemleri önemsizdir. Rastgele örnekler seçip bunları ne test setine ne de eğitim setine atayamayız çünkü geçmişteki değerleri tahmin etmek için gelecekte gelen değerleri kullanmanın bir anlamı yoktur. Basit bir ifadeyle, modelimizi eğitirken geleceğe bakmaktan kaçınmak isteriz. Gözlemler arasında zamansal bir bağımlılık vardır ve test sırasında bu ilişkiyi korumalıyız.



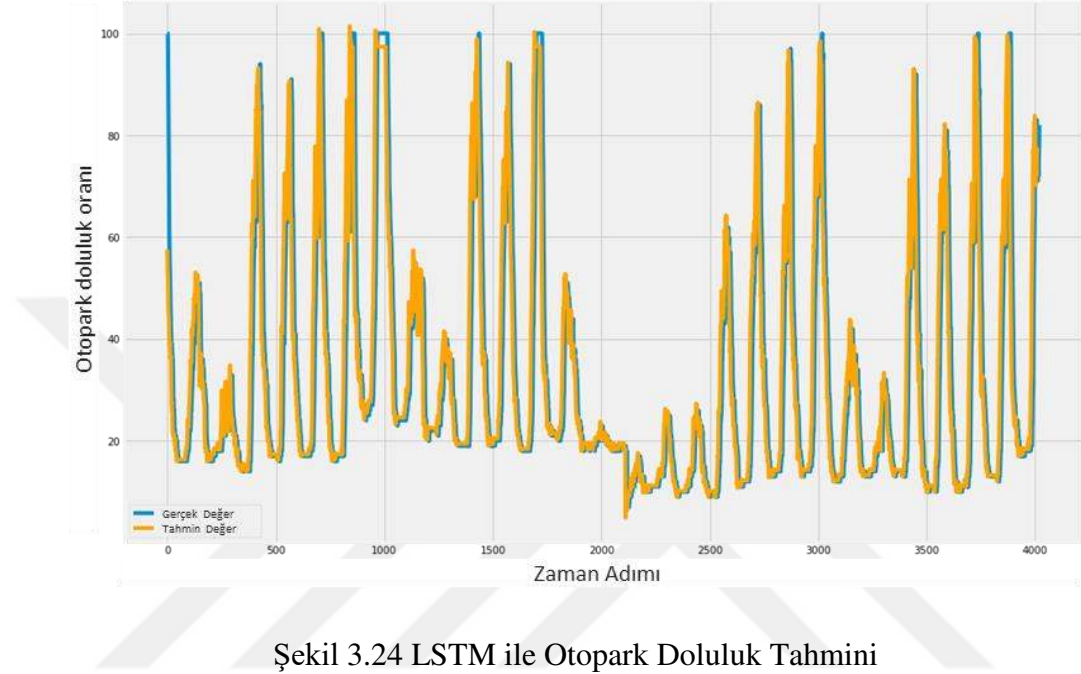
Şekil 3.22 LSTM, GRU, RNN Veri Modeli

Şekil 3.23'de RNN Modelinde bir otoparkın doluluk oranı ve testi için veri setlerinin gerçek ve tahmin edilen değerleri arasındaki ilişkiyi gösteren grafik. Mavi ile gösterilen seriler gerçek verilerin değerlerini gösterirken, test veri setinin tahmini sarı ile gösterilmiştir. RNN modeli, otoparkta tahmin yaparken kötü performans göstermedi.

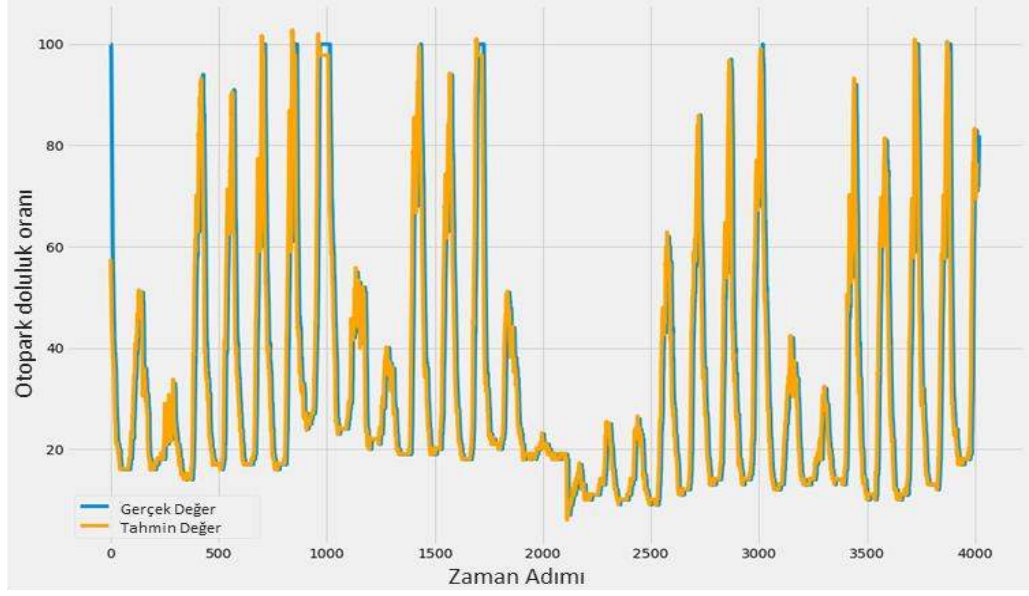


Şekil 3.23 RNN ile Otopark Doluluk Tahmini

Şekil 3.24'de, doluluk oranı veri setlerinin gerçek ve tahmin edilen değerleri ile LSTM ile oluşturulmuş bir modelin testi arasındaki ilişkiyi gösteren bir grafik yer almaktadır. LSTM ile oluşturulan modelin RNN'ye göre daha başarılı tahminler yaptığı görülmektedir. Ancak, LSTM modelini eğitmek için kullanılan süre, RNN'yi eğitmek için kullanılanlardan daha uzundur.



Şekil 3.25'te, bu çalışmada önerilen GRU ile oluşturulmuş bir modelin doluluk oranı ve testi için veri setlerinin gerçek ve tahmin edilen değerleri arasındaki ilişkiyi gösteren bir grafik. Grafiklerde görüldüğü gibi GRU en iyi tahmini yapan yöntemdir. GRU ve LSTM'nin tahmin başarısı, RNN'den daha iyidir.



Şekil 3.25 GRU ile Otopark Doluluk Tahmini

Çizelge 3.6’da GRU, LSTM ve RNN’den oluşturulan 27 farklı modelin hata değerlendirme sonuçları sunulmuştur. Çizelgedeki sonuçlar incelendiğinde derin öğrenme tekniklerinin düşük MAE, MSE ve RMSE değerleri ile otopark doluluk oranını tahmin etmede başarılı sonuçlar vermesi, bu modellerin park doluluk tahmin problemlerinde kullanılabileceğini göstermektedir.

Seçilen derin öğrenme teknikleri ile tasarlanan modellerin sonuçları test veri seti üzerinden değerlendirildiğinde, en düşük MAE değeri olan 0,70 ile GRU’nun en iyi tahmin performansına sahip olduğu sonucuna varılmıştır. Sonuçlar RMSE değerleri açısından değerlendirildiğinde en düşük RMSE değeri 1,77 olan önerilen GRU yöntemi en iyi sonuçları vermektedir.

Çizelge 3.6 Modellerin Hata Değerlendirmesi

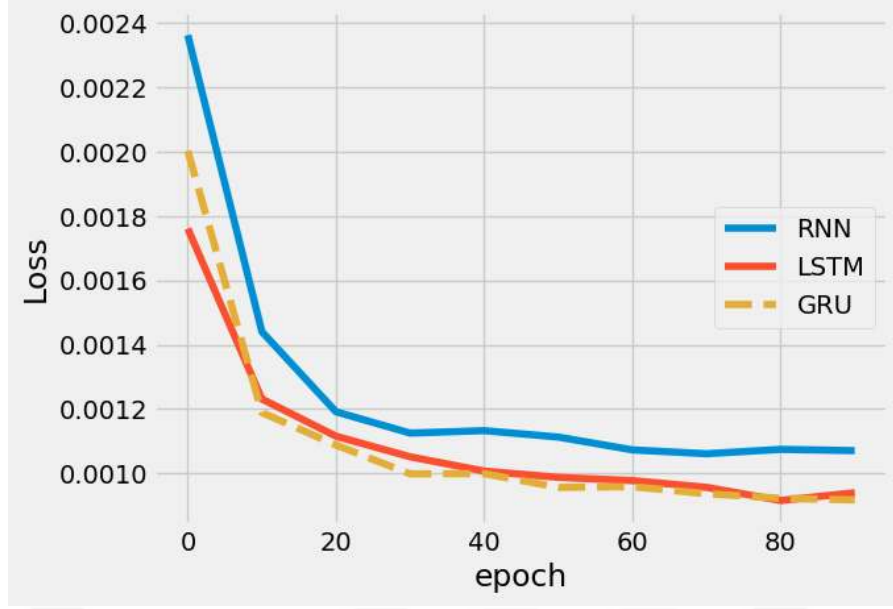
No	(%)Eğitim-Test	Model	Zaman Adımı	MAE	MSE	RMSE	Eğitim Süre
1	70 – 30	GRU	10	0,90	2,35	1,53	00:28:50
2		LSTM		1,56	4,56	2,14	00:36:30
3		RNN		0,95	2,56	1,60	00:11:40
4		GRU	20	0,93	2,54	1,59	00:39:00
5		LSTM		1,46	4,54	2,13	00:46:05
6		RNN		1,03	3,00	1,73	00:12:00
7		GRU	30	1,40	3,98	1,99	00:44:20
8		LSTM		1,62	7,04	2,65	00:46:05

Çizelge 3.6.(devam) Modellerin Hata Değerlendirmesi

9		RNN		1,22	3,12	1,77	00:13:30
10	80 – 20	GRU	10	1,57	6,46	2,54	00:42:20
11		LSTM		1,31	5,28	2,30	01:19:40
12		RNN		1,03	2,67	1,63	00:14:25
13		GRU	20	1,01	2,46	1,57	00:55:15
14		LSTM		1,02	3,12	1,77	00:48:55
15		RNN		2,33	11,41	3,38	00:13:40
16		GRU	30	1,14	3,07	1,75	00:53:40
17		LSTM		0,93	2,36	1,54	01:06:50
18		RNN		1,54	4,69	2,16	00:18:05
19	90 – 10	GRU	10	2,64	9,11	3,02	00:47:10
20		LSTM		1,20	3,14	1,77	00:45:00
21		RNN		1,24	3,95	1,99	00:15:05
22		GRU	20	1,25	3,63	1,91	00:43:55
23		LSTM		1,11	3,18	1,78	00:58:30
24		RNN		2,34	9,52	3,09	00:15:40
25		GRU	30	1,18	3,47	1,86	00:44:40
26		LSTM		2,88	11,95	3,46	00:56:50
27		RNN		1,35	4,62	2,15	00:16:40

Çizelge 3.6 incelendiğinde özet olarak şu sonuçlar çıkarılabilir;

Derin öğrenme modellerinde zaman adımı için en uygun değeri seçmek önemli bir hiper parametredir. Çıktı tahmini için dikkate alınacak değer sayısının belirlenmesinde kullanılan zaman adımlarının sayısı önemlidir. Derin öğrenme modellerinde kullanılacak zaman adımı sayısı konusunda net bir formül bulunmamakla birlikte, çoğunlukla veri seti ve deneylerden elde edilen sonuçlara göre karar verilmektedir. 10, 20 ve 30 zaman adım sayısının seçilmesinde veri seti ve deneylerden elde edilen sonuçlar etkili olmuştur. 10 zaman adımı ve %70-30 minimum öğrenme verisi ile oluşturulan GRU modeli, 0,90 MAE değeri ile en iyi performansı göstermiştir. %90-10 öğrenme ve 30 zaman adımı ile oluşturulan LSTM modeli en kötü performansı göstermiştir. Bu sonuçlar, derin öğrenme modellerinin performansını belirlemede zaman adımlarının sayısının önemli olduğunu göstermektedir. Değerler incelendiğinde lineer olmasa da birbirine paralel olarak arttığı veya azaldığı görülmektedir. 40-50 gibi zaman adımları kullanılırsa değerlerin fazla değişmeyeceği ve aynı paralelliğin devam edeceği öngörülmektedir. En iyi hiper parametrelerde GRU, %99,11, LSTM %98,01, RNN %97,86 maksimum doğruluk oranı elde edilmiştir.



Şekil 3.26 Epoch – Loss Grafiği

Şekil 3.26'de LSTM, GRU ve RNN modellerinin dönem sayısına göre kayıp değerlerinin analizi sunulmaktadır. Başlangıçta çok kayıp yaşayan RNN, dönem sayısı arttıkça neredeyse GRU ve LSTM'yi yakaladığı gözükmemektedir. GRU ve LSTM genel itibariye benzer özellikler göstermektedir. İlk 20 epoch'ta ciddi kayıplar yaşayan modeller, yaklaşık 60. epoch'tan sonra fazla bir değişim göstermemiştir.

İstanbul'daki otoparklarla ilgili literatürde az sayıda çalışma olmasına rağmen yapılan tez çalışmasında derin öğrenme ve mobil cihazlar kullanılarak dinamik bir mimari oluşturulmuş, kısa ve orta vadeli tahminler yapılarak mobil cihazlarda kullanıcılara sunulmuştur. Genel konsept açısından [48]'de önerilen mimarinin yapılan çalışmanın derin öğrenme kısmına benzer olduğu gözlemlenmiştir. İlgili çalışmada ise veri seti oluşturma için kullanılan yöntemin aynısı ISTPARK veri setine uygulanmış ve karşılaştırma bu yeni veri seti üzerinde LSTM çalıştırılarak elde edilen sonuçlarla karşılaştırılmıştır.

4. BULGULAR VE TARTIŞMA

Geliştirilen akıllı şehirler ile IoT cihazları sürekli veri üretiyor ve bu verilere daha kolay erişmemizi sağlıyor. Derin öğrenme teknikleri, içinde bulunduğumuz büyük veri çağında sıklıkla kullanılan bir yöntem haline gelmiştir. Akıllı şehirlerin artan nüfusu ile trafik yoğunluğu artmış ve park alanları yetersiz hale gelmiştir.

Bu çalışmada, akıllı şehirler için derin öğrenmeye dayalı mobil tabanlı yeni bir akıllı park sistemi mimarisi önerilmiştir. Belirli periyotlarda otopark ve hava durumu verileri toplanmıştır. Toplanan verilerle çok değişkenli zaman serileri oluşturulmuştur. Zaman serileri üzerinde başarılı tahminler yapan derin öğrenme modelleri ile kısa ve orta vadede park yerlerinin durumunu tahmin edilip son kullanıcıya sunulmaktadır.

Oluşturulan akıllı park sisteminde otopark doluluk oranını tahmin edecek en iyi yöntemin belirlenmesinde ISTPARK veri kümesi ve hava durumu veri kümesi üzerinde alınan deneysel sonuçlara bakılmıştır. LSTM, SVM, RF ve ARIMA modelleri ISTPARK veri kümesi üzerine uygulanmış yapılan deneylerden elde edilen sonuçlar değerlendirilmiştir. Değerlendirmede LSTM derin öğrenme yöntemi makine öğrenmesi yöntemlerinden çok daha iyi sonuçlar verdiği gözlemlenmiştir. Tez çalışmasının devamında GRU, LSTM ve RNN modelleri ile hem ISTPARK hem de hava durumu verileri üzerinde deneyler yapılmış ve karşılaştırılmıştır. Tüm senaryolarda yapılan deneyler MAE, MSE, RMSE ve Accuracy metriklerine bakılarak değerlendirilmiştir. Elde edilen sonuçlar incelediğinde GRU en iyi performansı göstermiştir. LSTM, GRU'ya yakın performansı gösterirken RNN performans olarak nispeten daha geride kalmıştır. RNN, LSTM ve GRU ile yapılan deneylerden çıkarımlar yapılmıştır.

1. RNN'ler aracılığıyla oluşturulan modellerin eğitim süresi performansları, GRU ve LSTM ile oluşturulan modellerden daha iyidir. Ancak MAE, MSE ve RMSE değerleri daha kötüdür. Test veri seti boyutu azaldıkça RMSE değerlerinde ciddi bir artış olmaktadır.
2. LSTM modelleri MAE, MSE ve RMSE değerleri açısından GRU modellerine çok yakın sonuçlar vermiş hatta bazı modellerde geçmiştir. Ancak genel olarak GRU'nun gerisinde kalmıştır. Aynı zamanda eğitim süresi en uzun olan

yöntemdir. Ayrıca zaman adımı sayısı arttıkça RMSE'de bir artış gözlemlenmiştir.

3. GRU, en düşük MAE, MSE ve RMSE değerine sahip modele sahiptir. Eğitim süresi RNN'ye göre geride kalsa da hafızası olan bir model olma vaadini vermektedir. Ayrıca zaman adımı sayısındaki değişim diğer yöntemlere göre RMSE değerini fazla etkilememiştir.

Yapılan çalışmada İstanbul'daki otoparklardan toplanan veri seti, önerilen derin öğrenme ve bulut tabanlı mimari ile literatürde çalışılan akıllı park sistemleri çözümlerine yeni bir bakış açısı ve yöntem getirmiştir. Literatürdeki yapılan çalışmalar incelendiğinde, [48]'de yapılan çalışmanın derin öğrenme yöntemi bakımından kıyaslandığında yapılan çalışmaya en yakın çalışmadır. İlgili çalışmada veri seti oluşturma için kullanılan yöntemin aynısı (zaman serileri 3 dakikalık frekanslara dönüştürülerek) İSTPARK veri setine uygulanmış ve karşılaştırma bu yeni veri seti üzerinde önerilen modeller çalıştırılarak elde edilen sonuçlarla karşılaştırılmıştır. Veri setleri farklı olsa da, otopark doluluk oranını tahmin ettiği için önerilen çalışmaya en yakın olanıdır. Karşılaştırılan çalışmadaki veri seti, Çin'in Shenzhen kentindeki bir otoparktan 14 gün boyunca toplanan verilerle oluşturulmuştur. Veri seti, otopark doluluk oranı ile tek değişkenli zaman serilerinden oluşmaktadır. Tez çalışmasında oluşturulan İSTPARK veri seti, otopark doluluk oranı tahminini etkileyen 17 farklı özellikten oluşmakta ve yaklaşık 1,5 yılda toplanan 4,5 milyon kayıttan oluşmaktadır. [48]'de kurulan modellerde kullanılan hiper parametreler RF, SVM, ARIMA, LSTM ve GRU modellerine aynı şekilde uygulanarak deneyler yapılmıştır. Elde edilen sonuçlar [48]'deki çalışmada olduğu gibi MAPE metriği üzerinden karşılaştırılmıştır. Çizelge 4.1'de elde edilen sonuçlar verilmiş önerilen yöntemin başarısı açıkça gösterilmiştir.

Çizelge 4.1 Model Karşılaştırılması

	Karşılaştırılan (Tek değişkenli zaman serisi)		Önerilen (Çok değişkenli zaman serisi)				
Zaman Adımı	BP MAPE	LSTM MAPE	RF MAPE	SVM MAPE	ARIMA MAPE	LSTM MAPE	GRU MAPE
1	8,97	6,82	11,03	13,40	7,39	1,02	0,97
5	18,21	9,72	15,66	18,22	12,34	2,40	2,28
10	25,64	12,33	18,46	20,59	15,72	4,29	3,64

Elde edilen bulgular incelendiğinde ve literatürdeki çalışmalar göz önüne alındığında GRU ve LSTM derin öğrenme yöntemi önerilen akıllı park sistemi mimarisinde son kullanıcıya güven verecek sonuçlar vermiştir.

5. SONUÇ VE ÖNERİLER

Derin öğrenme yöntemleri birçok alanda olduğu gibi otopark doluluk tahmininde başarılı sonuçlar vermektedir. Bu çalışmada otopark doluluk tahmini için bulut ve derin öğrenme tabanlı yeni bir mimari ve yeni bir mobil uygulama önerilmiştir. Otopark doluluk oranını tahmin etmek için iki farklı senaryo ve veri seti kullanılmıştır. Birinci senaryoda, çok değişkenli ve büyük veri setlerini neredeyse sorunsuz modelleyen derin öğrenme tabanlı LSTM modeli ve ISTPARK veri kümesi kullanılmıştır. Modelin etkinliğinin gösterilmesi için farklı parametrelerde ve farklı senaryolarda modeller oluşturulup SVM, Random Forest ve ARIMA modelleri ile karşılaştırılmıştır. Çizelge 3.4 ve 3.5’de elde edilen sonuçlar sunulmuştur. Veri seti Capacity, Density, Time, Day ve Holiday kombinasyonu ile eğitildiğinde %99,57 doğruluk oranı elde edilmiştir. İkinci senaryoda makine öğrenmesi yöntemleri üzerinde üstünlük kuran LSTM modeli, GRU ve RNN derin öğrenme modelleri ile karşılaştırılmıştır. Ayrıca otopark kullanımı önemli derece etkileyen hava durumu verileri ISTPARK veri kümesi ile birleştirilerek deneyler yapılmıştır. Çizelge 3.6’te yapılan deneylerde elde edilen sonuçlar sunulmuştur. Bu deneyler sonucunda %99,11 doğruluk oranı ile en iyi performansı GRU derin öğrenme modeli vermiştir. Yapılan iki farklı senaryo elde edilen simülasyon sonuçları incelendiğinde oluşturduğumuz mimari ve uygulamanın başarılı sonuçlar verdiği açıkça gözükmemektedir. Oluşturulan veri seti, akıllı park sistemi mimarisi ve kullanılan yöntemler ile literatüre de farklı alanlarda katkılarda bulunulmuştur.

Çalışmada kullanılan derin öğrenme modelleri ne kadar güncel olsa bile, BLSTM veya CNN-LSTM gibi hibrit modellerle çok değişkenli ISTPARK veri seti üzerinde çalışarak daha başarılı sonuçlar elde edilebilir.

Günümüzde trafik yoğunluğu büyük şehirlerin en büyük problem olarak karşımıza çıkmaktadır. Tabi ki park önünde bekleme, parktan çıkış, park yeri ararken araçların daha yavaş gitmesi bu trafik yoğunluğunun artmasına neden olduğu düşünülmektedir. Sonraki çalışmalarda otoparkların farklı parametreler ile trafik yoğunluğuna etkisi incelenecek ve derin öğrenme yöntemleri kullanılarak trafik yoğunluğunu azaltmak için yeni bir öneri sunulacaktır. Ayrıca önerilen mimari navigasyon uygulamalarına entegre edilerek sürücülerin daha kolay ulaşabilecekleri bir otopark öneri sistemine dönüştürülebilir.

6. KAYNAKLAR

- [1] R. Saborido ve E. Alba, “Software systems from smart city vendors”, *Cities*, c. 101, sayı March, s. 102690, 2020.
- [2] L. Quijano-Sánchez, I. Cantador, M. E. Cortés-Cediel, ve O. Gil, “Recommender systems for smart cities”, *Information Systems*, c. 92, s. 101545, 2020.
- [3] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, ve K. Sakurai, “Implementing Lightweight IoT-IDS on Raspberry Pi Using Correlation-Based Feature Selection and Its Performance Evaluation”, c. 926. *Springer International Publishing*, 2020.
- [4] M. Shafiq, Z. Tian, Y. Sun, X. Du, ve M. Guizani, “Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city”, *Future Generation Computer Systems*, c. 107, ss. 433–442, 2020.
- [5] M. Dener, M. A. Akcayol, S. Toklu, ve O. F. Bay, “Zamana bağlı dinamik en kısa yol problemi için genetik algoritma tabanlı yeni bir algoritma”, *Journal of Faculty of Engineering and Architecture of Gazi University*, c. 26, sayı 4, ss. 915–928, 2011.
- [6] T. Lin, H. Rivano, ve F. Le Mouel, “A Survey of Smart Parking Solutions”, *IEEE Transactions on Intelligent Transportation Systems*, c. 18, sayı 12, ss. 3229–3253, 2017.
- [7] C. Tang, X. Wei, C. Zhu, W. Chen, ve J. J. P. C. Rodrigues, “Towards smart parking based on fog computing”, *IEEE Access*, c. 6, ss. 70172–70185, 2018.
- [8] F. Canitez, D. Çelebi, ve E. Beyazit, “Establishing a metropolitan transport authority in Istanbul: A new institutional economics framework for institutional change in urban transport”, *Case Studies on Transport Policy*, c. 7, sayı 3, ss. 562–573, 2019.
- [9] D. C. Shoup, “Cruising for parking”, *Transport Policy*, c. 13, sayı 6, ss. 479–486, 2006.
- [10] C. Dowling, T. Fiez, L. Ratliff, ve B. Zhang. (2018, 11 Mayıs). *How Much Urban Traffic is Searching for Parking? Simulating Curbside Parking as a Network of Finite Capacity Queues*. [Online]. Erişim: <http://arxiv.org/abs/1702.06156>.
- [11] I. Kök, M. U. Şimşek, ve S. Özdemir, “A deep learning model for air quality prediction in smart cities”, *Proc. - 2017 IEEE International Conference on Big Data*, c. 2018-Janua, ss. 1983–1990, 2017.
- [12] F. Al-Turjman ve A. Malekloo, “Smart parking in IoT-enabled cities: A survey”, *Sustainable Cities and Society*, c. 49, sayı December 2018, 2019.
- [13] B. Kizilkaya, M. Caglar, F. Al-Turjman, ve E. Ever, “Binary search tree based hierarchical placement algorithm for IoT based smart parking applications”, *Internet of Things*, c. 5, ss. 71–83, 2019.
- [14] S. Yang, W. Ma, X. Pi, ve S. Qian, “A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-

- temporal data sources”, *Transportation Research Part C Emerging Technologies*, c. 107, sayı August, ss. 248–265, 2019.
- [15] J. Hanzl, “Parking Information Guidance Systems and Smart Technologies Application Used in Urban Areas and Multi-storey Car Parks”, *Transportation Research Procedia*, c. 44, sayı 2019, ss. 361–368, 2020.
 - [16] J. H. Shin ve H. B. Jun, “A study on smart parking guidance algorithm”, *Transportation Research Part C Emerging Technologies*, c. 44, ss. 299–317, 2014.
 - [17] Y. Geng ve C. G. Cassandras, “A new ‘Smart Parking’ System Infrastructure and Implementation”, *Procedia Social and Behavioral Sciences*, c. 54, sayı may 1877, ss. 1278–1287, 2012.
 - [18] Y. Atif, J. Ding, ve M. A. Jeusfeld, “Internet of Things Approach to Cloud-based Smart Car Parking”, *Procedia Computer Science*, c. 58, sayı Euspn, ss. 193–198, 2016.
 - [19] D. Thomas ve B. C. Koor, “A Genetic Algorithm Approach to Autonomous Smart Vehicle Parking system”, *Procedia Computer Science*, c. 125, ss. 68–76, 2018.
 - [20] Q. G. K. Safi, S. Luo, L. Pan, W. Liu, R. Hussain, ve S. H. Bouk, “SVPS: Cloud-based smart vehicle parking system over ubiquitous VANETs”, *Comput. Networks*, c. 138, ss. 18–30, 2018.
 - [21] R. Lu, X. Lin, H. Zhu, ve X. Shen, “SPARK: A new VANET-based smart parking scheme for large parking lots”, *Proceedins - IEEE INFOCOM*, ss. 1413–1421, 2009.
 - [22] D. Huang, G. D. Jiang, B. H. Sun, R. Q. Feng, ve J. P. Wu, “Smart park integrated management cloud platform architecture based on microservice governance framework”, *IOP Conference Series: Earth and Environmental Science*, c. 354, sayı 1, 2019.
 - [23] B. Costa, P. F. Pires, ve F. C. Delicato, “Towards the adoption of OMG standards in the development of SOA-based IoT systems”, *Journal of Systems and Software*, c. 169, s. 110720, 2020.
 - [24] L. Deng ve C. Zhou, “Prototype Framework of Smart City Base on Big Data and Smart Grid”, *International Conference on Computer Science and Mechanical Automation (CSMA) 2015*, ss. 102–107, 2016.
 - [25] H. Elhoseny, M. Elhoseny, S. Abdelrazek, H. Bakry, ve A. Riad, “Utilizing Service Oriented Architecture (SOA) in Smart Cities Utilizing Service Oriented Architecture (SOA) in Smart Cities BSC Faculty of Computer Science and Information Systems , Mansoura Lecturer of Information Systems , Faculty of Computer Scienc”, *International Journal of Advanced Computer Technology*, sayı June, 2016.
 - [26] K. C. Lan ve W. Y. Shih, “An intelligent driver location system for smart parking”, *Expert Systems with Applications*, c. 41, sayı 5, ss. 2443–2456, 2014.
 - [27] M. Bachani, U. M. Qureshi, ve F. K. Shaikh, “Performance Analysis of Proximity and Light Sensors for Smart Parking”, *Procedia Computer Science*, c. 83, sayı Ant, ss. 385–392, 2016.
 - [28] Z. Pala ve N. Inanç, “Smart parking applications using RFID technology”, 2007

1st Annual RFID Eurasia, 2007.

- [29] CIEC, “The Essential Chemical Industry – Online”, *2016 International Conference Internet Things Applications*, ss. 266–270, 2016.
- [30] I. Chatzigiannakis, A. Vitaletti, ve A. Pyrgelis, “A privacy-preserving smart parking system using an IoT elliptic curve based security platform”, *Computer Communications*, c. 89–90, ss. 165–177, 2016.
- [31] E. Polycarpou, L. Lambrinos, ve E. Protopapadakis, “Smart parking solutions for urban areas”, *2013 IEEE 14th International Symposium a World Wireless, Mobile and Multimedia Networks, WoWMoM 2013*, ss. 1–6, 2013.
- [32] J. Chinrungrueng, U. Sunantachaikul, ve S. Triamlumlerd, “Smart parking: An application of optical wireless sensor network”, *SAINT - 2007 International Symposium Applications Internet - Work. SAINT-W*, s. 66, 2007.
- [33] I. Belhajem, Y. Ben Maissa, ve A. Tamtaoui, “Improving Vehicle Localization in a Smart City with Low Cost Sensor Networks and Support Vector Machines”, *Mobile Networks and Applications*, c. 23, sayı 4, ss. 854–863, 2018.
- [34] G. Bovenzi, D. Ciuonzo, V. Persico, A. Pescapè, ve P. S. Rossi, “IoT-enabled distributed detection of a nuclear radioactive source via generalized score tests”, *Communications in Computer and Information Science*, c. 968, sayı December, ss. 77–91, 2019.
- [35] S. H. Javadi, H. Moosaei, ve D. Ciuonzo, “Learning wireless sensor networks for source localization”, *Sensors (Switzerland)*, c. 19, sayı 3, ss. 1–17, 2019.
- [36] P. R. L. De Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, ve A. L. Koerich, “PKLot-A robust dataset for parking lot classification”, *Expert System with Applications*, c. 42, sayı 11, ss. 4937–4949, 2015.
- [37] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, ve C. Vairo, “Deep learning for decentralized parking lot occupancy detection”, *Expert System with Applications*, c. 72, ss. 327–334, 2017.
- [38] G. Amato, F. Carrara, F. Falchi, C. Gennaro, ve C. Vairo, “Car parking occupancy detection using smart camera networks and Deep Learning”, *Proceeding - IEEE Symposium Computer and Communications*, c. 2016-Augus, ss. 1212–1217, 2016.
- [39] D. Acharya, W. Yan, ve K. Khoshelham, “Real-time image-based parking occupancy detection using deep learning”, *CEUR Workshop Proceedings*, c. 2087, ss. 33–40, 2018.
- [40] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj, ve K. Liu, “An edge based smart parking solution using camera networks and deep learning”, *Proc. - 2018 IEEE International Conference on Cognitive Computing ICC3 2018 - Part 2018 IEEE World Congress Services*, ss. 17–24, 2018.
- [41] M. Napiorkowska, D. Petit and P. Marti, “Three Applications of Deep Learning Algorithms for Object Detection in Satellite Imagery”, *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, ss. 4839–4842, 2018.
- [42] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, ve M. Zuair, “Deep learning approach for car detection in UAV imagery”, *Remote Sensing*, c. 9, sayı 4, 2017.
- [43] S. Valipour, M. Siam, E. Stroulia, ve M. Jagersand, “Parking-stall vacancy

- indicator system, based on deep convolutional neural networks”, *2016 IEEE 3rd World Forum Internet Things, WF-IoT 2016*, ss. 655–660, 2017.
- [44] V. Jain. (2019, 10 Eylül). *Doing Multivariate Time Series Forecasting with Recurrent Neural Networks*. [Online]. Erişim: <https://databricks.com/blog/2019/09/10/doing-multivariate-time-series-forecasting-with-recurrent-neural-networks.html>.
 - [45] Y. Zheng, S. Rajasegarar, C. Leckie, ve M. Palaniswami, “Smart car parking: Temporal clustering and anomaly detection in urban car parking”, *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intellihent Sensors, Sensor Networks Information Processing*, sayı April, ss. 21–24, 2014.
 - [46] S. Yamin Siddiqui, M. Adnan Khan, S. Abbas, ve F. Khan, “Smart occupancy detection for road traffic parking using deep extreme learning machine”, *Journal of King Saud University - Computer and Information Sciences*, ss. 1–7, 2020.
 - [47] Y. Zheng, S. Rajasegarar, ve C. Leckie, “Parking availability prediction for sensor-enabled car parks in smart cities”, *2015 IEEE 10th International Conference Intelligent Sensors, Sensor Networks and Information Processing ISSNIP 2015*, sayı April, ss. 7–9, 2015.
 - [48] L. Xiangdong, C. Yuefeng, C. E. N. Gang, ve X. Zengwei, “Prediction of short-term available parking space using LSTM model”, *14th International Conference Computer Science Education ICCSE 2019*, sayı Iccse, ss. 631–635, 2019.
 - [49] O. Brun, Y. Yin, J. A. Gonzalez, M. Ramos, E. Gelenbe, “IoT Attack Detection with Deep Learning To cite this version : HAL Id : hal-02062091 IoT Attack Detection with Deep Learning”, *Isc Security Work*, Feb 2018, Londres, United Kingdom, 2019.
 - [50] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, ve P. Campoy, “A review of deep learning methods and applications for unmanned aerial vehicles”, *Journal of Sensors*, c. 2017, 2017.
 - [51] S. Cakir, S. Toklu, ve N. Yalcin, “RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning”, *IEEE Access*, c. 8, ss. 183678–183689, 2020.
 - [52] F. Y. Yavuz, D. Ünal, ve E. Gül, “Deep learning for detection of routing attacks in the internet of things”, *International Journal of Computational Intelligence Systems*, c. 12, sayı 1, ss. 39–58, 2018.
 - [53] G. Thamilarasu ve S. Chawla, “Towards deep-learning-driven intrusion detection for the internet of things”, *Sensors (Switzerland)*, c. 19, sayı 9, 2019.
 - [54] Google. (2020, 11 Ocak). *Keras Documentation: About Keras*. [Online]. Erişim: <https://keras.io/about/>
 - [55] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, ve S. Khudanpur, “Extensions of recurrent neural network language model”, *ICASSP, IEEE International Conference Acoustics Speech Signal Processing - Proceedings*, ss. 5528–5531, 2011.
 - [56] Z. Huang, F. Yang, F. Xu, X. Song, ve K. L. Tsui, “Convolutional Gated Recurrent Unit-Recurrent Neural Network for State-of-Charge Estimation of Lithium-Ion Batteries”, *IEEE Access*, c. 7, ss. 93139–93149, 2019.

- [57] H. Yan, Y. Qin, S. Xiang, Y. Wang, ve H. Chen, “Long-term gear life prediction based on ordered neurons LSTM neural networks”, *Journal of the International Measurement Confederation*, c. 165, s. 108205, 2020.
- [58] X. H. Le, H. V. Ho, G. Lee, ve S. Jung, “Application of Long Short-Term Memory (LSTM) neural network for flood forecasting”, *Water (Switzerland)*, c. 11, sayı 7, 2019.
- [59] D. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, ve Z. Sun, “Well production forecasting based on ARIMA-LSTM model considering manual operations”, *Energy*, c. 220, s. 119708, 2021.
- [60] F. Rui, Z. Zuo, L. Li “Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction”, *31st Youth Academic Annual Conference of Chinese Association of Automation*, ss. 5–9, 2016.
- [61] S. Troia, R. Alvizu, Y. Zhou, G. Maier, ve A. Pattavina, “Deep Learning-Based Traffic Prediction for Network Optimization”, *International Conference on Transparent Optical Networks*, c. 2018-July, sayı July, 2018.
- [62] M. Phi. (2018, 24 Eylül). *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. [Online]. Erişim: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- [63] P. S. Kim, D. G. Lee, ve S. W. Lee, “Discriminative context learning with gated recurrent unit for group activity recognition”, *Pattern Recognition*, c. 76, ss. 149–161, 2018.
- [64] J. Wang, J. Yan, C. Li, R. X. Gao, ve R. Zhao, “Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction”, *Computers in Industry*, c. 111, ss. 1–14, 2019.
- [65] W. Li, H. Wu, N. Zhu, Y. Jiang, J. Tan, ve Y. Guo, “Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU)”, *Information Processing in Agriculture*, ss. 1–9, 2020.
- [66] J. Chung, C. Gulcehre, K. Cho, ve Y. Bengio. (2014, 11 Eylül). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. [Online]. Erişim: <http://arxiv.org/abs/1412.3555>.
- [67] I. El Naqa ve M. J. Murphy, “Machine Learning in Radiation Oncology”, *Machine Learning in Radiation Oncology*, ss. 3–11, 2015.
- [68] G. Shobha ve S. Rangaswamy, *Machine Learning*, 1. baskı, c. 38. Elsevier BV, 2018.
- [69] G. Biau ve E. Scornet, “A random forest guided tour”, *Sociedad de Estadística e Investigación Operativa*, c. 25, sayı 2, ss. 197–227, 2016.
- [70] P. Bonissone, J. M. Cadenas, M. Carmen Garrido, ve R. Andrés Díaz-Valladares, “A fuzzy random forest”, *International Journal of Approximate Reasoning*, c. 51, sayı 7, ss. 729–747, 2010.
- [71] Y. Qi, “Ensemble Machine Learning”, *Ensemble Machine Learning*, ss. 307–323, 2012.
- [72] Y. Hiroyasu, M. Yuji, “Support Vector Machine how to apply to the multi-valued classification problem”, *Information Processing Society of Japan Research Report Natural Language Processing Research Group Report* c. 2001, sayı 112, ss. 33–

38, 2001.

- [73] D. Meyer, F. Leisch, ve K. Hornik, “The support vector machine under test”, *Neurocomputing*, c. 55, sayı 1–2, ss. 169–186, 2003.
- [74] W. S. Noble, “What is a support vector machine?”, *Nature Biotechnology*, c. 24, sayı 12, ss. 1565–1567, 2006.
- [75] O. Kaynar, S. Taştan, “Zaman Serileri Tahmininde Arima-Mlp Melez Modeli”, *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, ss. 141–149, 2009.
- [76] J. Wang, Q. Gu, J. Wu, G. Liu, ve Z. Xiong, “Traffic speed prediction and congestion source exploration: A deep learning method”, *Proceeding - IEEE International Conference Data Mining, ICDM*, c. 0, ss. 499–508, 2016.
- [77] X. Ping, F. Yang, H. Zhang, W. Zhang, G. Song, ve Y. Yang, “Prediction and optimization of isentropic efficiency of vortex pump under full operating conditions in Organic Rankine Cycle waste heat recovery system based on deep learning and intelligent algorithm”, *Sustainable Energy Technologies and Assessments*, c. 42, sayı October, s. 100898, 2020.
- [78] V. H. Nhu, N. D. Hoang, H. Nguyen, P. T. T. Ngo, T. T. Bui, P. V. Hoa, P. Samui, D. T. Bui, “Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area”, *Catena*, c. 188, sayı November 2019, s. 104458, 2020.
- [79] S. Abirami ve P. Chitra, “Regional air quality forecasting using spatiotemporal deep learning”, *Journal of Cleaner Production*, c. 283, s. 125341, 2021.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Hikmet CANLI

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Bilgisayar Müh.	Düzce Üniversitesi	2022
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2017
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2015
Lise	Fen Bilimleri	Gölköy Lisesi	2010

YAYINLAR

Uluslararası Hakemli Dergilerde Yayınlanan Makaleler

1. Canlı, H., Toklu, “Deep Learning-Based Mobile Application Design for Smart Parking” *IEEE Access*, vol. 9, pp. 61171-61183, 2021
2. Canlı, H., Toklu, “Design and Implementation of a Prediction Approach Using Big Data and Deep Learning Techniques for Parking Occupancy”, *Arab J Sci Eng* ,pp. 1-16, 2021
3. Canlı, H., Toklu, “AVL based settlement algorithm and reservation system for smart parking systems in IoT-based smart”, *IAJIT* ,(Accepted), 2022

Ulusal Hakemli Dergilerde Yayınlanan Makaleler

1. Canlı, H., Toklu, “Implementation of Decision Support System with Data Mining Methods in the Quality Control Process of the Automotive Sector” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 7, pp. 102-114, 2019

Ulusal Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında Basılan Bildiriler:

1. Canlı, H., Toklu, “Decision Support System for Quality Control with Data Mining Method in Production Sector ” *International Symposium on Industry 4.0 and Applications (ISIA)*, 2017

