

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞ KAYNAK KULLANIMLARININ ZAMAN SERİSİ
ANALİZLERİ İLE KESTİRİMİ

BÜŞRA OKCU
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2023

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞ KAYNAK KULLANIMLARININ
ZAMAN SERİSİ ANALİZLERİ İLE
KESTİRİMİ

BÜŞRA OKCU
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DOÇ. DR. HABİL KALKAN

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES

**ESTIMATION OF NETWORK RESOURCE
UTILIZATIN WITH TIME SERIES ANALYSIS**

BÜŞRA OKCU

**A THESIS SUBMITTED FOR THE DEGREE OF MASTER
OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING**

THESIS SUPERVISOR
ASS. PROF. DR. HABİL KALKAN

GEBZE
2023

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 11/05/2023 tarih ve 2023/28 sayılı kararıyla oluşturulan jüri tarafından 07/06/2023 tarihinde tez savunma sınavı yapılan Büşra OKCU'nun tez çalışması Bilgisayar mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Doç Dr. Habil KALKAN

ÜYE

: Dr. Burcu YILMAZ

ÜYE

: Dr. Füsun ER

ONAY

Gebze Teknik Üniversitesi Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Büyük veri analizinin önemli dallarından biri olan zaman serisi tahmini ile, bir ya da daha fazla değişkene sahip geçmiş zamana ait veriler kullanılarak gelecek zaman noktalarına ait verilerin tahmin edilmesi hedeflenir. Günümüzde bulut bilişim ve büyük verinin popülaritesinin artması ile birlikte, veri merkezlerinde konumlandırılan fiziksel sunucuların gelecek dönemlerdeki kaynak kullanımlarının tahmin edilmesi, zaman serisi analizi alanında yapılan araştırmalar için yenilikçi ve gelişime açık bir alan haline gelmiştir. Veri merkezleri, bulut bilişime hizmet eden fiziksel sunucuları ve bu sunucuların donanım ekipmanlarını depolar. Veri merkezlerinde çalışan bu makinelerde iş birimlerini programlayan yönetim mevcut CPU/RAM kullanımı ve çalışan iş sayısı bilgilerini dikkate alamadığından hizmet sağlayıcılarının verimli bir şekilde kaynak tahsis etmesini zorlaştırır. Bu nedenle gerçekleştirilen çalışmada amaç, oldukça geniş bir hizmet ağına sahip olan bir bulut servis sağlayıcısına ait veri merkezlerinde konumlanan kümelerin kaynak kullanımlarını tahmin etmektir. Bu amaçla istatistiksel tahmin modelleri ve derin öğrenme tekniklerinden faydalananarak 44 adet öznitelik kullanılarak 6 farklı kaynak kullanım tahmin modeli geliştirilmiştir. İlk olarak bulut hizmetlerine ait iş yüklerinde uzun menzilli bağımlılığın ARIMA ve VAR gibi geleneksel yöntemler ile farklı geliştirilen tahmin modellerini nasıl etkilediği incelenmiştir. Derin öğrenme tekniklerinden ise RNN, LSTM, BiLSTM ve BiLSTM-RNN modelleri geliştirilerek, tahmin için kullanılacak olan geçmiş zaman adımı noktası sayısı yani pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde modellere verilmiştir ve gelecek 28 zaman adımı için çok adımlı zaman serisi tahmini yapılmıştır. 2 ayrı veri ayrıştırma yöntemi ile test edilmiştir. Son olarak geliştirilen çok değişkenli çok adımlı zaman serisi tahmin modellerine ait deneysel sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler: Derin Öğrenme, İstatistiksel Tahmin, Zaman Serisi, Tahmin Modelleri, ARIMA, LSTM, RNN, Kaynak Kullanım Tahmini.

SUMMARY

With time series estimation, which is one of the important branches of big data analysis, it is aimed to predict the data of future time points by using data from the past with one or more variables. Today, with the increasing popularity of cloud computing and big data, estimating future resource usage of physical servers located in data centers has become an innovative and open field for researches in the field of time series analysis. Data centers store physical servers serving cloud computing and the hardware equipment of these servers. Management that schedules work units on these machines running in data centers cannot take into account available CPU/RAM usage and worker job count information, making it difficult for service providers to allocate resources efficiently. For this reason, the aim of the study is to estimate the resource usage of the clusters located in the data centers of a cloud service provider with a wide service network. For this purpose, 6 different resource usage forecasting models were developed by using 44 features by using statistical forecasting models and deep learning techniques. First of all, it is examined how long-range dependency in cloud services workloads affects prediction models developed differently with traditional methods such as ARIMA and VAR. Among the deep learning techniques, RNN, LSTM, BiLSTM and BiLSTM-RNN models were developed and the number of past time step points to be used for prediction, that is, window sizes, were given to the models as 1 week, 2 weeks and 3 weeks, respectively, and multi-step for the next 28 time steps. time series estimation was made. It has been tested with 2 different data parsing methods. Finally, the experimental results of the developed multivariate multi-step time series forecasting models were compared.

Key Words: Deep Learning, Statistical Forecasting, Time Series, Forecast Models, ARIMA, LSTM, RNN, Resource Usage Forecast.

TEŞEKKÜR

Öncelikle yüksek lisans eğitimimde desteğini ve yardımlarını hiçbir zaman esirgemeyip, tez çalışmasının kapsamının belirlenmesinden tüm araştırmanın yürütülmesine kadar her adımda bilgisi ile bu çalışmanın gelişmesini ve gerçekleşmesini sağlayan danışmanım Doç. Dr. Habil Kalkan'a, çalışmamın temelini oluşturan veri setini hazırlamama olanak sağlayan görev aldığım şirket Turkcell'e ve değerli yöneticilerime, göstermiş olduğu manevi desteklerinden dolayı canım aileme ve sevgili eşim Salih Okcu'ya en içten teşekkürlerimi sunarım.



İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER VE KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xii
TABLolar DİZİNİ	xiii
1. GİRİŞ	1
1.1. Tezin Amacı, Katkısı ve İçeriği	2
2. LİTERATÜR ARAŞTIRMASI	4
3. ZAMAN SERİSİ ANALİZİ	12
3.1. Zaman Serisinin Bileşenleri	12
3.2. Tek Değişkenli Zaman Serisi Modelleri	13
3.3. Çok Değişkenli Zaman Serisi Modelleri	14
3.4. Birim Kök Testleri	14
3.4.1 Dickey-Fuller Testi	15
3.4.2 Genişletilmiş Dickey-Fuller Testi	16
3.5. Bilgi Kriterleri	17
3.5.1 Akaike Bilgi Kriteri	17
3.5.2 Bayes Bilgi Kriteri	18
4. YÖNTEM VE ARAÇLAR	20
4.1. VERİ SETİNİN OLUŞTURULMASI	21
4.2. VERİ ÖNİŞLEME	23
4.2.1 Boş Oluşan Raporlar ve Kapatılan Kümeler	24
4.2.2 Eksik Değerlerin Doldurulması	24
4.2.3 Düzgeleme	25
4.3. MODEL GELİŞTİRME	26
4.3.1 İstatistiksel Tahminleme Modelleri	26

4.3.2 Derin Öğrenme Teknikleri	32
4.4. VERİ AYIRMA YÖNTEMLERİ	39
4.4.1 Veri Bazında Ayırma Yöntemi	39
4.4.2 Küme Bazında Ayırma Yöntemi	39
5. DENEYSEL SONUÇLAR	40
6. DEĞERLENDİRME	55
KAYNAKLAR	57
ÖZGEÇMİŞ	62



SİMGELER ve KISALTMALAR DİZİNİ

<u>Simgeler ve</u>	<u>Açıklamalar</u>
<u>Kısaltmalar</u>	
μ	: Aritmetik ortalama
$\prod$	: Çarpım sembolü
ε_t	: Hata terimi
ρ	: Korelasyon katsayısı
σ	: Standart sapma
τ	: Tau
Σ	: Toplam sembolü
%	: Yüzde ifadesi
ADF	: Genişletilmiş Dickey-Fuller
AIC	: Akaike Bilgi Kriteri
AR	: Otoregresyon
ARIMA	: Otoregresif Bütünleşik Hareketli Ortalama
ARMA	: Otoregresif Hareketli Ortalama
BIC	: Bayes Bilgi Kriteri
BiLSTM	: Çift Yönlü Uzun Kısa Süreli Bellek
BRR	: Bayes Yükselti Regresyonu
CDF	: Kümülatif Dağılım Fonksiyonu
CPU	: Merkezi İşlem Birimi
DF	: Dickey-Fuller
DT	: Karar Ağacı
FLNN	: Fonksiyonel Bağlantı Sinir Ağı
GA	: Genetik Algoritma
GB	: Gigabayt
GTÜ	: Gebze Teknik Üniversitesi
HTTP	: Hiper Metin Aktarım Protokolü
KB	: Kilobayt
LSTM	: Uzun Kısa Süreli Bellek
MA	: Hareketli Ortalama

MB	: Megabayt
MLP	: Çok Katmanlı Algılayıcılar
MSE	: Kök Kare Hatası
OED	: Dikey Deney Tasarımı
PSO	: Parçacık Sürü Optimizasyonu
RAM	: Rasgele Erişim Belleği
ReLU	: Rectified Linear Units
RMSE	: Ortalama Kök Kare Hatası
RNN	: Yinelemeli Sinir Ağı
SIC	: Schwarz Bilgi Kriteri
SRD	: Servis Yedekliliği Programı
SVM	: Karar Destek Makinası
SVR	: Destek Vektör Regresyonu
TB	: Terabayt
VAR	: Vektör Otoregresyon
VM	: Sanal Sunucu
vROPs	: vRealize Operasyon Yönetimi

ŞEKİLLER DİZİNİ

<u>Şekil No:</u>	<u>Sayfa:</u>
Şekil 4.1: Küme mimarisi.	20
Şekil 4.2: Yinelemeli sinir ağı modeli.	34
Şekil 4.3: LSTM model yapısı.	35
Şekil 4.4: BiLSTM mimarisi.	38
Şekil 5.1: ARIMA modeli – “gerçek-tahmin” grafiği.	41
Şekil 5.2: VAR modeli – “gerçek-tahmin” grafiği.	43
Şekil 5.3: RNN modeli “gerçek-tahmin” grafiği.	45
Şekil 5.4: LSTM modeli “gerçek-tahmin” grafiği.	46
Şekil 5.5: BiLSTM modeli “gerçek-tahmin” grafiği.	47
Şekil 5.6: BiLSTM-RNN modeli “gerçek-tahmin” grafiği.	48
Şekil 5.7: Küme bazında ayırma yöntemi RMSE sonuçları	49
Şekil 5.8: Veri ayrıştırma yöntemleri RMSE sonuçları.	50
Şekil 5.9: Küme bazında ayırma yöntemi RMSE sonuçları (k-katlamalı çapraz doğrulama).	51
Şekil 5.10: Pencere boyutlarına göre model bazında RMSE sonuçları.	52
Şekil 5.11: “t+28” zaman adımı tahmini için pencere boyutlarına göre model bazında RMSE sonuçları.	53
Şekil 5.12: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=1 Hafta).	53
Şekil 5.13: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=2 Hafta).	54
Şekil 5.14: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=3 Hafta).	54

TABLolar DİZİNİ

<u>Tablo No:</u>	<u>Sayfa:</u>
Tablo 4.1: Öznitelikler.	22
Tablo 4.2: "Haftanın Günü" öznitelğine one-hot-encoding uygulanışı.	23
Tablo 4.3: Genişletilmiş Dickey-Fuller Testi Sonucu.	29
Tablo 4.4: OtomatikARIMA test sonuçları.	30
Tablo 5.1: ARIMA modeli küme bazında ayırma yöntemi RMSE değerleri.	40
Tablo 5.2: Öznitelik – RMSE değerleri.	42
Tablo 5.3: RNN modeli küme bazında ayırma yöntemi RMSE değerleri.	44
Tablo 5.4: LSTM modeli küme bazında ayırma yöntemi RMSE değerleri.	45
Tablo 5.5: BiLSTM modeli küme bazında ayırma yöntemi RMSE değerleri.	47
Tablo 5.6: BiLSTM-RNN modeli küme bazında ayırma yöntemi RMSE değerleri.	48
Tablo 5.7: Küme bazında ayırma yöntemi (k-katlamalı çapraz doğrulama) tur bazında RMSE sonuçları	51

1. GİRİŞ

Zaman serisi analizi; zaman birimlerinde organize edilen verilerin analizidir. Zaman serileri kronolojik olarak üretilir ve aşağıdaki şekilde bir X vektörü olarak tanımlanabilir:

$$X = \{x(1), x(2), x(3), \dots, x(n)\} \quad (1.1)$$

Zaman serisi, sıralanmış veri noktalarının bir dizisidir. Bu diziye ait verilerin analiz edilmesinde geçmiş ve şimdiki zaman ölçümleri kullanılarak gelecek zaman verilerinin eğiliminin tahmin edilmesi hedeflenir. Böylece geçmiş veriler analiz edilirken, güncel veriler izlenebilir ve gelecek zaman adımları hakkında fikir sahibi olunabilir. Zaman sıralı gelen düzenli verilerden anlamlı bilgiler çıkarmak amacı ile bir takım istatistiki analiz modelleri uygulanır ve bu analiz ile öngörü modelleri oluşturulabilir.

Zaman serileri yüksek boyut ve zamansal bağımlılık özelliklerine sahiptir. Yüksek boyut özelliği her bir zaman noktasının ayrı bir boyut olarak ele alınmasını ifade eder. Zamansal bağımlılıklar ise sayısal olarak özdeş iki noktanın farklı sınıflara ait olabilmesi anlamına gelir. Zaman serileri, zaman noktasındaki örnekleme değişkenlerinin sayısına göre ikiye ayrılmaktadır. Bunlar tek değişkenli zaman serileri ve çok değişkenli zaman serileridir [1], [2], [3].

Borsa fiyatları takibinden bir salgın yayılımının takibine, uyku izlemeden bir ses sinyalinin kaydedilmesine kadar gerçek dünya verilerinin zaman serisi olarak kaydedilmesi oldukça yaygın bir yöntem olup bu zaman serileri verilerinden anlamlı bilgi çıkarmak için literatürde çok sayıda çalışma gerçekleştirilmiştir. Zaman serisi verilerinde yapılan analiz ve tahmin çalışmalarında özbağımlısal tümleşik hareketli ortalama (ARIMA) gibi geleneksel yöntemler yaygın olarak kullanılmıştır [3-7]. Fakat, makine öğrenmesi modellerinin gelişmesi ile birlikte Çok Katmanlı Algılayıcılar (MLP), Karar Destek Makinası (SVM), Karar Ağaçları (DT) gibi çeşitli algoritmalar ile birtakım zaman serisi analiz çalışmaları da gerçekleştirilmiştir [2], [8], [9]. Son yıllarda ise derin öğrenme modelleri giderek gelişmiş, diğer birçok alanda olduğu gibi, zaman serisi analizinde de popülerliğini artırarak klasik makine öğrenmesi modellerinin önüne geçmiştir. Derin öğrenme, basit ancak doğrusal olmayan modülleri birleştirerek orijinal girdinin çok düzeyli bir temsiliyi elde edebilen

yeni bir yaklaşımdır. GPU hızlarının artması, derin ağların daha etkin ve daha hızlı eğitilebilmesini sağladığından, geliştirilen tahmin modellerinin başarı oranlarını da artırmıştır. Bu sayede hemen hemen her alanda kullanımı yaygınlaşmıştır. [2], [8].

Bu tez çalışmasında ise Türkiye’de faaliyet gösteren en büyük altyapıya sahip bulut servis sağlayıcısına ait veri merkezinden elde edilen veriler kullanılarak bir dizi zaman serisi analiz çalışması gerçekleştirilmiştir. Çalışmada, 01 Mart 2019 ile 01 Şubat 2022 tarihleri arasında farklı sayıda sunucu içeren 53 adet kümenin bellek kaynak kullanımına ilişkin verileri toplanmış ve veriler çok değişkenli bir zaman sırası ile düzenlenmiştir. Yapılan incelemeler ile her bir kümenin geçmiş dönem verileri kullanılarak bellek kaynaklarının yüzdelik kullanımına yönelik deneysel tahmin çalışmaları gerçekleştirilmiştir.

1.1. Tezin Amacı, Katkısı ve İçeriği

Son dönemde popülaritesi giderek artan ve yaygın bir kullanım alanına sahip olan bulut bilişim, farklı sayıda kullanıcının hızlı ve dinamik bir şekilde merkezi veya dağıtık hizmet havuzuna paylaşımlı olarak eşzamanlı erişimini sağlar. Bulut bilişimin esneklik, depolama ve olağanüstü durum kurtarma gibi avantajlı nitelikleri bulunmaktadır. Bulut bilişim sistemlerine hizmet eden her veri merkezinde çalışan tipik makineler, gün içerisinde yüzlerce işi çalıştırabilirler. Değişen kaynak gereksinimlerine sahip bu makinelere iş birimlerini programlayan sistemler, veri merkezlerinde artan zaman karmaşıklıklarının üstesinden gelme konusunda ve bunlara uyum sağlamada başarısız olabilmektedir. Çünkü, yönetim sistemleri, makinenin merkezi işlem birimi (CPU) kullanımı, bellek (RAM) kullanımı ve çalışan iş sayısı gibi mevcut durumunu dikkate alamamaktadır. Bu durum ise hizmet sağlayıcısının verimli bir şekilde kaynak tahsis edebilmesini zorlaştırmaktadır. Veri merkezleri ise hizmet sağlamada sürekliliği sağlamak için kaynak kapasitesini artırmakta ve sürekli talep edilenden fazla kaynağı kullanıma hazır halde tutma yoluna gitmektedir. Google’ın veri merkezinde yer alan makineleri için bu durum incelendiğinde, sunucuların CPU kaynaklarının yaklaşık %45’ini kullandığı gözlemlenirken, RAM kaynaklarının ise %60 civarında kullanıldığı gözlemlenmiştir. Bu sebeplerden ötürü, veri merkezlerindeki CPU ve RAM kullanım ihtiyaçlarının tahmin edilebilmesi, bulut hizmetleri işletmecilerinin daha verimli kapasite planlaması yapabilmesini ve

kaynakların doğru alanlarda yönetilebilmesini sağlar. Ayrıca aktif çalışan makinelerin sayısının da dinamik olarak ölçeklendirilmesine katkı sağlar [4], [9], [10].

Gelişen bu ihtiyaçlar doğrultusunda bu tez kapsamında, Türkiye'nin en büyük veri merkezi altyapısına sahip bulut hizmet sağlayıcısında yer alan ve içerisinde farklı sayıda sunucu barındıran 53 adet kümenin 01 Mart 2019 ile 01 Şubat 2022 tarihleri arasındaki kaynak kullanım verilerinin kullanılarak kümelerin bellek kaynak kullanım değerlerinin tahmin edilmesi hedeflenmiştir. İlk olarak kaynak kullanımlarının zaman serisi için vROPS isimli platform aracılığı ile her günün altışar saatlik zaman dilimine ait raporlar oluşturularak veri seti hazırlanmıştır. Hazırlanan veri seti, zaman serisi analizine uygun hale getirilmesi için birtakım ön işleme aşamalarından geçirilmiştir. 2 farklı veri ayırma yöntemi kullanılarak istatistiksel yaklaşımlardan ARIMA ve VAR modelleri, derin öğrenme yaklaşımlarından RNN, LSTM, BiLSTM ve BiLSTM-RNN modelleri oluşturulmuştur. Bu modeller kullanılarak kaynak kullanım tahminleri üzerindeki her bir modelin başarısı ayrı ayrı değerlendirilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Zaman serilerinin analizi finans, sağlık, iletişim, endüstri, hava tahmini gibi birçok alanda yaygın olarak kullanılmaktadır. Bu nedenle zaman serisi tahmini alanında literatürde çok çeşitli çalışmalar sunulmuştur. Bu tez kapsamında ise zaman serisi analizinin hem çeşitli alanlarında hem de kaynak kullanım tahmini alanında literatürde sunulan çalışmalar incelenmiş ve bir takım yenilikçi yaklaşımlarda bulunulmuştur.

Malhotra ve arkadaşlarının 2015 yılında yayınladıkları çalışmada yığınlanmış LSTM mimarisi ile zaman serisi modeli oluşturmak ve bu model ile tahmin edilen değerlerin gerçek değerlerden çok farklı olduğu bölgelerin tespit edilmesi hedeflenmiştir [11]. Sunulan yaklaşım, dört gerçek dünya veri kümesi işlemiştir. Bunlardan "çoklu sensör motor verileri" olarak adlandırılan veri seti kamuya açık değilken; elektrokardiyogramlar, "Space Shuttle Marotta Valve" zaman serisi ve "Power Demand" veri kümeleri açık kaynaklıdır. Çalışmada tahminlerin her biri ve tahmini yapılan gerçek değerlerin her biri için bir tahmin hatası hesaplanmıştır. Ardından hesaplanan hatalar çok değişkenli bir gauss dağılımında kullanılarak her gözleme bir olasılık değeri ($p(t)$) atanmıştır. Olasılığı $p(t) < \tau$ olan her gözlem, anormallik olarak kabul edilmiştir. Deneysel sonuçlar, daha basit RNN tabanlı uygulamalara kıyasla uzun vadeli bağımlılıkları yakalamada önerilen yığınlanmış LSTM yaklaşımının daha iyi sonuçlar verdiğini göstermiştir.

Ellefsen ve arkadaşlarının 2019 yılında sundukları çalışmada yarı denetimli öğrenme temelli bir model kullanılarak "kalan faydalı ömür" tahmini gerçekleştirilmiştir [12]. Önerilen yarı denetimli derin mimari yapısı 5 katmandan oluşmaktadır. İlk katmanda sınırlı Boltzman makineleri (RBM), ikinci ve üçüncü katmanda LSTM, dördüncü katmanda ileri beslemeli sinir ağı (FNN) ve son katmanda hata hesaplamalarını işlemek ve kalan faydalı ömür tahminlerini gerçekleştirmek için bir çıktı katmanı eklenmiştir. Geliştirilen modelin eğitimi esnasında çeşitli hiper parametrelerin ayarlanması için bir genetik algoritma yaklaşımı uygulanmıştır ve NASA tarafından sunulan C-MAPSS [13] veri kümesinde test edilmiştir. Çalışmada bu yaklaşım hem eğitim verileri tamamen etiketlenerek hem de etiketli eğitim verilerinin sayısı azaltılarak test edilmiştir. Önerilen yarı denetimli öğrenme modelinin performansını değerlendirmek için RMSE kullanılmıştır. Sonuç olarak çalışmada

sunulan yarı denetimli öğrenme modeli kalan faydalı ömür tahmini görevlerinde, LSTM ve CNN kullanılan diğer çalışmalara kıyasla daha iyi sonuç verdiği gözlemlenmiştir.

Wang ve arkadaşlarının 2018 yılında yayınladıkları çalışmada kalan faydalı ömür tahmini için BiLSTM ağı ile yeni bir veri odaklı yaklaşım önerilmiştir [14]. Değerlendirme yöntemi olarak RMSE kullanılmıştır. Önerilen yöntem NASA tarafından sunulan C-MAPSS [13] veri kümesinde test edilmiştir. Elde edilen deneysel sonuçlarda LSTM, MLP, SVR, RVR, CNN ve LSTM modellerine kıyaslama geliştirilen modelin daha başarılı kalan faydalı ömür tahmini yaptığı gözlemlenmiştir.

Kızrak ve arkadaşının 2019 yılında sundukları çalışmada derin öğrenme modellerinden LSTM kullanılarak uçak motorlarının kalan yaşam ömürlerinin (RUL) tahmin edilmesi hedeflenmiştir [15]. Çalışmada NASA tarafından oluşturulan CMAPSS [13] veri kümesi kullanılmıştır. Test verileri RUL değeri 150'den küçük olan motorlar bakıma girmeli, diğerleri ise sağlıklı şekilde elle etiketlenerek problemin bir sınıflandırma problemine dönüştürülmesi sağlanmıştır. Geliştirilen model 3 katmanlı bir LSTM yapısından ve bir sönümleme katmanından oluşmaktadır. İlk iki LSTM katmanı 100 LSTM birimi, son katman ise 75 LSTM birimi içermektedir. Sönümleme işlemi için %50 değeri kullanılmıştır. Kayıp hesabının ölçümünde, çıktı değeri 0 ile 1 arasında olasılık değeri üreten ikili çapraz entropi kullanılarak geliştirilen modelin performansının değerlendirilebilmesi sağlanmıştır. Buna ek olarak doğruluk, hassasiyet, hatırlama ve F1-skoru metrikleri de hesaplanarak değerlendirmeye alınmıştır. Ayrıca, LSTM'nin regresyon başarısının ölçümü için eğitim verilerinden hesaplanan RUL değeri ile LSTM çıkışında hesaplanan değer arasındaki ortalama mutlak hata (OMH) hesaplanmıştır. Geliştirilen modelin başarısının değerlendirilebilmesi aynı veri kümesi farklı öğrenme metotları ile test edilmiştir. Önerilen model lojistik regresyon ve MLP yöntemlerine kıyasla daha başarılı sonuç üretmiştir.

Jain ve Satish'in 2009 yılında yayınladıkları çalışmada bir sonraki günün elektrik güç tüketiminin tahmini için kümeleme tabanlı yapay sinir ağı kullanan bir kısa vadeli yük tahmin modeli geliştirmişlerdir [16]. Tahmin için kullandıkları giriş parametreleri yük, sıcaklık ve haftanın günü bilgileridir. Eğitim ve test desenleri için her güne ait ortalama yük değeri hesaplanır. Oluşturulan desenler, test desenlerinin günlük ortalama yükü işe eğitim desenlerinin günlük ortalama yükü arasında belirlenen bir

eşik deęerine g re k melenir.  nerilen model k meleme olmadan yapılan tekniklerle kıyaslanmıř ve k meleme teknięinin daha iyi sonu lar verdięi g zlemlenmiřtir.

Bi ve arkadaşları 2021 yılında sundukları  alıřmada zaman serisi analizi i in entegre bir derin  ğrenme y ntemi tasarlamıřlardır [17].  alıřmalarında bulut sistemleri i in iř y k  ve kaynak kullanımlarının tahmini i in BiLSTM ve GridLSTM modelleri entegre edilerek BG-LSTM olarak adlandırdıkları bir model  nermiřlerdir.  alıřmada standart sapmanın azaltılması amacı ile logaritmik bir iřlem uygulanır. G r lt  ve aykırı deęerler ise g   l  bir filtre ile kaldırılmıřtır. Verilere normalleřtirme iřlemi uygulandıktan sonra  nerilen y ntemin tahmin doęruluęu "Google Cluster Trace" [19] veri setinde test edilmiřtir.  nerilen yaklařımın yaygın olarak benimsenen ARIMA ve LSTM gibi y ntemlere kıyasla daha iyi sonu  vermiřtir.

Mehmood ve arkadaşlarının 2018 yılında yayınladıkları  alıřmada topluluk tabanlı iř y k  tahmin mekanizması geliřtirilmiřtir [18].  nerilen y ntem 3 mod lden oluřmaktadır. Bunlar kaynak y neticisi, tahmin birimi ve kaynak ayırıcısıdır. Kaynak y neticisi birimi kaynak kullanım bilgilerinin kaydedildięi bir veri tabanı iken, tahmin mod l  iř y k  tahmininin yapıldıęı birimdir.  alıřmada "Google Cluster Trace" [19] veri seti kullanılmıřtır.  nerilen topluluk tabanlı tahmin mekanizmasında 2 seviyeli  ğrenme mekanizması geliřtirilmiřtir. İlk seviye temel  ğrenicidir ve k-en yakın komřu ve karar aęacı uygulanmıřtır. İkinci seviye meta  ğrenicidir. İlk seviyede temel  ğreniciler ile eęitim ger ekleřtirilmiřtir ardından ikinci seviyede testler ger ekleřtirilmiřtir. İkinci seviye, ilk seviyedeki  ğrenicilerin  ıktılarını birleřtirir, sınıflandırma sonucunu iyileřtirmek i in hataları d zeltir. Deęerlendirme metodu olarak F- l  m ve duyarlılık kullanılırken tahmin hatasını  l mek i in RMSE kullanılmıřtır.  nerilen topluluk tabanlı mekanizma sinir aęı, karar aęacı, SVM gibi modellere kıyasla daha bařarılı sonu lar  retmiřtir.

Borkowski ve arkadaşları 2016 yılında g rev bařına ve kaynak bařına bulut kaynak kullanımlarını tahmin etmek i in bir yaklařım sunmuřtur [20]. Ge miř g rev y r tme verileri kullanılarak makine  ğrenmesi tabanlı tahmin modelleri ger ekleřtirilmiřtir.  nerilen tahmin modelini deęerlendirebilmek i in Travis CI ve Github  zerinden kapsamlı bir veri seti oluřturulmuřtur. Deęerlendirme y ntemi olarak yapay sinir aęı kullanılmıřtır. Deęerlendirme metodu olarak RMSE kullanılmıřtır.  alıřmada ařırı uyum problemi yařandıęı g zlemlense de kabul edilebilir bir performans sergiledięi g zlemlenmiřtir.

Kumar ve arkadaşları, 2017 yılında yaptıkları çalışmada bulut veri merkezleri için iş yükü tahmin modeli geliştirmeyi hedeflemişlerdir [9]. Çalışmalarında NASA, Calgary ve Saskatchewan sunucularının sırasıyla 2 aylık, bir yıllık ve 7 aylık HTTP günlüklerini (*log*) inceleyerek 3 farklı veri seti oluşturulmuşlardır. Çalışmada önerilen modelde, tahmin biriminin çıktısı, veri merkezinin mevcut durumunu saklayan “Kaynak Yönetimi” birimine aktarılmıştır. Ön işleme adımında, veriler (0,1) aralığında normalleştirilmiş ve eğitim penceresi (*training window*) boyutuna göre yeniden ayarlanmıştır. Eğitim penceresi ise bir sonraki modeli tahmin etmek için kullanılan bir dizi model olarak tanımlanmıştır. Model, 4 alt modülden oluşmaktadır. Birinci modül, önceki bilgilerle birlikte eğitim penceresi (*tw*) girdi vektörünü ($x_1, x_2, \dots, x_{tw}$) almıştır. Başlangıçta bu bilgi 0 olarak atanmıştır. İkinci modül, ($x_2, x_3, \dots, x_{tw+1}$) girişini birinci modülün çıkışıyla birlikte almıştır. Benzer şekilde, üçüncü modül de, önceki birimin çıktısı ile birlikte girdi vektörlerini almıştır. Son olarak dördüncü modül tahmin değerini üretmiştir. Modelin doğruluğunu ölçmek için ortalama kök kare hatası (*RMSE*) hesaplanmıştır. Yapılan deneysel çalışmalarda amaç ise, ortalama kare hatasını en aza indirmektir.

Deepak Janardhanan ve Enda Barrett 2017 yılında, fiziksel makinelerin CPU kullanım tahminini ARIMA ve LSTM ağlarını kullanarak zaman serisi analizi ile gerçekleştirmişlerdir [4]. Çalışmanın amacı veri merkezlerindeki sistemlerin kaynakları daha verimli ve dinamik bir şekilde yönetmesinin sağlanmasıdır. Tahmin problemini çözmek için değişen mimariye sahip iki model oluşturulmuştur. İlk olarak, geleneksel bir yöntem olan ARIMA modeli kullanılırken, ikinci olarak LSTM modeli kullanılmıştır. Ardından ARIMA modellerine karşı LSTM modelinin kullanımının değerlendirilmesi yapılmıştır. LSTM modelinin, ARIMA modeline kıyasla daha az tahmin hatasına sahip olduğu gözlemlenmiştir. Çalışmada, Google tarafından yayınlanan bir kümeye ait fiziksel sunucuların 29 gününe ait kaynak kullanım verilerini içeren bir veri seti kullanılmıştır. “*Google Cloud Trace*” [19] isimli veri seti 5 farklı tablodan oluşmaktadır. 12453 fiziksel makinenin 29 günlük iş yükü bilgisinden oluşur. Yapılan çalışma sonucunda, doğrusal olmayan veriler için LSTM modellerinin ARIMA modellerinden çok daha iyi öğrenme yeteneği olduğu ve daha tutarlı performans sergilediği gözlemlenmiştir.

Gupta ve arkadaşları, bir çok çalışmada kullanılan Google veri setini kullanarak çok değişkenli zaman serilerinde LSTM ve BiLSTM modelleri geliştirerek, iş yükü kaynak kullanım tahmin modeli geliştirmişlerdir [21]. Çalışmalarında ilk olarak bulut

kaynak kullanımının gelecek zaman adımlarındaki tahmini için çok değişkenli zaman serileri için bir LSTM modeli geliştirilmiştir. İkinci olarak, kaynak kullanımının gelecekteki tahmini için ileri ve geri tahmin ağlarının etkisini analiz eden BiLSTM modeli geliştirilmiştir. Gupta ve arkadaşları LSTM ağlarının zaman serilerine dayalı kaynak kullanım verilerindeki uzun menzilli bağımlılıkları iyi modellediğini gözlemlenmiştir. Ayrıca, LSTM ve BiLSTM modellerinin önerilen çok değişkenli uzantıları, tek değişkenli modellerden daha iyi tahminler ürettiği bildirilmiştir.

Zhang ve arkadaşları 2016 yılında, bulut sistemlerinin iş yükü tahminini gerçekleştirebilmek için yine aynı “Google Cloud Trace” [19] veri setinin CPU ve RAM metriklerini kullanarak RNN tabanlı bir yaklaşım önermişlerdir [22]. Çalışmalarında, veriler arasında güçlü bir doğrusal ilişki olma ihtimaline karşı yapay sinir ağının öğrenme yeteneğinin zayıf olacağını dikkate alarak, diferansiyel dönüşüm kullanarak birleştirilmiş verilerin doğrusal derecesini azaltmayı hedeflemişlerdir. Çalışmada, RNN modelinin en uygun parametrelerini bulabilmek için dikey deney tasarımı yapılmıştır (OED: *Orthogonal Experimental Design*). OED, çok düzeyli ve faktörlü problemleri çözmek için bir tasarım yöntemidir. Deneysel hataların hesaplanabilmesi için Ortalama Kare Hatası (MSE) kullanılmıştır. Çalışma sonucunda CPU kaynak tahmini için 3 giriş nodu, 3 gizli nod, 0.01 öğrenme derece olarak ayarlandığı zaman en yüksek doğruluk değerine sahip olmuştur, RAM kaynak tahmini için 3 giriş nodu, 5 gizli nod, 0.005 öğrenme derece olarak ayarlandığı zaman en yüksek doğruluk değerine sahip olmuştur.

Gao ve arkadaşlarının 2020 yılında yayınladıkları çalışmada istatistiksel, makine öğrenimi ve derin öğrenme yaklaşımlarının CPU kullanımı tahmin doğruluğu üzerinde deneysel karşılaştırmalar sunulmuştur [23]. Çalışmada, giriş ve tahmin veri noktaları arasında “m” zaman noktası aralığı tutan “m-boşluk tahmini” yöntemi önerilmiştir. “Google Cloud Trace” [19] veri setinde her bir görev için ilişkili 5 dakikalık kullanım raporlama süresi içinden rastgele seçilen 1 saniyelik CPU/RAM kullanım örneği alınmıştır. İstatistiksel yaklaşımlardan ARIMA, makine öğrenimi yaklaşımlarından SVR (Destek Vektör Regresyonu) ve BRR (*Bayesian Ridge Regresyon*), derin öğrenme yaklaşımlarını temsil etmek için ise LSTM seçilmiştir. Doğruluk performansını göstermek için kümülatif dağılım fonksiyonu (CDF) kullanılmıştır. Deneysel sonuçlar CDF bazlı model sıralamasını $ARIMA \approx LSTM < BRR$ olarak göstermiştir.

Malik ve arkadaşlarının 2022 yılında yayınladıkları makalede, bulut veri merkezlerindeki fazla ve eksik tedarik sorunlarının üstesinden gelmek için kaynak kullanımı tahmini hedeflenmiştir [24]. Bu çalışmada, Genetik Algoritma (GA) ve Parçacık Sürü Optimizasyonu (PSO) ile Fonksiyonel Bağlantı Sinir Ağı (FLNN) kullanarak çoklu kaynak kullanımı tahminine odaklanılmaktadır. Geliştirilen yeni model, Hibrit GA-PSO algoritması olarak isimlendirilmiştir. İlk olarak, sinir ağını eğitmek için hibrit model kullanılmış ve daha sonra sinir ağı, kaynak kullanımı tahmini için kullanılmıştır. Önerilen teknik, “Google Cloud Trace” [19] veri seti üzerinde değerlendirilmiştir. Deneysel hataların hesaplanabilmesi için Ortalama Mutlak Hata (MAE) kullanılmıştır. Deneysel sonuçlar, önerilen modelin geleneksel tekniklere kıyasla daha iyi doğruluk sağladığını göstermiştir.

Al-Asaly ve arkadaşlarının 2022 yılında yayınladıkları çalışmada gelecek dönem CPU kullanım talebini tahmin etmek amacıyla PlanetLab tarafından sunulan CPU kullanım verilerini kullanarak, difüzyon evrişimli tekrarlayan sinir ağı (DCRNN) modeli geliştirmişlerdir [25]. Geliştirdikleri DCRNN modelinin, geleneksel derin öğrenme modellerine kıyasla bulut bilişim sistemlerinde mevcut olan tutarsız ve doğrusal olmayan iş yükünün gelecek kaynak kullanım tahminini zorlaştırması problemini çözmede daha başarılı olduğunu öne sürmüşlerdir. Önerilen derin öğrenme modelinin amacı tahmin doğruluğunu iyileştirmektir. Tahmin doğruluğunu değerlendirebilmek için kök ortalama kare hatası ve ortalama mutlak yüzde hatası kullanılmıştır. Önerilen model ile 0,18'lik bir ortalama mutlak yüzde hatası ve 2,40'lık kök-ortalama kare hatası elde edilmiştir. Elde edilen deneysel sonuçlara göre, önerilen yaklaşımın diğer derin öğrenme modellerinden daha iyi performans gösterdiği gözlemlenmiştir.

Anupama ve arkadaşlarının 2021 yılında yayınladıkları makalede hem mevsimsel hem de mevsimsel olmayan iş yüklerini başarılı tahmin edebilmek için makine öğrenme ve istatistiksel yöntemleri kullanarak hibrit bir model geliştirmişlerdir [26]. Geliştirilen modelde mevsimsel verilerin başarılı tahmini için SARIMA modeli, mevsimsel olmayan verilerin tahmini için ise LSTM veya ARIMA modeli kullanılmıştır. Çalışmada Bitbrains veri merkezinin gerçek kaynak kullanım verileri olan fastStorage kullanılmıştır. Veri seti günlük, saatlik ve dakikalık iş yükü kullanımlarını içeren farklı zaman aralıklarına sahiptir. Çalışmada geliştirilen model bellek kullanımı için 1,52 CPU kullanımı için 0.49 MAPE skoru ile tahmin gerçekleştirmiştir. Önerilen yöntemin sonuçları hem mevsimsel hem de düzensiz iş

yükü modellerini başarılı tahmin edebilmiştir ve LSTM ağınnn ARIMA modelinden daha iyi tahmin doğruluğuna sahip olduğunu göstermiştir.

Bao ve arkadaşlarının 2023 yılında bulut iş yükü tahmini için yayınladıkları çalışmada "Uyarlanabilir Model Madenciliği" (APM) tabanlı bir model önermişlerdir [27]. Çalışmalarında Alibaba ve Google'a ait bulut veri merkezlerinden alınan iş yükü verilerini içeren bir veri seti kullanmışlardır. APM modelinin iş yükü tahminindeki başarısını değerlendirebilmek için çeşitlik tahmin aralıkları ile birtakım deneyler gerçekleştirmişlerdir. Belirli bir iş yükü için, gelecekteki bir iş yükünü tahmin etmek için hataya dayalı bir ağırlık toplama yöntemi de geliştirmişlerdir. Gerçekleştirilen deneyler sonucunda önerilen APM modelinin klasik tahmin metotlarına kıyasla %19,62 daha yüksek tahmin başarısı gösterdiği gözlemlenmiştir.

Saxena ve arkadaşlarının 2020 yılında gerçekleştirdikleri çalışmada da bulut veri merkezlerinin esnekliği ve ölçeklenebilirliğini sağlamak amacıyla iş yükü tahmini gerçekleştirmek amaçlanmıştır [28]. Geliştirilen tahmin modelinde Otomatik Adaptif Diferansiyel Evrim (AADE) algoritması uygulanmıştır. Çalışmada NASA ve Saskatchewan HTTP loglarından oluşan iki ayrı veri seti kullanılmıştır. Önerilen AADE algoritması, geri yayılım (BP) ve kendi kendine uyumlu diferansiyel evrim (SaDE) tabanlı tahmin yaklaşımlarına kıyasla sırasıyla %97,4 ve %94,8'e varan doğruluk artışı sağlamıştır. Ek olarak önerilen modelin yakınsama hızı, BP ve SaDE tabanlı tahmin yaklaşımlarından 2-10 kat daha hızlı olduğu gözlemlenmiştir.

2022 yılında Xu ve arkadaşları tarafından yayınlanan çalışmada bulut veri merkezlerinde iş yükü tahmini için etkili denetimli öğrenmeye dayalı derin sinir ağı yaklaşımı (esDNN) önerilmiştir [29]. Gerçekleştirilen çalışmanın ilk aşamasında çok değişkenli veri setinin denetimli öğrenme zaman serisine dönüştürülebilmesi için kayan pencere kullanılmıştır. Sonrasında ise tahmin doğruluğunun sağlanması için kapalı tekrarlayan birim (GRU) uygulanmıştır. Çalışmada Alibaba ve Google bulut veri merkezlerinden elde edilen gerçek verilere dayalı iki ayrı veri seti kullanılmıştır. Geliştirilen modelin performansının değerlendirilebilmesi için ortalama kare hatası kullanılmıştır. esDNN modeli, yalnızca GRU kullanan yaklaşımlara kıyasla ortalama kare hatalarını %15 oranında azaltmıştır.

2018 yılında Yu arkadaşları, bulut sistemlerinde daha iyi kaynak yönetimi sağlamak amaçlı çalışan uygulamaların iş yüklerini doğru bir şekilde tahmin etmeye yönelik bir çalışma gerçekleştirmişlerdir [30]. Çalışmalarında regresyon tabanlı modeller ile tahmini zor olan yoğun iş yüklerine sahip sistemler için yeni görevlerin iş

yükü tahminini yapmayı hedefleyen, büyük bir görev havuzunun iş yüklerini içeren verilerin kullanıldığı "iş havuzu tabanlı" bir model geliştirmişlerdir. Bunun için kümeleme tabanlı bir öğrenme yaklaşımı sunmuşlardır. Bu yaklaşımda, ilk olarak iş havuzundaki görevler, iş yüklerine göre kümelenir. Her kümenin özelliklerini öğrenmek için bir sinir ağı kullanılır. Yeni bir görev geldiğinde, kendisine en uygun olan kümeye atanabilmesi için ilk iş yükü modeli ve gönderme (submission) parametreleri kullanılır. Bu adımın ardından, yeni görevin gelecek dönem iş yükü tahmini için sinir ağı kullanılır. Bu yaklaşıma ek olarak kümelemeye dayalı olmayan bir model daha gerçekleştirmişlerdir. Ancak elde ettikleri deneysel sonuçlar, küme yaklaşımının daha doğru iş yükü tahmini yaptığını göstermiştir.

Bulut veri merkezlerinde kaynak kullanım tahminlerine yönelik geliştirilen çalışmalar incelendiğinde, [9] nolu çalışmada bulut veri merkezleri için Nasa, Calgary ve Saskatchewan sunucularının logları kullanılmıştır ve geliştirilen tahmin modelleri sunucuya özeldir. [28] nolu çalışmada ise Nasa ve Saskatchewan sunucularına ait veriler kullanılmıştır. [4], [17,18] ve [21-24] nolu çalışmalarda Google Cluster Trace veri seti kullanılmıştır. "Google Cloud Trace" isimli veri seti 5 farklı tablodan oluşmaktadır. 12453 fiziksel makinenin 29 günlük iş yükü bilgisinden oluşur. Bu veriseti ile gerçekleştirilen tahmin çalışmaları, her bir fiziksel makineye özeldir. [26] nolu çalışmada Bitbrains veri merkezinin gerçek kaynak kullanım verileri olan fastStorage kullanılmıştır. Veri seti sunucu bazında günlük, saatlik ve dakikalık iş yükü kullanımlarını içeren farklı zaman aralıklarına sahiptir. [27] ve [29] nolu çalışmada Alibaba ve Google'a ait bulut veri merkezlerinden alınan iş yükü verilerini içeren bir veri seti kullanmışlardır. Literatürde taraması sonuçlarına göre yapılan kaynak kullanım tahmin çalışmaları sunucu özelinde gerçekleştirilmiştir. Tez çalışması kapsamında gerçekleştirdiğimiz çalışmada ise, birden fazla fiziksel sunucunun birleşiminden oluşan kümelere yönelik kaynak kullanım tahmini yapılmıştır. Bu yönü ile özgün bir çalışmadır.

3. ZAMAN SERİSİ ANALİZİ

3.1. Zaman Serisinin Bileşenleri

Eşit zaman aralıklarında kronolojik sıraya göre kesikli biçimde tutulan zaman serisi verileri, bir t anında kayda alınmış gözlemler serisidir. Zaman serileri, ayrık ve sürekli zaman serileri olarak ele alınabilmektedir. Gözlemlenen veriler sürekli bir zaman aralığında ($T_0 = [0,1]$) kaydedilerek elde edilmiş ise sürekli zaman serisini temsil ederken; saatlik, günlük, aylık vb. gibi sabit zaman aralıklarında yapılan gözlemler ayrık zaman serisini temsil eder. Tez çalışması kapsamında ayrık zaman serileri üzerinde deneysel çalışmalar gerçekleştirilmiştir [31], [32].

Zaman serileri 4 temel bileşenden oluşur. Bu bileşenleri aşağıdaki şekilde açıklamak mümkündür:

- Eğilim (Trend): Zaman serisine ait bir değişkenin dönem içerisinde gösterdiği artışa ya da azalışa denir. Zaman serisinde trend, doğrusal ya da eğrisel olabilir.
- Mevsimsel Değişim (Seasonal Variation): Sistematiik bir döngüsellik içeren mevsimsel bileşenlerde, periyotlar sabit uzunluğa sabittir. Aylık, çeyreklik dönemlere ait değişimler örnek olarak verilebilir.
- Döngüsel Değişim (Cyclic Variation): Bir yıl harici periyotlara ait değişimdir. Döngüsel değişimlerde, mevsimsel değişimlerden farklı olarak periyot uzunları sabit değildir.
- Rasgele (Düzensiz) Dalgalanmalar (Irregular Fluctuations): Ortalama etrafında gerçekleşen hareket, formül ile ifade edilemiyorsa rasgele dalgalanmadır [32], [33].

Zaman serilerinde 2 farklı model tipi kullanılır. Bunlar toplamsal (*additive*) ve çarpımsal (*multiplicative*) modellerdir:

$$\text{Toplamsal Model: } y_t = T_t + S_t + C_t + I_t \quad (3.1)$$

$$\text{Çarpımsal Model: } y_t = T_t \times S_t \times C_t \times I_t \quad (3.2)$$

t anındaki zaman serisi bileşenleri, y_t : gözlem, T_t : eğilim, S_t : mevsimsel, C_t : döngüsel ve I_t düzensiz varyasyon olarak temsil edilir. Toplamsal modelde dört bileşen

birbirinden bağımsız olabilirken, çarpımsal modelde bileşenler birbirini etkileyebilmektedir [34].

3.2. Tek Değişkenli Zaman Serisi Modelleri

Tek değişkenli zaman serisi; aynı değişkenin geçmiş ve zaman sıralı gözlemlerine dayalı olarak, eşit zaman artışlarıyla ardışık olarak kaydedilen gözlemlerden oluşan bir zaman serisini ifade eder. Genellikle tek bir sayı sütunu olarak verilse de aslında zaman serisinde örtük bir değişkendir. Veriler eşit aralıklıysa, indekxin açıkça verilmesi gerekmez. Tek değişkenli tahminler halen günümüzde birçok alanda tahmin modellerinde kullanılmaktadır. Tek değişkenli zaman serisi tahminlerinin performansındaki iyileştirmeler operasyonel ve stratejik planlamalarla ilgili maliyetleri azaltmak için oldukça önemlidir [35], [36].

Tek değişkenli zaman serileri ile ilgili tahmin yöntemlerinin dayandığı üç önemli unsur bulunmaktadır:

- Tahmini yapılacak zaman serisi değişkeninin, gelecek zaman dönemlerinde değişmeyeceği varsayılır.
- Eşit zaman aralıklarının tahmin değerlerinden gelen sürekli olmayan zaman serilerine uygulanır.
- Zaman serisini oluşturan etkili unsurlar ve tesadüfi değişkenler çıkarılarak gelecek zaman adımlarına ait tahmin yapılması hedeflenir.

Trend, mevsimsel, artık ayrışmalar (*Trend, Seasonal, Residual Decompositions*), frekans tabanlı methodlar (*Frequency Based Methods*), otoregresif modeller (*Autoregressive (AR) Models*), hareketli ortalama (*Moving Average (MA) Models*) ve Box-Jenkins yaklaşımı (*Box-Jenkins Approach*) tek değişkenli zaman serilerinde kullanılan ortak yaklaşımlardır [5], [35].

Tez çalışmamızda odak noktamız, tek değişkenli zaman serisi analiz yöntemlerinden ziyade çok değişkenli zaman serisi analiz yöntemleri kullanarak tahmin modelleri geliştirmek olmuştur.

3.3. Çok Değişkenli Zaman Serisi Modelleri

Çok değişkenli zaman serisi modellerinde, iki veya daha fazla zaman serisi arasındaki neden-sonuç ilişkisinin tanımlanarak, tahmin etme ve kontrol amacıyla kullanılması hedeflenir. Çok değişkenli zaman serileri, kendi geçmiş değerleri ile birlikte başka değişkenler ve geçmiş değerlerine de bağlıdır. Değişkenler arasındaki nedensel ilişkiyi göz ardı eden tek bir değişkenin tahmin edilen değerinin, diğer değişkenlerden etkilendiği varsayılmaktadır. Zaman serisi bileşenleri arasındaki ilişkinin belirlenmesi, çok değişkenli zaman serisi analizinin temel amacıdır.

Çok değişkenli zaman serilerinde ki temel kavramlardan biri durağanlıktır ancak çok değişkenli bir zaman serileri lineer dönüşüm kullanılarak durağan hale getirilebilirler [5], [37].

Tez çalışmamızda kullanılan Vektör Otoregresif Seriler (VAR), çok değişkenli zaman serisi modellere örnek olarak verilebilir. Vektör Otoregresif modeli, Granger nedensellik testini baz alır. Eğer iki içsel değişken var ise hem kendi hem de diğer içsel değişkenin belli bir döneme kadar olan gecikmeli değerleri ile ilişkilendirilir [6].

3.4. Birim Kök Testleri

Zaman serisi analizlerinde temel süreçlerden birisi serinin durağanlığının tespitidir. Aşağıdaki özelliklere sahip bir Y_t serisi tanımlayalım:

$$E(Y_t) = \mu \quad (3.3)$$

$$\text{var}(Y_t) = Y_0 \quad (3.4)$$

$$\text{cov}(Y_t, Y_{t+k}) = Y_k \quad (3.5)$$

Zaman serisinin başlangıç noktasının t olduğunu varsayarsak ve başlangıç noktasının t 'den " $t+k$ "ye kaydırıldığını düşünürsek; eğer ki Y zaman serisi durağan bir seri ise Y_t ve Y_{t+k} serilerinin ortalama, varyans ve kovaryans değerleri aynı olmalıdır. Bu durumda $k=0$ olduğunda aşağıdaki eşitlik sağlanmalıdır:

$$\text{cov}(Y_t, Y_{t+0}) = \text{var}(Y_t) = \sigma^2 \quad (3.6)$$

Yani durağan zaman serilerinin ortalama, varyans ve kovaryans değerleri zamandan bağımsızdır.

Zaman serilerinin durağanlığının tespiti için biçimsel ve biçimsel olmayan yöntemler kullanılmaktadır. Tez çalışmamız kapsamında, kullanacağımız veri setinin durağanlığının tespiti için biçimsel yöntemlerden olan birim kök testlerinden “Genişletilmiş Dickey-Fuller Testi” kullanılmıştır. Birim kök testleri ise aşağıdaki şekilde listelenmektedir:

- Dickey-Fuller Testi
- Genişletilmiş Dickey-Fuller Testi
- Kwiatkowski-Phillips-Schmidt-Shin Testi
- Philips-Perron Testi
- DF-GLS Testi
- Zivot-Andrews Testi

3.4.1 Dickey-Fuller Testi

Dickey-Fuller testi, zaman serilerinin durağanlığının tespit edilmesi amacıyla 1979 yılında geliştirilen ilk birim kök testidir. Bir zaman serisinin durağanlığının tespitinde kullanılan en yaygın test Dickey-Fuller testidir. Bir zaman serisinin geçtiği süreci anlayabilmek için, serinin her bir zaman adımında aldığı değerin daha önceki zaman adımlarında aldığı değerlerle regresyonunun bulunması gerekmektedir. Dickey-Fuller testi kullanılarak serilerin durağan olup olmadıkları test edilir [5], [38].

Dickey-Fuller testinin açıklanması için “ $Y_t = \alpha_1 Y_{t-1} + u_t$ ” birinci mertebeden özbağımlı süreci ele alındığında, α_1 değişkenin 3 durumu mevcuttur:

- $|\alpha_1| < 1$ ise kök kararlıdır, seri durağandır.
- $|\alpha_1| = 1$ ise birim kök söz konusudur, seri durağan değildir.
- $|\alpha_1| > 1$ ise kök kararsızdır.

Dickey Fuller testinde kullanılan denklemler aşağıdaki şekildedir:

- Sabit terimsiz model: $\Delta Y_t = \rho Y_{t-1} + u_t$ (3.7)

- Sabit terimli model: $\Delta Y_t = \beta_0 + \rho Y_{t-1} + u_t$ (3.8)

- Sabit terimli ve trendli model: $\Delta Y_t = \beta_0 + \beta t + \rho Y_{t-1} + u_t$ (3.9)

Burada ρ parametresinin en küçük kareler tahmincisi normal dağılıma uygun değildir. Bu nedenle Dickey-Fuller testinin Monte Carlo uygulamasında ortaya çıkarılan τ (tau) istatistiği kullanılır. Dickey-Fuller testi için hipotezler ise şu şekildedir:

- $H_0: \rho = 0$ yani seri durağan değildir.
- $H_1: \rho < 0$ yani seri durağandır [38].

3.4.2 Genişletilmiş Dickey-Fuller Testi

Genişletilmiş Dickey-Fuller Testi, 1984 yılında Said ve Dickey tarafından geliştirilmiştir. Dickey Fuller testinin genişletilmiş versiyonudur. DF testinde hata teriminin (ε_t) sabit varyanslı ve otokorelasyonsuz olduğu varsayılmaktadır ancak DF testindeki hata terimi değişen varyans özelliği gösterebilir ya da özilintili olabilir. Böyle bir durumda ise DF testinden elde edilen τ istatistiği yanıltıcı olmaktadır. DF testinin aksine, ADF testinde hata terimi otokorelasyonlu olarak kabul edilir. ADF testinde DF testinde kullanılan sabitsiz-trendsiz, sabitli ve sabitli trendli denklemleri dikkate alınarak bağımlı değişken olan Y_t 'nin ilk farklarının bir yada daha fazla gecikmeli değeri modele bağımsız değişken olarak eklenerek, otokorelasyon sorununun ortadan kaldırılması hedeflenir. İlgili denklemler aşağıdaki şekilde gösterilebilmektedir:

- Sabitsiz-Trendsiz ADF denklemi:

$$\Delta Y_t = \rho Y_{t-1} + \beta_i \sum_{i=1}^m \Delta Y_t - i + u_t \quad (3.10)$$

- Sabitli ADF denklemi:

$$\Delta Y_t = \alpha_0 + \rho Y_{t-1} + \beta_i \sum_{i=1}^m \Delta Y_t - i + u_t \quad (3.11)$$

- Sabitli- Trendli ADF denklemi:

$$\Delta Y_t = \alpha_0 + \alpha_1 t + \rho Y_{t-1} + \beta_i \sum_{i=1}^m \Delta Y_t - i + u_t \quad (3.12)$$

ADF testlerinde gecikme sayısına karar vermek önemli bir konudur. Bunun için Schwarz (SIC) ve Akaike (AIC) bilgi kriterlerinden faydalanılır. Uygun gecikme sayısına karar vermek için, bu değerlerin en küçüğü seçilir ve AIC ile SIC değerlerinde artış meydana gelir.

Hipotezlerin değerlendirilmesi DF testlerinde olduğu gibidir. Eğer üç modelden en az birisi durağanlık testinden geçmiş, H_0 reddedilmiş ise serinin durağan olduğu söylenebilir [5], [38], [39].

3.5. Bilgi Kriterleri

3.5.1 Akaike Bilgi Kriteri

Akaike Bilgi Kriteri (AIC), geliştirilen modelin üretildiği verilerle uyumunu değerlendirebilmek amacıyla geliştirilen matematiksel bir yöntemdir. AIC esasında tahmin hatasının bir tahmin edicisidir ve her bir modelin kalitesini diğer modellerin her birine göre tahmin ederek model seçimi için bir araç sağlar. AIC değeri şu şekilde hesaplanır:

- Modeli oluşturmak için kullanılan bağımsız değişkenlerin sayısı
- Modelin maksimum olabilirlik tahmini

En az sayıda bağımsız değişken kullanarak, en yüksek varyasyonu açıklayan model, AIC'e göre en iyi modeldir. Geliştirilen bir modelin AIC değeri aşağıdaki gibi hesaplanır:

$$AIC = 2K - 2\ln(L) \quad (3.13)$$

Yukarıda tanımlanan formüle göre K değeri modeldeki tahmini parametre sayısını temsil ederken, L değeri model için olabilirlik fonksiyonunun maksimize edilmiş değeridir. Veriler için bir dizi aday model verildiğinde, minimum AIC değerine sahip olan model tercih edilir. AIC; modelin verilerle olan uyumunu ödüllendirirken, aşırı uyumu önlemek amaçlı tahmin edilen parametre sayısının artan işlevi olan bir ceza sistemine sahiptir [40], [41], [42].

Çalışma kapsamında, Vektör Otoregresif modellerinin değerlendirilmesinde AIC kullanılmıştır.

3.5.2 Bayes Bilgi Kriteri

Sonlu bir model seti arasından en uygun model seçimi için yaygın olarak kullanılan diğer bir bilgi kriteri de Bayes Bilgi Kriteridir (*BIC*). Kısmi olarak olabilirlik fonksiyonuna dayalıdır ve AIC ile yakın ilişkilidir fakat AIC' den farklı olarak iki rakip model için Bayes faktörünün bir tahmini olarak bir Bayes çerçevesi içinde türetilmişlerdir. Modelleri ayarlarken bağımsız parametre ekleyerek olasılığı artırmak mümkündür fakat bu fazla uyuma (*overfitting*) neden olacaktır. AIC'de olduğu gibi bu durumu önlemek için modeldeki parametre sayısına dayalı bir ceza terimi vardır fakat AIC' den farklı olarak BIC'de ceza süresi daha fazladır. Bununla birlikte herhangi bir maksimum olabilirlik tabanlı model grubuna oldukça geniş bir şekilde uygulanabilir. Her iki bilgi kriteride, veriler için en uygun modeli belirleme amacına sahiptir.

BIC matematiksel olarak aşağıdaki şekilde tanımlanmaktadır:

$$BIC = \ln(n)K - 2 \ln(L) \quad (3.14)$$

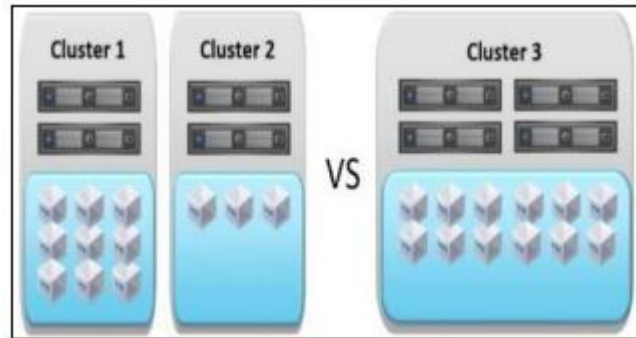
Yukarıda tanımlanan formüle göre n değeri toplam veri noktası sayısını, K değeri modeldeki tahmini parametre sayısını temsil ederken, L değeri model için olabilirlik fonksiyonunun maksimize edilmiş değeridir. Modeller, karşılık gelen BIC

değerleri kullanılarak test edilebilir. Model değerlendirme aşamasında BIC değerinin düşük olması, daha düşük ceza koşullarını ve buna bağlı olarak daha iyi bir modeli gösterir. AIC ve BIC için formül gösterimlerini dikkate aldığımızda tek fark, BIC' in AIC' in dikkate almadığı formüldeki gözlem sayısını dikkate almasıdır [43], [44].



4. YÖNTEM VE ARAÇLAR

Bulut bilişime hizmet eden veri merkezlerinde fiziksel sunucular kümelendirilerek yönetilmektedir. Fiziksel makinelerden oluşan bu kümeye (*cluster*) yeni bir fiziksel makine dahil edildiğinde, eklenen makinenin kaynakları kümenin kaynaklarına da dahil olmuş olur. Kümeler, içerdikleri tüm fiziksel sunucuların kaynaklarına sahip olduğundan bu kaynakları yönetirler. Şekil 4.1’ de küme mimarisinin fiziksel ve sanal makineler ile nasıl yönetildiği gösterilmektedir. Şekli incelediğimizde, siyah dikdörtgen bloklar fiziksel sunucuları temsil etmektedir. Beyaz küpler ise aynı kümede bulundukları fiziksel sunucuların üzerinde çalışan sanal sunucuları temsil etmektedir. Küme içerisindeki sunucuları kapsayan gri yapı ise şaseyi temsil etmektedir. Şase farklı fiziksel sunucuları bir araya getiren metal bir yapıdır ve birden fazla fiziksel makineyi, depolama ve çevre birimlerini bir araya getirerek tek bir fiziksel gövdede barındırılmasını sağlar. Bu durumda Küme1, 2 fiziksel makine ile 9 sanal makineden meydana gelirken Küme2, 2 fiziksel makine ile 3 sanal makineden meydana gelmiştir. Küme3 ise 4 fiziksel makine ile 12 sanal makineden oluşmaktadır [10], [44], [45].



Şekil 4.1: Küme mimarisi.

Şaselerdeki her bir fonksiyonel/fiziksel parça, güç kaynakları yedekli yapıda olsalar dahil, bu parçalardan herhangi birinde arıza meydana gelirse, şaseler arası bellek paylaşımı gerekebilir. Bu nedenle SRD yani servis yedekliliği arka plan programı, şaseler arası yedeklilik durumunu kontrol eder; koşulları izler ve gerekirse şaseler arasında bellek paylaşımı için bir geçiş gerçekleştirir. Böylece veri merkezlerinde şase yedekliliği sağlanmış olur. Örneğin bir X kümesi 2 şase, 18 fiziksel

sunucu içeriyorsa, X kümesinin bellek kullanım değeri %50'nin üstüne çıkması durumunda şase yedekliliğinin riske girdiği çıkarımı yapılabilir [10], [46].

Tez çalışmasının beklenen çıktısı ise bir kümenin gelecekteki bellek kaynak kullanım değerlerini tahmin edebilmektir. Bu sayede şase yedeklilik durumunun riske gireceği zaman dönemleri belirlenebilir. Yedekliliğin riske gireceği dönemlerin önceden tespit edilmesi ile yeni fiziksel sunucu satın alımı, mevcut fiziksel sunucular arası daha yüksek kapasiteli sunucular ile kümeler arası değişiklik yapmak gibi aksiyonların alınması sağlanabilir. Donanım satın alma süreçleri özellikle kamu kurumlarında genellikle 2 ila 4 ay arası bir zaman almaktadır. Bu durum göz önünde bulundurulduğunda ihtiyacın önceden tespit edilmesi ile sistemdeki kesinti riskinin en aza indirilmesinin yanı sıra yedeklilik riskinin önlenmesi ve ihtiyaç fazlası fiziksel makine alımının önlenmesi gibi oldukça kritik aksiyonların alınması sağlanabilir [10], [47].

4.1. VERİ SETİNİN OLUŞTURULMASI

Tez çalışması kapsamında, bellek kaynak kullanım değeri tahmin edilecek veri merkezindeki kümelere ait veriler yapay zekâ destekli bir servis sağlayan “vRealize Operations Manager” (vROPs) platformunun raporlama aracından elde edilmiştir. vROPs fiziksel makine ve barındırdıkları sanal makinelerin uygulama düzeyinde sunduğu ön görülerin izlenmesine yönelik bir bilgi teknolojileri operasyonu yönetim aracı olarak adlandırılmaktadır. vROPs, bulut sistemleri altyapısı yönetimi için, performans/kapasite iyileştirmeleri, sistem izleme/alarm oluşturma, sorun giderme/düzeltilme seçeneklerinin yanında çeşitli gösterge panelleri ile birlikte raporlama imkanları sağlamaktadır. [10], [48], [49].

Türkiye'nin en büyük altyapısına sahip bulut hizmetleri sağlayıcısına ait veri merkezlerinde barındırılan 53 adet küme için “01 Mart 2019” ile “01 Şubat 2022” tarihleri arasındaki veriler VROPS platformu üzerinden raporlanmıştır. Raporlar, tüm verilerin altışar saatlik aralıkları (00:00-06:00, 06:00-12:00, 12:00-18:00, 18:00-00:00) için oluşturulmuştur. Bu durumda her bir gün için 53 kümenin bellek kullanımına yönelik bilgiler içeren 4 ayrı rapor elde edilmiştir. Veri setinde 36 farklı öznitelik değeri mevcuttur. Buna, tarih “date time” özniteliği indeks olarak eklenmiş ve zaman serisi

formatına dönüştürülmüştür. Tablo 4.1, çalışmada kullanılan özniteliklerin listesini ve bu özniteliklerin ne anlama geldiğini göstermektedir [10].

Tablo 4.1: Öznitelikler.

ÖZNİTELİK	AÇIKLAMA
Küme İsmi	
Bellek Kullanımı (%)	Toplam kullanılabilir belleğin yüzdesi olarak şu anda kullanımda olan bellek.
Fiziksel Sunucu Başına VM Sayısı	
Toplam VM Sayısı	Gücü Açık Çalışan VM Sayısı
Toplam Fiziksel Sunucu Sayısı	Gücü Açık Çalışan Fiziksel Sunucu Sayısı
Çalışan VM Sayısı	Fiziksel sunucuda barındırılan toplam VM sayısı
Çalışan Fiziksel Sunucu Sayısı	Kümede barındırılan toplam fiziksel sunucu sayısı
Küme Kullanılabilirliği (%)	Kümede çalışan fiziksel sunucuların yüzdesi.
Çalışan Fiziksel Sunucu Başına Ortalama Çalışan VM Sayısı	
Geri Alınabilir Boşta VM'ler Bellek (KB)	
Bellek İş Yüğü (%)	
Bellek Kullanımı	kullanım düzeyi. VM'leri çalıştırmak için rezervasyonları, limitleri ve ek yükü dahil edilir.
Bellek Kullanımı / Kullanılabilirlik (%)	
Kullanılabilir Bellek	
Kullanılabilir Bellek Kapasitesi	VM'ler için vSphere HA ve diğer vSphere hizmetleri için yapılan rezervasyonlar
Bellek Takas Oranı (In)	
Toplam Bellek Kapasitesi	
Bellek Takas Oranı (Out)	Geçerli aralık sırasında belleğin etkin bellekten diske değiştirilme hızı.
Takas Edilebilir Bellek	Değiştirilen bellek miktarı.
Ortak Bellek Paylaşımı	Paylaşılan ortak bellek miktarı
Paylaşılan Bellek	Paylaşılan bellek miktarı
Rezerve Edilen Bellek Kapasitesi	
Tüm Tüketicilere Ayrılan Bellek(KB)	Tüm Powered On VM'lerde ve Pod'larda Ayrılan Bellek.
Tüm Güç Açık Tüketicilere Ayrılan Bellek (KB)	Tüm VM'lerde ve Pod'larda Ayrılan Bellek.
Maks. VM Bellek Çekişmesi (%)	Tüm VM'ler arasında Maks. VM Bellek Çekişmesi
Makine Talebi (KB)	KB cinsinden Bellek Makinesi Talebi.
Bellek Fiziksel Sunucu Kullanımı	Makine kullanımı.
Bellek Yığın	Yığın için ayrılan bellek miktarı.
Bellek Aktif Misafir	Aktif olarak kullanılan bellek miktarı.
Bellek Granted	Kullanılabilecek bellek miktarı.
Bellek ESX Sistem Kullanımı	Vmkernel ve ESX kullanıcı düzeyi hizmetleri tarafından bellek kullanımı.
Bellek Geçerli Fazla İşlem Oranı	Açık VM'lere ayrılan bellek, Bellek Toplam Kapasitesine bölünür.
Bellek Contention (%)	
Bellek Contention	Ortalama bellek çekişmesi.
Tüketilen Bellek	Açık VM'lere ayrılan bellek, Bellek Toplam Kapasitesine bölünür.
Sıkıştırılmış Bellek	

4.2. VERİ ÖNİŞLEME

Veri setindeki öznitelikler farklı veri tiplerinden oluşmaktadır ve özniteliklerin sahip olduğu değerler farklı ölçü birimlerine sahip olabilmektedir. Örneğin “Kullanılabilir Bellek” özniteliği, A kümesi için 1,11 Terabayt (TB) değerine sahip iken, B kümesi için 232,45 Gigabayt (GB) değerine sahip olabilmektedir. Bu durum kapasite metriklerinin nümerik kaydedilmesini ve aynı birimde kaydedilebilmesini engellemektedir. Bu durumun önüne geçebilmek için her bir metrikte en çok tekrarlayan ölçü birimi tespit edilerek, öznitelik değerleri aynı ölçü birimine dönüştürülmüştür. Ardından dönüşüm yapılan özniteliğin ismine, dönüşüm yapılan ölçü birimi eklenmiştir. Bu duruma örnek aşağıdaki şekilde gösterilebilir:

“Öznitelik 1” → “Öznitelik1 (Ölçü Birimi)”

Veri setinde bulunan diğer sayısal veriler ek bir dönüşüm işlemi uygulanmadan kullanılmıştır. Verinin okunduğu tarih bilgisi ise yıl, ay, gün, saat periyodu ve haftanın günü olarak ek öznitelik olarak etiketlenerek yeni bir öznitelik olarak veri setine eklenmiştir. Saat dilimi özniteliği, günün her bir periyodunu temsil edecek şekilde 0 ile 3 arasında; haftanın günü (*week_day_number*) özniteliği ise haftanın günlerini temsil edecek şekilde 0 ile 6 arasında etiketlenmiştir. Pazar gününü temsil eden “haftanın günü” özniteliği 6, salı günü için 1 olarak etiketlenmiştir. Burada pazar gününün, salı gününe kıyasla daha ağırlıklı bir değere sahip olması doğru olmayacaktır. Bu durumu engellemek için, “haftanın günü” özniteliğine “one-hot encoding” uygulanmıştır. Tablo 4.2’de “one-hot-encoding” uygulandıktan sonraki durumu göstermektedir [10].

Tablo 4.2: "Haftanın Günü" özniteliğine one-hot-encoding uygulaması.

Haftanın Günü_0	Haftanın Günü_1	Haftanın Günü_2	Haftanın Günü_3	Haftanın Günü_4	Haftanın Günü_5	Haftanın Günü_6
0	0	0	0	1	0	0

4.2.1 Boş Oluşan Raporlar ve Kapatılan Kümeler

vROPS üzerinden oluşturulan raporlar incelendiğinde, kaynak verisi toplanan bazı kümelerin zaman içerisinde kapatıldığı, bazı kümelerden ise bazı günlere ait altışar saatlik raporlar için veri alınmadığı görülmüştür. Çalışma kapsamında rapor oluşturulan tarih aralıklarında 58 adet kümeden 5'inin zaman içerisinde kapatıldığı tespit edilmiştir. Bu nedenle bu kümelere ait veriler veri setinden çıkarılmıştır.

Raporların oluşturulduğu tarih aralığında aktif olarak çalışmaya devam eden fakat bazı periyotlarda kaynak kullanım verileri raporlanamamış kümelerde ise enterpolasyon uygulanmıştır. Kayıp olan periyoda ait raporların enterpolasyon işlemini gerçekleştirmek için, bir önceki ve bir sonraki periyotlara ait raporlardaki verilerinin ortalaması alınacak şekilde hesaplama yapılmıştır. Mutlaka ki enterpolasyon sonucunda elde edilen verilerin, gerçek değerlere yakınlığı test edilmelidir. İnterpolasyon işleminin doğruluğunu test edebilmek için aynı işlem eksiksiz olan verilerde de test edilmiştir. Test sonucunda, elde edilen değerler ile gerçek değerler arasında %1-2'lik kadar fark olduğu tespit edilerek enterpolasyon işleminin güvenilirliği doğrulanmış oldu [10].

4.2.2 Eksik Değerlerin Doldurulması

Veri setinde yapılan incelemelerde özniteliklerden bazılarının raporlarda boş değer içerdiği gözlemlenmiştir. Aşağıdaki özniteliklere arıza olan veya altyapı çalışması yapılan saat aralığında boş değerler atandığından 0 ile doldurulmuştur [10]:

- Fiziksel Sunucu Başına VM Sayısı (*Summary|VM Count per Host*)
- Toplam VM Sayısı (*Summary|Total Number of VMs*)
- Toplam Fiziksel Sunucu Sayısı (*Summary|Total Number of Hosts*)
- Çalışan VM Sayısı (*Summary|Number of Running VMs*)
- Çalışan Fiziksel Sunucu Sayısı (*Summary|Number of Running Hosts*)

Doğrusal regresyon analizi, bir değişkenin değerini başka bir değişkenin değerine göre tahmin etmek için kullanılır. Aşağıdaki öznitelikler için ise doğrusal regresyon kullanılarak veri setindeki eksik bilgiler doldurulmuştur [10]:

- Küme Kullanılabilirliği (%) (*Summary|Cluster Availability (%)*)
- Maks. VM Bellek Çekişmesi (%) (*Memory|Max VM Memory Contention (%)*)
- Bellek | Geçerli Fazla İşlem Oranı (*Memory|Current Overcommit Ratio*)
- Bellek Çekişmesi (*Memory|Contention (%)*)

4.2.3 Düzgeleme

Çalışma kapsamında hazırlanan veri seti içinde birçok öznitelik mevcuttur ve bu özniteliklerin boyutları farklı olabilmektedir. Ancak boyutu büyük olan öznitelik değerleri, model eğitilirken daha büyük ağırlığa neden olabilir ve tahmin doğruluğunu o ekseninde kaydırabilir ve bu yaklaşım yanlıştır. Çünkü tahmin doğruluğu, ağırlığı küçük olan bir değer tarafında da olabilir. Bu nedenle veri setindeki tüm verilerin aynı ölçekte olabilmesi için model eğitim aşamasından önce düzgelemesi gerekmektedir. Çalışmada z-skor ve minimum-maksimum düzgeleme yöntemleri kullanılarak test edilmiştir [10], [50].

4.2.3.1 Z-Skor Düzgeleme

Z-skor düzgeleme yöntemi, istatistiksel yöntemlerden biridir. Veri setinde uç değerler mevcut ise bu değerler sonuçlara daha fazla etki yaparlar. Ancak bu değerlerin de diğer veriler gibi tahmin yapmak için geliştirilen modele yakın oranda katkı sağlaması gerekir. Z-skor düzgeleme yöntemi ile ilk olarak mevcut verilerin ortalaması ve standart sapması hesaplanır. Z-skor formülü kullanarak modelin tahmin için eğitilmesinde eş katkı yapması sağlanır. Z-skor formülü matematiksel olarak aşağıdaki şekilde gösterilmektedir [50]:

$$z = (X - \mu) / \sigma \quad (4.1)$$

Formül 4.1’de tanımlanan X değeri bir değişkene ait rastsal değişken, μ değeri verilerin ortalaması, σ ise X değişkeninin standart sapmasıdır. z ise düzgelemiş veridir.

4.2.3.2 Minimum-Maksimum Düzgeleme

Minimum-Maksimum düzgeleme yöntemi mühendislik problemlerinde kullanılan en yaygın yöntemlerden biridir. Bu yöntemde, veri setindeki orijinal serinin maksimum (MaxT) ve minimum (MinT) değerine dayalı olarak veri noktalarını [MinT, MaxT] aralığından [MinN, MaxN] aralığına doğrusal olarak dönüştürmektedir. Dönüşüm sonrasında veriler arasındaki ilişki korunur. Matematiksel olarak aşağıdaki şekilde gösterilmektedir [50], [51]:

$$x' = (X - \min(X)) / (\max(X) - \min(X)) \quad (4.2)$$

x' , düzgelemen veriye temsil etmektedir. Çalışmada z-skor ve min-maks düzgeleme yöntemleri ayrı ayrı test edilmiştir ve min-maks düzgeleme ile daha iyi sonuçlar elde edilmiştir.

4.3. MODEL GELİŞTİRME

Veri ön işleme aşaması tamamlandıktan sonra istatistiksel tahmin modelleri ve derin öğrenme metotları kullanılarak çeşitli modeller geliştirilmiştir. İstatistiksel tahmin modellerinden “Bütünleşik Otoregresif Hareketli Ortalama” (ARIMA) ve “Vektör Otoregresyon” (VAR) modelleri test edilmiştir. Derin öğrenme tekniklerinden ise “Yinelemeli Sinir Ağı” (RNN), “Uzun-Kısa Süreli Bellek” (LSTM), “Çift Yönlü Uzun-Kısa Süreli Bellek” (BiLSTM) ve BiLSTM-RNN modelleri geliştirilip test edilmiştir. Hiperparametreler, gizli katmandaki farklı eğitim dönemi ve nöron sayısı kombinasyonlarıyla her bir model için birden fazla ağı eğitilmesinden sonra ayarlanmıştır.

4.3.1 İstatistiksel Tahminleme Modelleri

İstatistiksel yaklaşımlar, zaman serisi analizi alanında iş yükünü tahmin etmede kullanılan popüler yöntemlerdendir. Zaman serisi modelleri farklı istatistiksel methodlar ile temsil edilebilmektedir. Literatürde bulunan bir çok çalışma da doğrusal zaman serisi modelleri olan Otoregresif (Autoregressive - AR) ve Hareketli Ortalama (Moving Average - MA) modelleri sıklıkla uygulanmaktadır. Bunun yanı sıra otoregresif ve

hareketli ortalama modellerini birleştiren, Otoregresif Hareketli Ortalama (*Autoregressive Moving Average - ARMA*) ve Otoregresif Bütünleşik Hareketli Ortalama (*Autoregressive Integrated Moving Average – ARIMA*) yöntemleri de sıklıkla kullanılmaktadır [32]. Ancak bu modeller, tek değişkenli zaman serisi modellerinde kullanılmaya uygundur. Tez çalışmamız kapsamında hazırlanan veri seti, çok değişkenli zaman serisi modelleri için uygundur. Bundan dolayı tek değişkenli otoregresif sürecin bir uzantısı olan ve çok değişkenli zaman serisi modellerinden biri olan Vektör Otoregresif modeller ile testler gerçekleştirilmiştir. Bunun yanı sıra, tek değişkenli zaman serisi performansını çok değişkenli zaman serisi modelleri ile karşılaştırmak adına ARIMA modeli de deneysel çalışmalarda kullanılmıştır.

4.3.1.1 Otoregresif Modeli

Otoregresif modeller, p 'ninci dereceye sahip otoregresif modeli yani AR(p) olarak tanımlanır ve aşağıdaki denklem ile ifade edilir:

$$x_t = c + \sum_{i=1}^p \phi_i x_{t-i} + \varepsilon_t \quad (4.3)$$

$\phi_1, \dots, \phi_n$ modelinde c sabit terimi temsil ederken, ε_t hata terimini ifade eder. c sabit terimi birçok çalışmada ihmal edilmektedir. AR(p) modeli ϕ_i parametrelerine dayanmaktadır ve parametreler Yule-Walker denklemleri ile hesaplanır:

$$\gamma_m = \sum_{k=1}^p \phi_k \gamma_{m-k} + \sigma_\varepsilon^2 \gamma_m \quad (4.4)$$

$M = 0 \dots p$ aralığındadır ve $p+1$ adet denklem oluşur. γ_m , Kronecker Delta Fonksiyonunu temsil eder. X 'in otokorelasyon fonksiyonu γ_m olarak ifade edilir. σ_ε ise girdi gürültü sürecinin standart hatasıdır [52].

4.3.1.2 Hareketli Ortalama Modeli

Hareketli ortalama modeli, q derecesine sahip hareketli ortalamalar modeli yani MA(q) olarak tanımlanır ve aşağıdaki denklem ile ifade edilir:

$$X_t = \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (4.5)$$

$\theta_1, \dots, \theta_q$ modelin parametrelerinde $\varepsilon_t, \varepsilon_{t-1}$ modelin hata terimlerini ifade eder. Bu modelde belirli bir zaman adımında, zaman serisi değişkeninin değeri q simgesi ise; daha önceki her bir zaman noktasında yapılan hataların, ağırlıklı olarak bileştirilmesi ile açıklanmaktadır [52].

4.3.1.3 Oto regresif Bütünleşik Hareketli Ortalama

Oto regresif Bütünleşik Hareketli Ortalama Modeli (ARIMA), tek değişkenli zaman serisi modellerinde bir değişkenin geçmiş değerlerini kullanarak gelecek dönem değerlerini tahmin etmek için kullanılır. Otomatik regresyon ve hareketli ortalama yaklaşımlarını kullanır.

Bir zaman serisinin özellikleri zamanla değişiyor ise, bu seri durağan olmayan bir seridir ve belli bir trende ve/veya mevsimsellik bileşenlerine sahiptir. ARIMA modeli, zaman serisi verilerinin durağan olduğunu varsayar ve zaman serisinin ortalama ve varyans değerleri sabit kalmalıdır. Bu nedenle zaman serisinin mevsimselliği kontrol edilmelidir ve modeli uygulamadan önce fark alma adımı uygulanarak zaman serisindeki durağan olmayan bileşenler kaldırılmalıdır. Fark alma adımı zaman serisi verilerinin daha kararlı hale gelmesini sağlar. ARIMA modeli matematiksel olarak aşağıdaki şekilde ifade edilir.

$$y_t = c_t + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (4.6)$$

ARIMA modeli p, d ve q olmak üzere 3 parametre içermektedir. Parametrelerin her biri şu şekilde açıklanabilir:

- p : “Otomatik Regresyon”u temsil eder. ARIMA modelinin tahmin yaparken geçmişte kaç değer kullandığını ifade eder.
- d : Durağanlığı temsil eder. Eğer zaman serisi durağan değil ise, durağan olmayan bileşenleri çıkarmak için yapılan farklılaştırma dönüşümlerinin sayısını ifade eder.
- q : “Hareketli Ortalama”yı temsil eder. Modelin hata bileşenin gecikmesini ifade eder [7], [23], [53].

Tez çalışması kapsamında veri setindeki her bir kümenin “Bellek Kullanımı (%)” değerinin gelecek dönemler için tespiti için ARIMA modeli uygulanmıştır. Model uygulanmadan önce ilk olarak veri setinin durağanlığının tespit edilmesi gerekmektedir. Genişletilmiş Dickey-Fuller (ADF) testi ile veri setinin durağanlığının değerlendirilmesi için istatistiksel bir kontrol yapılmıştır. Eğer ADF testi ile veri setinin durağan olmadığı tespit edilirse, durağan hale getirmek için fark alma adımı uygulanması gerekmektedir. Tablo 4.3.’de gösterildiği üzere, gerçekleştirilen ADF testinin sonucuna göre veri setinin durağan olduğu görülmektedir.

Tablo 4.3: Genişletilmiş Dickey-Fuller Testi Sonucu.

Memory Usage (%)	P-Value = 0.0 => Stationary.
Summary VM Count per Host	P-Value = 0.0 => Stationary.
Summary Total Number of VMs	P-Value = 0.0 => Stationary.
Summary Total Number of Hosts	P-Value = 0.0 => Stationary.
Summary Number of Running VMs	P-Value = 0.0 => Stationary.
Summary Number of Running Hosts	P-Value = 0.0 => Stationary.
Summary Cluster Availability (%)	P-Value = 0.0 => Stationary.
Summary Average Running VM Count per Running Host	P-Value = 0.0 => Stationary.
Memory Workload (%)	P-Value = 0.0 => Stationary.
Memory Utilization (GB)	P-Value = 0.0 => Stationary.
Memory Usage / Usable (%)	P-Value = 0.0 => Stationary.
Memory Usable Memory (TB)	P-Value = 0.0 => Stationary.
Memory Usable Capacity (TB)	P-Value = 0.0 => Stationary.
Memory Total Capacity (TB)	P-Value = 0.0 => Stationary.
Memory Swap Out (GB)	P-Value = 0.0 => Stationary.
Memory Shared Common (GB)	P-Value = 0.0 => Stationary.
Memory Shared (GB)	P-Value = 0.0 => Stationary.
Memory Reserved Capacity (GB)	P-Value = 0.0 => Stationary.
Memory Memory Allocated on all Powered On Consumers (GB)	P-Value = 0.0 => Stationary.
Memory Memory Allocated on all Consumers (GB)	P-Value = 0.0 => Stationary.
Memory Max VM Memory Contention (%)	P-Value = 0.0 => Stationary.
Memory Machine Demand (GB)	P-Value = 0.0 => Stationary.
Memory Host Usage (GB)	P-Value = 0.0 => Stationary.
Memory Heap (MB)	P-Value = 0.0 => Stationary.
Memory Guest Active (GB)	P-Value = 0.0 => Stationary.
Memory Granted (TB)	P-Value = 0.0 => Stationary.
Memory ESX System Usage (GB)	P-Value = 0.0 => Stationary.
Memory Current Overcommit Ratio	P-Value = 0.0 => Stationary.
Memory Contention (%)	P-Value = 0.0 => Stationary.
Memory Contention (Byte)	P-Value = 0.0 => Stationary.
Memory Consumed (GB)	P-Value = 0.0 => Stationary.

Tablo 4.3: Devam.

Memory Compressed (GB)	P-Value = 0.0 => Stationary.
year	P-Value = 0.0 => Stationary.
month	P-Value = 0.0 => Stationary.
day	P-Value = 0.0 => Stationary.
hour	P-Value = 0.0 => Stationary.
weekday_0	P-Value = 0.0 => Stationary.
weekday_1	P-Value = 0.0 => Stationary.
weekday_2	P-Value = 0.0 => Stationary.
weekday_3	P-Value = 0.0 => Stationary.
weekday_4	P-Value = 0.0 => Stationary.
weekday_5	P-Value = 0.0 => Stationary.
weekday_6	P-Value = 0.0 => Stationary.

Model oluşturulmadan önce veri setinin analiz edilmesi gereken birçok veri noktasına sahip olmasından dolayı, günlük aralıklarla sıkıştırılarak zamanın yeniden örneklenmesi sağlanmıştır. Böylece analiz edilmesi daha kolay olan daha az veri noktası elde edilmiştir.

Geliştirilen modelin başarılı tahminler yapabilmesi için en uygun parametrelerin seçilmesi çok kritiktir. Bu nedenle çalışmada OtomatikARIMA (*AutoARIMA*) kullanımı tercih edilmiştir. OtomatikARIMA, ARIMA modeli için en uygun p,d ve q parametrelerini otomatik olarak bulur. OtomatikARIMA ile en uygun parametreler belirlenirken, Akaike Bilgi Kriteri (AIC) kullanılmıştır. AIC değeri, geliştirilen modelin karmaşıklığının ve verilere ne kadar iyi uyduğunun tespit edilmesini sağlar. AIC puanı daha düşük olan modeller, veriye daha iyi uyum sağlamıştır denilebilir. OtomatikARIMA ise AIC değerini en aza indirmek için adım adım arama gerçekleştirir. OtomatikARIMA ile yapılan çalışma sonucunda en iyi ARIMA modeli için seçilen parametre değerleri ve AIC değeri Tablo 4.4.'de gösterilmiştir.

Tablo 4.4: OtomatikARIMA test sonuçları.

ARIMA(0,0,0)(0,0,0)[0]	AIC=-363272.258,	Time=13.89 sec
ARIMA(1,0,0)(0,0,0)[0]	AIC=inf,	Time=17.80 sec
ARIMA(0,0,1)(0,0,0)[0]	AIC=-608852.607,	Time=30.78 sec
ARIMA(1,0,1)(0,0,0)[0]	AIC=-1534710.068,	Time=35.62 sec

Tablo 4.4: Devam.

ARIMA(2,0,1)(0,0,0)[0]	AIC=-1541905.140,	Time=40.96 sec
ARIMA(2,0,0)(0,0,0)[0]	AIC=inf,	Time=21.49 sec
ARIMA(3,0,1)(0,0,0)[0]	AIC=-1539345.095,	Time=202.62 sec
ARIMA(2,0,2)(0,0,0)[0]	AIC=-1542345.491,	Time=69.26 sec
ARIMA(1,0,2)(0,0,0)[0]	AIC=-1537366.516,	Time=52.58 sec
ARIMA(3,0,2)(0,0,0)[0]	AIC=-1543669.011,	Time=67.79 sec
ARIMA(4,0,2)(0,0,0)[0]	AIC=-1539834.076,	Time=240.46 sec
ARIMA(3,0,3)(0,0,0)[0]	AIC=-1544814.255,	Time=94.58 sec
ARIMA(2,0,3)(0,0,0)[0]	AIC=-1544826.743,	Time=80.60 sec
ARIMA(1,0,3)(0,0,0)[0]	AIC=-1544730.055,	Time=67.06 sec
ARIMA(2,0,4)(0,0,0)[0]	AIC=-1544817.102,	Time=29.78 sec
ARIMA(1,0,4)(0,0,0)[0]	AIC=-1544833.157,	Time=91.76 sec
ARIMA(0,0,4)(0,0,0)[0]	AIC=-1034589.392,	Time=169.36 sec
ARIMA(1,0,5)(0,0,0)[0]	AIC=-1544838.104,	Time=55.38 sec
ARIMA(0,0,5)(0,0,0)[0]	AIC=-1110088.236,	Time=79.62 sec
ARIMA(2,0,5)(0,0,0)[0]	AIC=-1544828.063,	Time=45.45 sec
Best model:	ARIMA(1,0,5)(0,0,0)[0]	
Total fit time:	1786415 seconds	

4.3.1.4 Vektör Otoregresyon

Vektör Otoregresyon (VAR) modeli, 1980 yılında Simms tarafından önerilen, tek değişkenli otoregresif sürecin uzantısı olan çok değişkenli zaman serisi modelidir. Granger nedensellik testi modelini temel alır. VAR modeli, gelecek dönem adımlar için tahmini yapılacak değişkenin, veri setindeki diğer değişkenler ile arasındaki

bağımlılığı inceleyerek, tahminler geliştirir. X_t ve Y_t zaman serileri ele alındığında, VAR modeli matematiksel olarak aşağıdaki şekilde tanımlanmaktadır:

$$Y_t = \alpha + \sum_{j=1}^m \beta_j Y_{t-j} + \sum_{j=1}^m \delta_j X_{t-j} + \varepsilon_{1t} \quad (4.7)$$

$$X_t = \alpha + \sum_{j=1}^m \theta_j Y_{t-j} + \sum_{j=1}^m \vartheta_j X_{t-j} + \varepsilon_{2t} \quad (4.8)$$

Formüllerde gösterilen ε_{1t} ve ε_{2t} değişkenleri hata terimleridir. X 'in gecikmeli değerleri ve Y 'nin gecikmeli değerleri birbirlerini etkilemektedir. 4.7 ve 4.8 nolu denklemlerin sağ tarafında ise yalnızca gecikmeli değişkenler bulunur. VAR modeli n -denklemden oluşur ve değişkenlerin p -gecikmelerini içerir. Her değişkenin ilk p gecikmeleri, her değişken için regresyon tahminleyicisi olarak kullanılır. VAR modeli geliştirirken en önemli adımlardan biri p değerine yani gecikme değerine karar vermektir. En sık kullanılan gecikme seçim kriterleri Hannan-Quinn Bilgi Kriteri (*HQ*), Schwarz-Bayesian Bilgi Kriteri (*BIC*) ve Akaike Bilgi Kriteridir (*AIC*). Tez çalışması kapsamında p değerine karar verebilmek için *AIC* kullanılmıştır. p değeri arttıkça model daha karmaşık hale geleceğinden, *AIC* karmaşık modeli cezalandırır. [6], [54], [55].

VAR modeli oluşturulmadan önce, ARIMA modelinde olduğu gibi veri setinin analiz edilmesi gereken birçok veri noktasına sahip olmasından dolayı, günlük aralıklarla sıkıştırılarak zamanın yeniden örneklenmesi sağlanmıştır. VAR modeli oluşturulurken, diğer yaklaşımlardan farklı olarak her bir küme için yeni bir model oluşturulur. Her bir değişkenin modelin başarısına olan etkisini değerlendirebilmek amacıyla, tahmini yapılacak “Bellek Kullanımı (%)” özniteliği ile her bir öznitelik sırası, ikili olarak modele verilmiştir. “Bellek Kullanımı (%)” özniteliğinin tahmin edilmesine en çok katkı sağlayan özniteliği değerlendirebilmek için RMSE değerleri hesaplanarak sıralanmıştır.

4.3.2 Derin Öğrenme Teknikleri

Derin öğrenme teknikleri, insan beyninin nöron yapısını yani veri işleme şeklini taklit ederek yeni bir yöntem sunan makine öğrenme yöntemidir. Esasında derin öğrenme, makine öğrenmesinin bir alt sınıfıdır. Derin öğrenme tekniklerinde, makine öğrenmesinden farklı olarak aynı anda birden çok katmanda işlem yapılabilmesi ve manuel işlem olmaksızın tüm sistemin otomatik hale getirilmesi hedeflenmektedir.

Örnek olarak, makine öğrenmesi modellerinde daha çok insan müdahalesi ile oluşturulan parametre tanımlama, özellik çıkarma gibi işlemler derin öğrenme tekniklerinde modele bizzat yaptırılmaktadır. Derin öğrenme tekniklerinde oldukça büyük oranda veri işlemek ve kendi kendine öğrenme sonucu bir hesaplama modeli çıkarmak hedeflenir ve böylece veri kümesindeki çıktıları tahmin edecek yapay zekanın eğitilmesi sağlanır. Derin öğrenme modellerinde hem denetimli hem de denetimsiz öğrenme kullanılabilir [3], [56].

Bir derin öğrenme modeli geliştirilirken, tanıtılmak istenen nesne türleri, eğitim seti olarak bilgisayara yüklenir. Derin öğrenme algoritmaları sebep-sonuç ilişkisi kurabildiğinden yararsız kodlar elenerek, yararlı kodlar verimli bir şekilde kullanılabilir. Geliştirilen derin öğrenme modelinin katman sayısı arttıkça bilgisayar daha fazla veri yani bilgi ile besleneceğinden daha yüksek oranda doğruluk oranı elde edilebilir [3], [56].

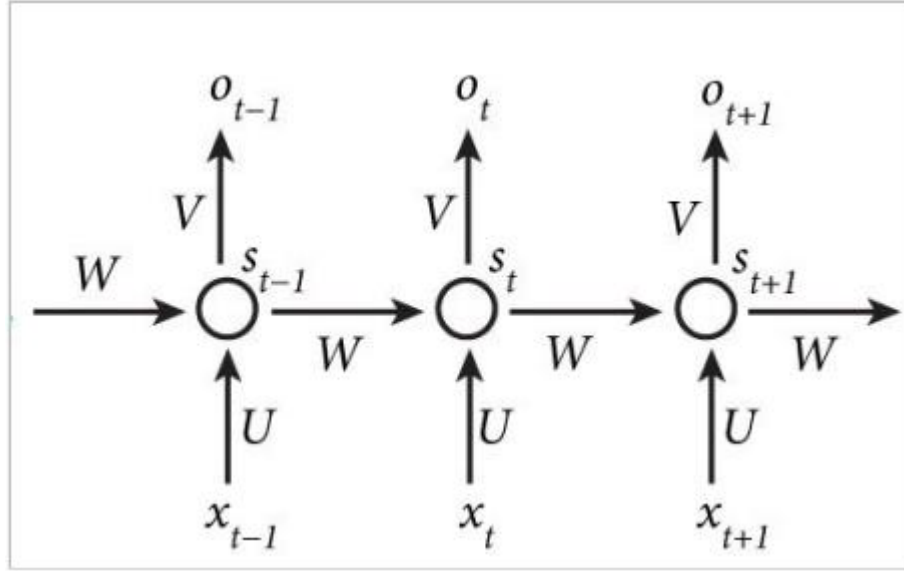
Bulut teknolojisinin gelişmesi ile birlikte, CPU ve GPU gücü giderek hızlanmıştır. Veri miktarının da giderek artmasıyla, derin öğrenme yöntemleri hemen hemen her alanda uygulanabilir hale gelmiştir [3], [56].

Tez çalışması kapsamında, derin öğrenme tekniklerinden RNN, LSTM, BiLSTM ve BiLSTM-RNN kullanılarak zaman serisi tahmin modelleri geliştirilmiştir.

4.3.2.1 Yinelemeli Sinir Ağı (RNN)

Geleneksel sinir ağı kullanılan modellerde, modele verilen girdi ve çıktıların birbirinden bağımsız olduğu varsayılır. Ancak bu yaklaşım pek çok günümüz problemlerinin birçoğuna uygun değildir. Yinelemeli Sinir Ağı (*RNN*), geleneksel yaklaşımların aksine, bir dizi bilgiden yararlanırken, bir önceki katmandan elde edilen çıktıları da girdi olarak kullanır. Yani RNN birimlerinin çıktıları sadece mevcut girdiye bağlı değildir, geçmişte öğrenebilmektedir. Bu bağlamda RNN, sinir ağı boyunca bilgiyi elinde tutabilen bir bellek gibi davranır. Ancak RNN'in bazı sınırları vardır. Pratik uygulamalarda bir çok adımda bellek özelliğini yitirebilmektedir. Bu problemi aşabilmek için RNN'in "Uzun-Kısa Süreli Bellek" ve "Kapılı Tekrarlayan Birim" olmak üzere 2 çeşidi ortaya çıkmıştır [3], [32], [57].

RNN'in çalışma prensibini daha iyi anlayabilmek için Şekil 4.2. incelenebilir:



Şekil 4.2: Yinelemeli sinir ağı modeli.

Çalışma şemasını incelediğimizde x_t ; t anındaki girdiyi temsil ederken, s_t ; t anındaki saklı durumu temsil eder ve RNN'in hafıza kısmını oluşturur. Hesaplanan bir önceki saklı durum ve mevcut giriş değeri ile, s_t değeri aşağıdaki şekilde bulunur:

$$s_t = f(U x_t + W s_{t-1}) \quad (4.9)$$

Formülde kullanılan f fonksiyonu, doğrusal olmayan ReLU veya tanh fonksiyonu olabilir.

O_t , t anındaki çıktıyı temsil eder ve aşağıdaki formül ile hesaplanabilir:

$$o_t = \text{softmax}(V s_t) \quad (4.10)$$

RNN'de U, V ve W parametreleri ağ boyunca kullanılır.

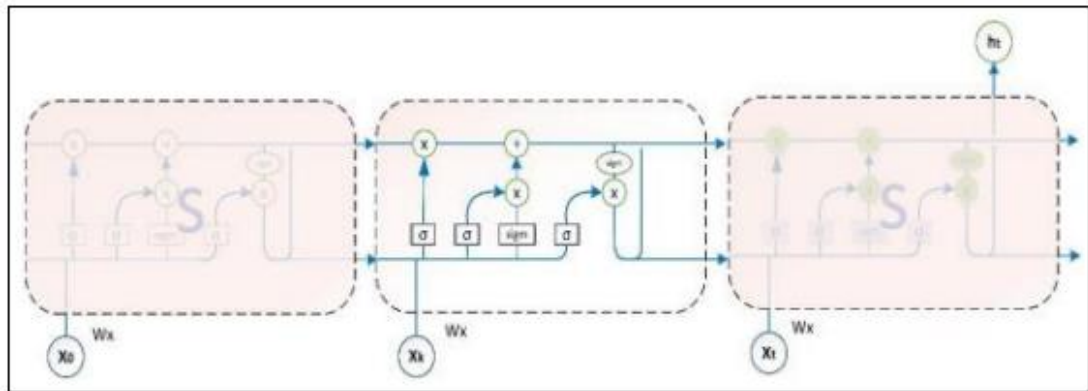
- U: Gizli katmanın ağırlık değerini
- V: Çıktı katmanının ağırlık değerini
- W: Gizli durum geçişinin ağırlık değerini temsil eder [32].

Tez çalışması kapsamında “Bellek Kullanımı (%)” değerinin gelecek dönem değerlerinin tahmini için geliştirilen RNN modelinde, tahmin için kullanılacak olan geçmiş zaman adımı noktası sayısı yani pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde modele verilmiştir. Her bir güne ait 4 adet zaman adımı noktası

bulunmaktadır. Bu nedenle pencere boyutu 1 hafta seçildiğinde 28 adet, 2 hafta seçildiğinde 56 adet, 3 hafta seçildiğinde ise 84 adet zaman adımı modele verilir. Geliştirilen modelde aktivasyon fonksiyonu olarak $[0, +\infty)$ aralığında değer veren “Relu” fonksiyonu kullanılmıştır. Modelde kullanılacak hiperparametreler, farklı nöron sayısı ve farklı eğitim dönemi kombinasyonları ile birden fazla RNN ağının eğitilmesi sonucu belirlenmiştir. Bu deneyler sonucunda RNN ağının nöron sayısı 50 olarak ayarlanırken, öğrenme dönemi sayısı 50 olarak belirlenerek 1 gizli katmana yapılandırıldığı zaman en iyi tahmin sonuçları elde edilmiştir. Modelde en iyileme fonksiyonu ‘adam’ olarak belirlenmiştir [10].

4.3.2.2 Uzun Kısa Süreli Bellek (LSTM)

Uzun Kısa Süreli Bellek, tekrarlayan yapay sinir ağlarının uzun vadeli bağımlılıklarını öğrenmek için kullanılan başka bir topolojisidir. LSTM, standart beslemeli sinir ağlarının aksine, geri bildirim bağlantılarına sahiptir. LSTM modeli hücre durumu bilgisini kaydeden hafıza blokları ve kullandığı çeşitli kapılar ile bilgi akışını kontrol eder. Hafıza blokları bir ya da daha fazla hafıza hücresinin toplamsal ve çarpımsal geçit birimlerinden oluşmaktadır. LSTM giriş, unutma ve çıkış geçitleri sayesinde ise hangi bilginin hatırlanıp, hangi bilginin unutulacağına veya bir sonraki adıma geçeceğine karar verir. Bu sayede gelen bilgi tahmin yapmak için anlamlı ise bir sonraki aşamaya aktarılarak hücreler boyunca taşınır. Şekil 4.3’de LSTM modelinin yapısı ve bir hafıza hücresini göstermektedir [3], [56], [58].



Şekil 4.3: LSTM model yapısı.

LSTM ağı, giriş katmanından aktarılan verilerden ağ aktivasyonlarını hesapladıktan sonra çıkış katmanına iletmek için bir eşleme hesaplanır. Bu hesaplamalar için gerekli tekrarlı denklemler aşağıdaki şekildedir [56]:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (4.11)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (4.12)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (4.13)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \quad (4.14)$$

$$m_t = o_t \odot h(c_t) \quad (4.15)$$

$$y_t = \phi(W_{ym}m_t + b_y) \quad (4.16)$$

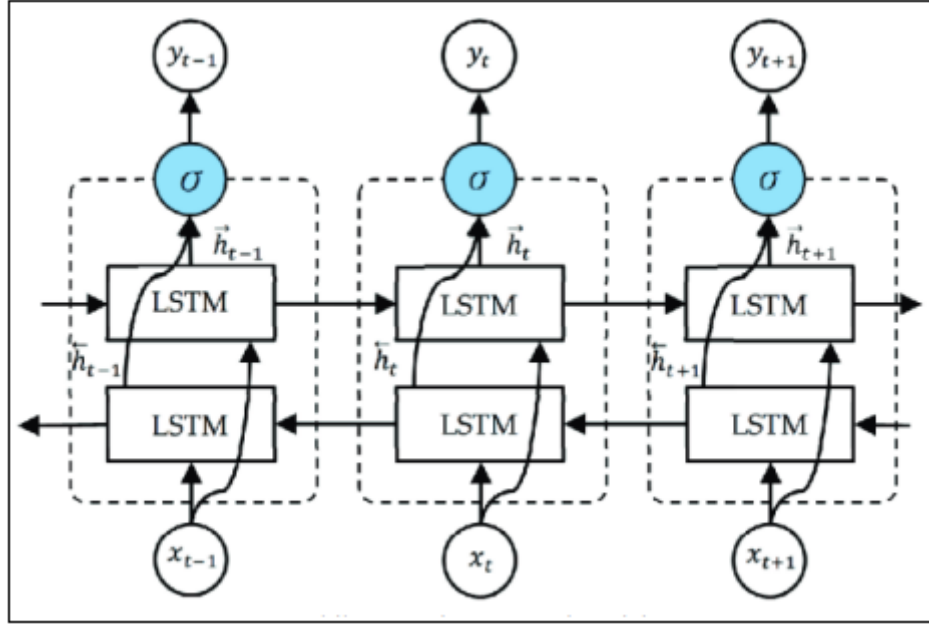
Formülde kullanılan terimlerin açıklaması şu şekildedir [56]:

- W : ağırlık matrisi
- W_{ix} : giriş geçidinden girişe kadar kullanılan ağırlık matrisi
- W_{ic} , W_{fc} , W_{oc} : geçit katmanlarının hücre durumuna bakan köşegen ağırlıklı matrisler
- b : önyargıyı temsil eden terim
- b_i : giriş geçidi önyargı vektörü
- σ : sigmoid aktivasyon fonksiyonu
- i : giriş geçidi
- f : ihmal geçidi
- o : çıktı geçidi
- c : hücre aktivasyon vektörü
- m : hücre çıktı aktivasyon vektörü
- $\odot$: vektörlerin parça bazlı ürünü
- g : hücre girdisi
- h : hücre çıktısı
- ϕ : tanh işlevi

Tez çalışması kapsamında “Bellek Kullanımı (%)” değerinin gelecek dönem değerlerinin tahmini için geliştirilen LSTM modelinde de RNN modelinde olduğu gibi pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde modele verilmiştir. Geliştirilen modelde aktivasyon fonksiyonu olarak $[0, +\infty)$ aralığında değer veren “Relu” fonksiyonu kullanılmıştır. Modelde kullanılacak hiperparametreler, RNN’de olduğu gibi, farklı nöron sayısı ve farklı eğitim dönemi kombinasyonları ile birden fazla LSTM ağınnın eğitilmesi sonucu belirlenmiştir. Nöron sayısı 50 ve öğrenme dönemi sayısı 50 olarak belirlenerek 1 gizli katmana yapılandırıldığı zaman en iyi tahmin sonuçları elde edilmiştir. Modelde en iyileme fonksiyonu olarak ‘adam’ kullanılmıştır [10].

4.3.2.3 Çift Yönlü Uzun Kısa Süreli Bellek (BiLSTM)

Çift Yönlü Uzun Kısa Süreli Bellek yani BiLSTM, LSTM yapısının hem pozitif hem negatif zaman yönlerinde eğitilen bir versiyonudur. BiLSTM mimarisinde giriş verileri sıralı ve ters sıralı biçimde eş zamanlı olarak taranabilir. BiLSTM mimarisi, ileri ve geri yönlü olmak üzere çift yönlü iki paralel katman içerir ve ileri-geri geçişler bu katmanlardan yayılır. Çift yönlü olması nedeni ile, LSTM modelinde tanıtılan yinelemeli denklemler hem ileri hem geri yönde uygulanmaktadır. Şekil 4.4’de BiLSTM mimarisi gösterilmektedir.



Şekil 4.4: BiLSTM mimarisi.

H_{ft} ve h_{bt} sırası ile ileri yayılım katmanının gizli durumunu ve geri yayılım katmanının gizli durumunu temsil ederken, BiLSTM ağının gizli durumunu temsil eden h_t aşağıdaki denklem ile hesaplanabilir:

$$y_t = [h_{ft} \oplus h_{bt}] \quad (4.17)$$

Bu denkleme göre y_t , her bir zaman adımın çıktısını ifade eder ve elde edilen gizli durum değerine doğrusal katman ve softmax katmanı uygulanır [59], [60].

Tez çalışması kapsamında geliştirilen BiLSTM modelinde de tahmin için pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde modele verilerek test edilmiştir. Geliştirilen BiLSTM ağı 64 nöronlu bir katman ve gizli bir katman içerir. Öğrenme dönemi sayısı 50 olarak belirlenmiştir. Aktivasyon fonksiyonu olarak ReLU tercih edilmiştir ve en iyileme fonksiyonu ‘adam’ olarak seçilmiştir [10].

4.3.2.4 BiLSTM – RNN

Tez çalışması kapsamında, derin öğrenme tekniklerinden son olarak BiLSTM ve RNN ağları birlikte kullanılarak bir model oluşturulmuş ve test edilmiştir. Diğer derin öğrenme tekniklerinde olduğu gibi, geçmiş 1 hafta, 2 hafta ve 3 haftaya ait zaman adımı noktaları pencere boyutu olarak seçilerek sırası ile test edilmiştir. Modelde aktivasyon fonksiyonlarından ReLU seçilmiştir. Model, 1 katmanlı 64 nörona sahip

BiLSTM ağından ve 1 katmanlı 128 nöronlu RNN ağından ve bunların ardından ağa eklenen 1 gizli katmandan meydana gelmektedir. Öğrenme dönemi sayısı 50 olarak ayarlanırken, en iyileme fonksiyonu ‘adam’ olarak seçilmiştir [10].

4.4. VERİ AYIRMA YÖNTEMLERİ

Yapılan çalışmada geliştirilen modellerin değerlendirme aşamasından önce, veri seti 2 farklı yaklaşımla eğitim seti ve test seti olarak ayrılmıştır. Bu yöntemler veri bazında ayırma yöntemi ve küme bazında ayırma yöntemi olarak isimlendirilmiştir.

4.4.1 Veri Bazında Ayırma Yöntemi

Veri bazında ayırma yönteminde, veri setinde mevcut olan 53 adet kümeye ait verilerin tamamının %80’i modelin eğitilmesi için kullanılırken, tüm kümelerin verilerinden geri kalan %20’si test verisi olarak ayrılmıştır. Geliştirilen model, kümelerin tamamına ait %80’lik eğitim verisi ile başlangıçta bir kere eğitilir. Eğitimin ardından 53 kümenin her biri için ayrı ayrı test edilir [10].

4.4.2 Küme Bazında Ayırma Yöntemi

Küme bazında ayırma yönteminde, veri setinde mevcut olan 53 kümeden 42 tanesinin bütün verileri eğitim için kullanılmıştır. Geriye kalan 11 adet kümenin verileri ise test verisi olarak ayrılmıştır. Geliştirilen model, modelin eğitilmesi için ayrılan 42 kümenin tüm verileri ile başlangıçta bir sefer eğitilir. Eğitimin ardından geriye test etmek için ayrılan 11 kümenin verileri ile ayrı ayrı test edilir [10].

Küme bazında ayırma yöntemi ile geliştirilen modellerin başarısını değerlendirebilmek için, k-katlamalı çapraz doğrulama yöntemi ile testler tekrarlanmıştır. K değeri 5 seçilerek, 5 ayrı turda her bir kümenin hem eğitim hem de test için kullanılması sağlanmıştır.

5. DENEYSEL SONUÇLAR

Bulut bilişim veri merkezlerinde konumlanan kümelerin bellek kaynak kullanımlarının tahmin edilebilmesi için istatistiksel tekniklerden ARIMA ve VAR modelleri geliştirilirken, derin öğrenme tekniklerinden RNN, LSTM, BiLSTM ve BiLSTM-RNN modelleri geliştirilmiştir. VAR modeli haricinde geliştirilen tüm tahmin modelleri hem veri bazında ayırma yöntemi hem de küme bazında ayırma yöntemleri ile değerlendirilmiştir. Her bir modelin başarısının test edilmesi için derin öğrenme tekniklerinde sıklıkla kullanılan değerlendirme yöntemlerinden kök ortalama kare hatası (RMSE) kullanılmıştır.

ARIMA ile sadece tek değişkenli zaman serisi modelleri geliştirmek mümkündür. Ancak oluşturulan veri seti 44 öznitelikten oluşmaktadır ve tez çalışmasında asıl amaç bellek kullanım değerinin tahmin edilmesi için tüm özniteliklerin modelde kullanılmasını sağlamaktır. Literatür araştırması esnasında benzer çalışmalar incelenirken, birçok çalışmada tek değişkenli zaman serisi modeli geliştirildiği gözlemlendiğinden, bu çalışma da yeni bir yaklaşım geliştirebilmek adına çok değişkenli zaman serisi modelleri tasarlamak hedeflenmiştir. Bunun yanı sıra istatistiksel modellerden ARIMA'nın benzer çalışmalarda kullanımı çok yaygın olduğundan, bu tez çalışması kapsamında test edilmesine yer verilmiştir.

Tablo 5.1, ARIMA modelinin küme bazında ayırma yöntemi ile test edilmesi sonucu elde edilen RMSE değerlerini göstermektedir. ARIMA modeli test edildiği 11 kümede ortalama 6,75 RMSE değerine sahiptir.

Tablo 5.1: ARIMA modeli küme bazında ayırma yöntemi RMSE değerleri.

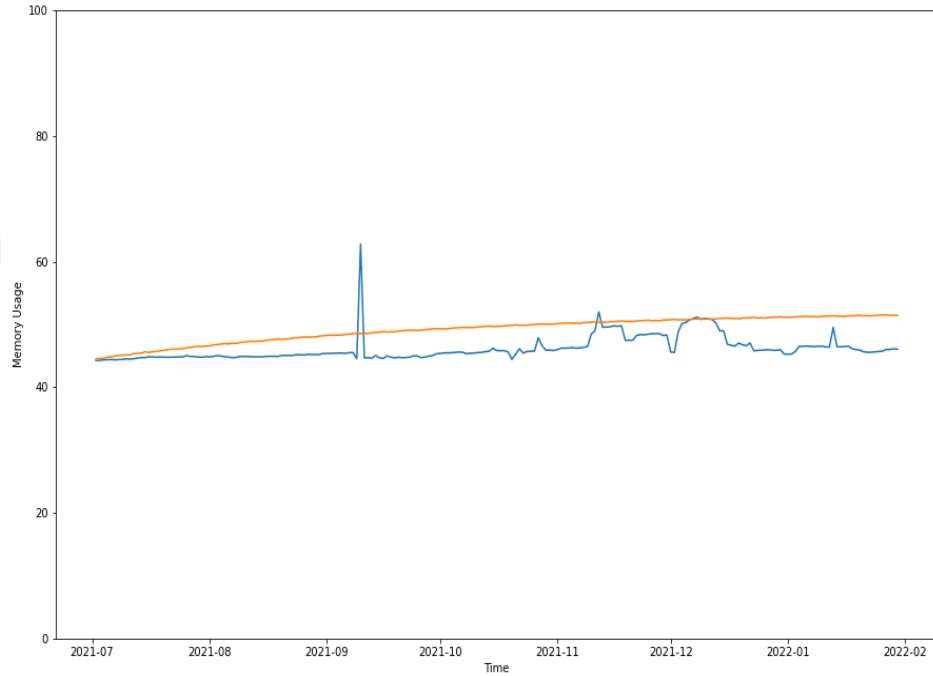
RMSE Değerleri	
Küme1	3,43
Küme2	8,91
Küme3	3,01
Küme4	3,12
Küme5	21,55
Küme6	3,36
Küme7	9,72
Küme8	6,32
Küme9	3,48
Küme10	4,88

Tablo 5.1: Devam.

RMSE Değerleri	
Küme11	6,42

ARIMA modeli veri bazında ayırma yöntemi ile test edildiğinde, 53 küme için ortalama 7,33 RMSE değerine sahip olmuştur. 53 küme arasında en düşük RMSE değeri 4,12 iken en yüksek RMSE değeri 22,34 olmuştur.

Şekil 5.1, ARIMA modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” grafiğini göstermektedir. Grafikte x-ekseni zaman adımlarını temsil ederken, y-ekseni “Bellek Kullanım (%)” değerini göstermektedir. Mavi noktalar gerçek kaynak kullanım değerleri gösterirken, turuncu noktalar bellek kaynak kullanım değerine ait tahmin edilen değerleri temsil etmektedir.



Şekil 5.1: ARIMA modeli – “gerçek-tahmin” grafiği.

VAR modelinin oluşturulurken her bir küme için yeni bir model oluşturulmasından dolayı, VAR modelini değerlendirirken sadece veri bazında ayırma yöntemi uygulanabilmektedir. VAR modeli, ARIMA’dan farklı olarak çok değişkenli zaman serisi modellemeye uygundur. Bölüm 4.2.4.4’de bahsedildiği üzere, her bir özniteliğin modelin başarısına olan etkisini değerlendirebilmek amacıyla, tahmini yapılacak “Bellek Kullanımı (%)” özniteliği ile her bir öznitelik sırası ile, ikili olarak

modele verilmiştir. Tablo 5.2, “Bellek Kullanımı (%)” öz niteliği ile birlikte modele verilen her bir öz nitelik için hesaplanan RMSE değerlerini göstermektedir.

Tablo 5.2: Öz nitelik – RMSE değerleri.

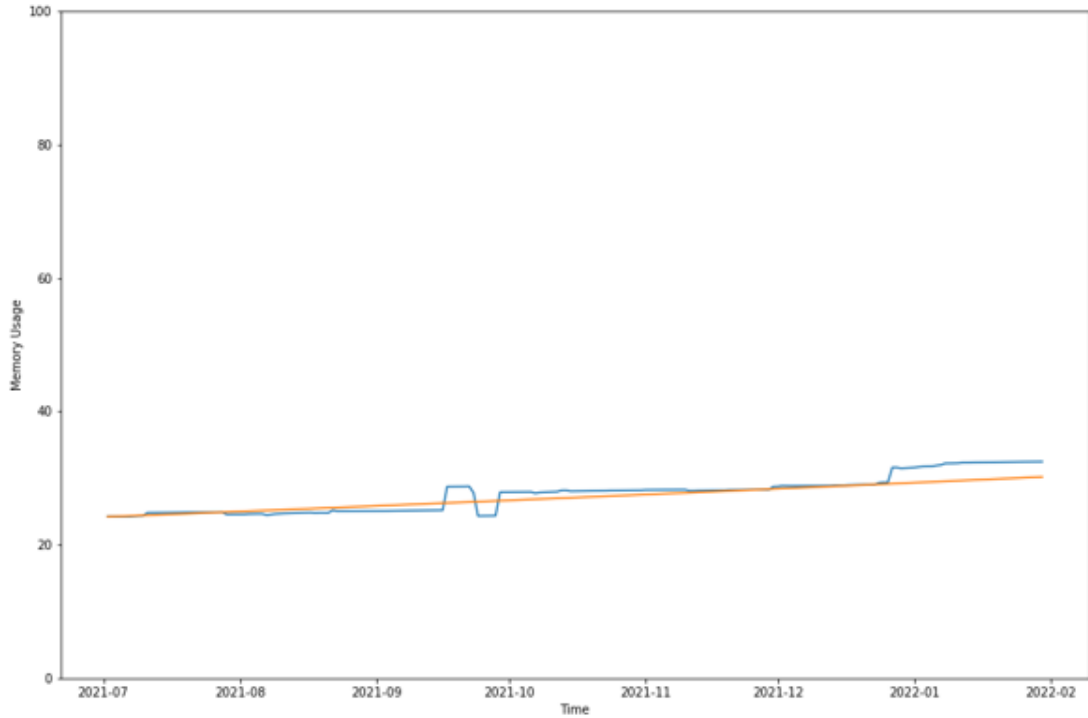
ÖZNİTELİK	RMSE
Memory Usage / Usable (%)	5,835
Memory Workload (%)	5,869
Memory Usable Capacity (TB)	5,923
Memory Usable Memory (TB)	6,111
Memory Utilization (GB)	6,114
Memory Shared Common (GB)	6,122
Summary Average Running VM Count per Running Host	6,125
Summary VM Count per Host	6,134
Memory Reserved Capacity (GB)	6,149
weekday_5	6,176
weekday_3	6,177
day	6,206
weekday_6	6,247
weekday_2	6,276
weekday_0	6,285
weekday_1	6,291
weekday_4	6,389
Memory Consumed (GB)	6,399
Memory Memory Allocated on all Consumers (GB)	6,453
Memory Memory Allocated on all Powered On Consumers (GB)	6,459
year	6,474
Memory Shared (GB)	6,495
month	6,51
Memory Machine Demand (GB)	6,536
Summary Total Number of Hosts	6,544
Summary Number of Running Hosts	6,546
Memory Total Capacity (TB)	6,548
Summary Cluster Availability (%)	6,59
Summary Number of Running VMs	6,733
Summary Total Number of VMs	6,797
Memory Guest Active (GB)	6,869
Memory Contention (%)	6,899
Memory Host Usage (GB)	6,923
Memory ESX System Usage (GB)	7,038
Memory Max VM Memory Contention (%)	7,097
hour	7,138
Memory Granted (TB)	7,258
Memory Current Overcommit Ratio	7,33

Tablo 5.2: Devam.

ÖZNETELİK	RMSE
Memory Heap (MB)	8,362
Memory Contention (Byte)	8,919
Memory Compressed (GB)	9,299
Memory Swap Out (GB)	9,328

Tablo 5.2 incelendiğinde, bellek kullanımı tahminimi için en yüksek katkıyı sağlayan özneteliğin, en düşük RMSE değerine sahip olmasından dolayı, “Bellek Kullanımı/Kullanılabilirlik (%)” olduğu görülmektedir. VAR modeli ile “Bellek Kullanım (%)” değerinin tahmin değerleri hesaplanırken modele verilen her öznetelik ikilisinin, her bir küme ve zaman adımı için tahmin değerleri tutulur. Ardından her öznetelik ikilisinin tahmin değerlerinin ortalaması alınır. Elde edilen sonuç, o kümenin ilgili zaman adımıdaki tahmin değeri olarak kaydedilir.

Şekil 5.2, VAR modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” grafiğini göstermektedir. Grafikte x-ekseni zaman adımlarını temsil ederken, y-ekseni “Bellek Kullanım (%)” değerini göstermektedir. Mavi noktalar gerçek kaynak kullanım değerleri gösterirken, turuncu noktalar bellek kaynak kullanım değerine ait tahmin edilen değerleri temsil etmektedir.



Şekil 5.2: VAR modeli – “gerçek-tahmin” grafiği.

Ancak grafik incelendiğinde gerçek bellek kullanım değerleri için deseni çok iyi oluşturmadığını, yalnızca doğrusal bir çizgi oluşturabildiğini ve maksimum-minimum noktaları yakalamakta başarısız olduğunu söylemek mümkündür.

VAR modeli, veri bazında ayırma yöntemi ile test edildiğinde 53 küme için elde edilen RMSE değerlerinin ortalaması 5,81 olmuştur.

Test sonuçlarını incelediğimizde, istatistiksel tahmin metotlarının yetersiz kaldığı sonucuna varılmıştır. Bu nedenle derin öğrenme yöntemleri araştırılarak 4 farklı derin öğrenme modeli daha geliştirilmiştir. Geliştirilen derin öğrenme modellerinde 44 adet öznetelik kullanılarak gelecek 28 zaman adımına ait tahminler geliştirilmiştir. Bu nedenle gerçekleştirilen çalışmada çok değişkenli çok adımlı zaman serisi modelleri geliştirilmiştir.

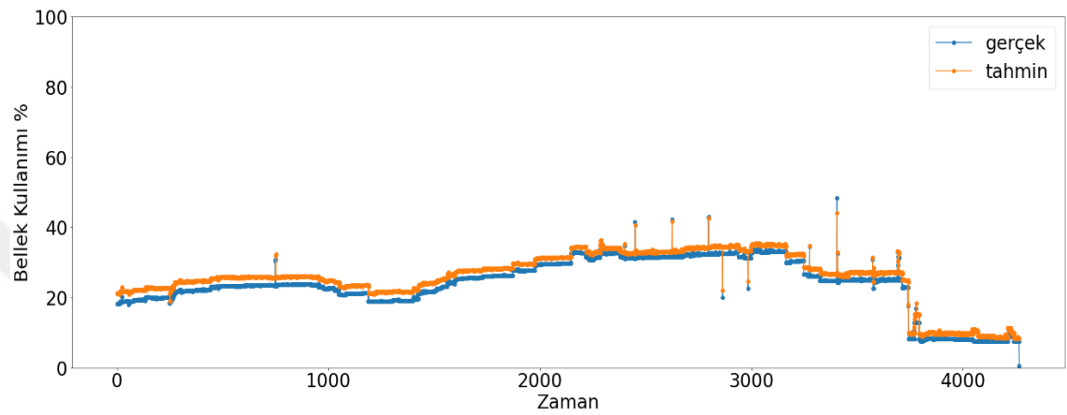
İlk olarak derin öğrenme tekniklerinden, RNN modeli geliştirilmiştir. Tablo 5.3, RNN modelinin küme bazında ayırma yöntemi ile test edilmesi sonucu, test için kullanılan her bir küme için elde edilen RMSE değerlerini göstermektedir. Tablo 5.3.'de verilen değerler, pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmin edilmesinden elde edilen RMSE değerlerini göstermektedir. Küme bazında ayırma yönteminde RNN modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 11 kümenin ortalama RMSE değeri 2,15 olarak elde edilerek en başarılı sonuç elde edilmiştir.

Tablo 5.3: RNN modeli küme bazında ayırma yöntemi RMSE değerleri.

RMSE Değerleri	
Küme1	1,55
Küme2	0,93
Küme3	2,36
Küme4	0,98
Küme5	2,62
Küme6	4,46
Küme7	5,28
Küme8	1,19
Küme9	1,37
Küme10	2,11
Küme11	1,07

RNN modeli veri bazında ayırma yöntemi ile test edilirken pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde ayarlanmıştır. Veri bazında ayırma yönteminde RNN modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 53 kümenin ortalama RMSE değeri 2,52 olarak elde edilerek en başarılı sonuç elde edilmiştir.

Şekil 5.3, RNN modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” bellek kaynak kullanım grafiğini göstermektedir.



Şekil 5.3: RNN modeli “gerçek-tahmin” grafiği.

RNN modelinin geliştirilmesinin ardından derin öğrenme tekniklerinden, LSTM modeli geliştirilmiştir. Tablo 5.4, LSTM modelinin küme bazında ayırma yöntemi ile test edilmesi sonucu, test için kullanılan her bir küme için elde edilen RMSE değerlerini göstermektedir. Tablo 5.4.’de verilen değerler, pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmin edilmesinden elde edilen RMSE değerlerini göstermektedir. Küme bazında ayırma yönteminde LSTM modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 11 kümenin ortalama RMSE değeri 2,69 olarak elde edilerek en başarılı sonuç elde edilmiştir.

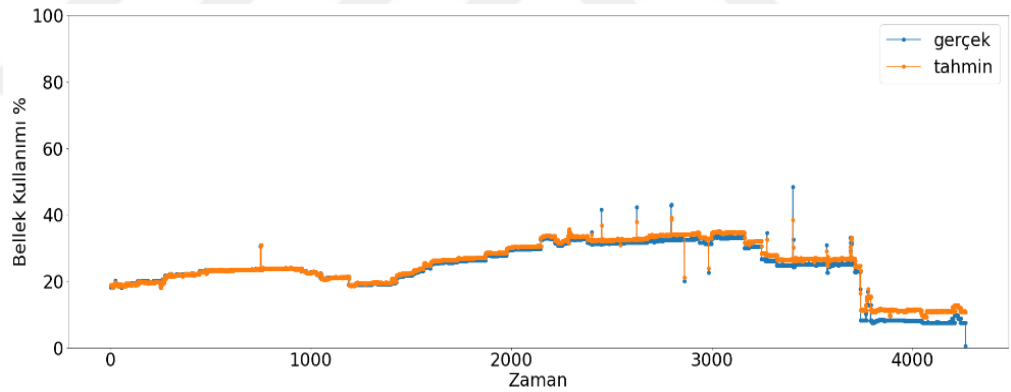
Tablo 5.4: LSTM modeli küme bazında ayırma yöntemi RMSE değerleri.

RMSE Değerleri	
Küme1	1,93
Küme2	0,42
Küme3	2,49
Küme4	0,94

Tablo 5.4: Devam.

RMSE Değerleri	
Küme5	3,07
Küme6	12,24
Küme7	6,32
Küme8	1,89
Küme9	1,29
Küme10	2,30
Küme11	3,65

LSTM modeli veri bazında ayırma yöntemi ile test edilirken pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde ayarlanmıştır. Veri bazında ayırma yönteminde LSTM modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 53 kümenin ortalama RMSE değeri 2,76 olarak elde edilerek en başarılı sonuç elde edilmiştir. LSTM modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” grafiği ise Şekil 5.4’de sunulmaktadır.



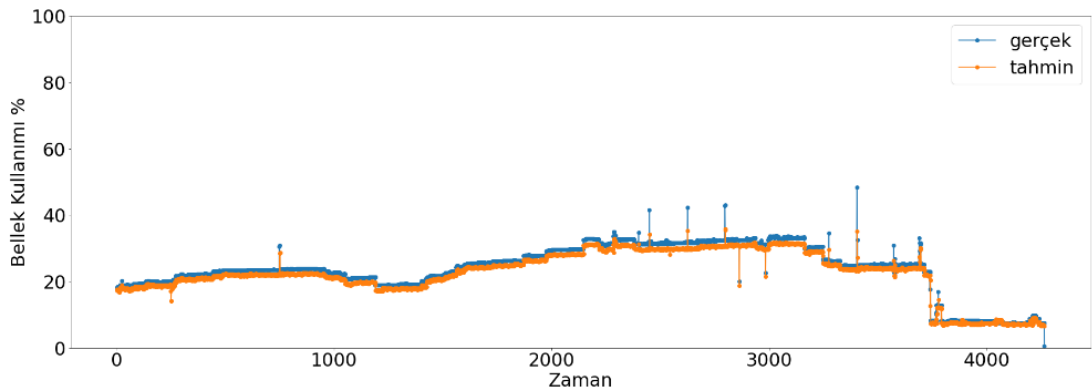
Şekil 5.4: LSTM modeli “gerçek-tahmin” grafiği.

Derin öğrenme tekniklerinden geliştirilen bir diğer model ise BiLSTM’dir. Tablo 5.5, BiLSTM modelinin küme bazında ayırma yöntemi ile test edilmesi sonucu, test için kullanılan her bir küme için elde edilen RMSE değerlerini göstermektedir. Tablo 5.5.’de verilen değerler, pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmin edilmesinden elde edilen RMSE değerlerini göstermektedir. Küme bazında ayırma yönteminde BiLSTM modeli için pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 11 kümenin ortalama RMSE değeri 2,63 olarak elde edilerek en başarılı sonuç elde edilmiştir.

Tablo 5.5: BiLSTM modeli küme bazında ayırma yöntemi RMSE değerleri.

RMSE Değerleri	
Küme1	1,33
Küme2	0,61
Küme3	1,88
Küme4	1,89
Küme5	2,53
Küme6	7,37
Küme7	5,24
Küme8	2,40
Küme9	1,41
Küme10	2,29
Küme11	2,57

BiLSTM modeli veri bazında ayırma yöntemi ile test edilirken pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde ayarlanmıştır. Veri bazında ayırma yönteminde BiLSTM modeli için pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 53 kümenin ortalama RMSE değeri 2,83 olarak elde edilerek en başarılı sonuç elde edilmiştir. BiLSTM modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” grafiği ise Şekil 5.5’de sunulmaktadır.



Şekil 5.5: BiLSTM modeli “gerçek-tahmin” grafiği.

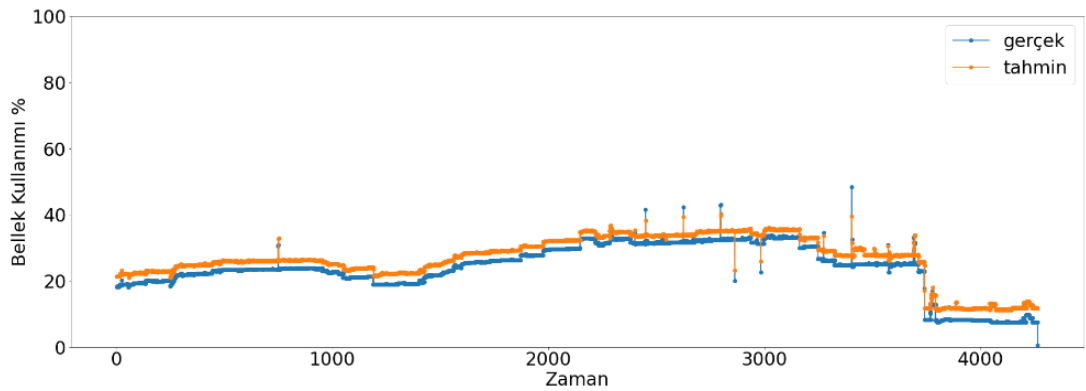
Son olarak geliştirilen model ise BiLSTM-RNN modelidir. Tablo 5.6, BiLSTM-RNN modelinin küme bazında ayırma yöntemi ile test edilmesi sonucu, test için kullanılan her bir küme için elde edilen RMSE değerlerini göstermektedir. Tablo 5.6.’da verilen değerler, pencere boyutu 1 hafta seçildiğinde, $t+1$ zaman adımının tahmin edilmesinden elde edilen RMSE değerlerini göstermektedir. Küme bazında

ayırma yönteminde BiLSTM-RNN modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 11 kümenin ortalama RMSE değeri 3,29 olarak elde edilerek en başarılı sonuç elde edilmiştir.

Tablo 5.6: BiLSTM-RNN modeli küme bazında ayırma yöntemi RMSE değerleri.

RMSE Değerleri	
Küme1	1,86
Küme2	0,97
Küme3	2,82
Küme4	2,78
Küme5	2,03
Küme6	8,65
Küme7	5,13
Küme8	3,47
Küme9	2,97
Küme10	3,09
Küme11	3,12

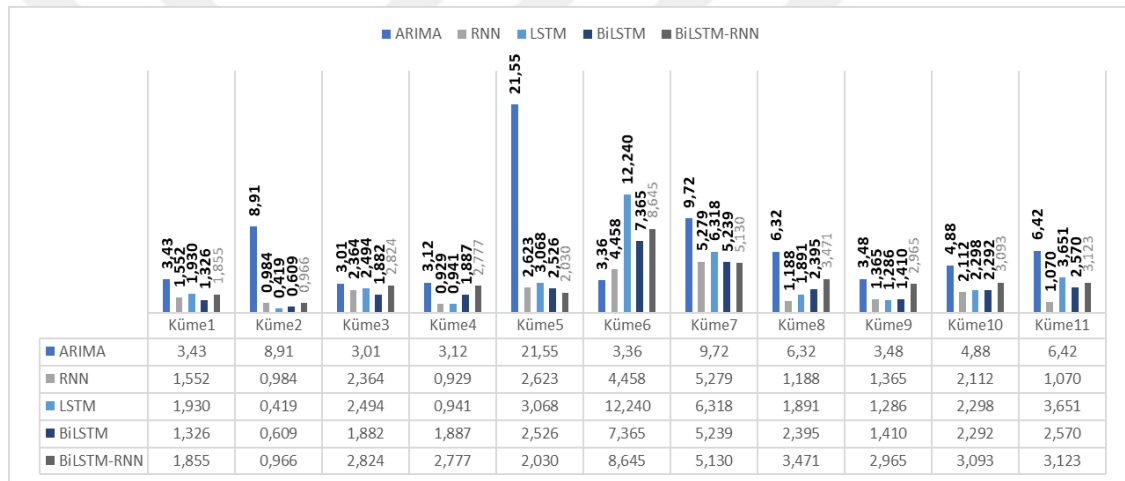
BiLSTM-RNN modeli veri bazında ayırma yöntemi ile test edilirken pencere boyutları sırası ile 1 hafta, 2 hafta ve 3 hafta olacak şekilde ayarlanmıştır. Veri bazında ayırma yönteminde BiLSTM-RNN modeli için pencere boyutu 3 hafta seçildiğinde, $t+1$ zaman adımının tahmini için 53 kümenin ortalama RMSE değeri 3,47 olarak elde edilerek en başarılı sonuç elde edilmiştir. BiLSTM-RNN modeli ile en başarılı tahmin yapılan küme için “gerçek-tahmin” grafiği ise Şekil 5.5’de sunulmaktadır.



Şekil 5.6: BiLSTM-RNN modeli “gerçek-tahmin” grafiği.

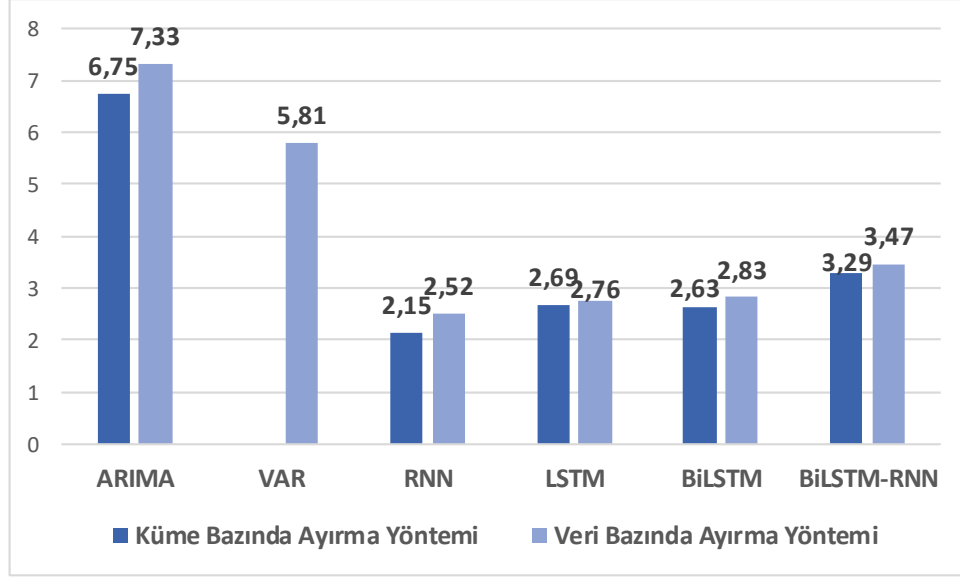
İstatistiksel tahmin yöntemlerinden ARIMA ve VAR modelleri, derin öğrenme tekniklerinden ise RNN, LSTM, BiLSTM ve BiLSTM-RNN modelleri hem veri bazında ayırma yöntemi hem de küme bazında ayırma yöntemi ile farklı pencere boyutları ile çok adımlı tahmin modeli yapılacak şekilde test edilmiştir. Yapılan testler sonucunda küme bazında veri ayırma yöntemi ile geliştirilen modeller daha yüksek doğruluk oranında kaynak kullanım tahminine sahip olmuştur.

53 küme için ortalama RMSE değeri, pencere boyutu 1 hafta olacak şekilde ayarlandığında, bir sonraki zaman adımı tahmin edildiğinde RNN modeli ile 2,728, LSTM modeli ile 3,245, BiLSTM modeli ile 2,831 ve BiLSTM-RNN modeli ile 3,503 olarak elde edilmiştir. Şekil 5.7 küme bazında ayırma yöntemi ile test edilen her bir küme için geliştirilen modellerin RMSE sonuçlarını göstermektedir.



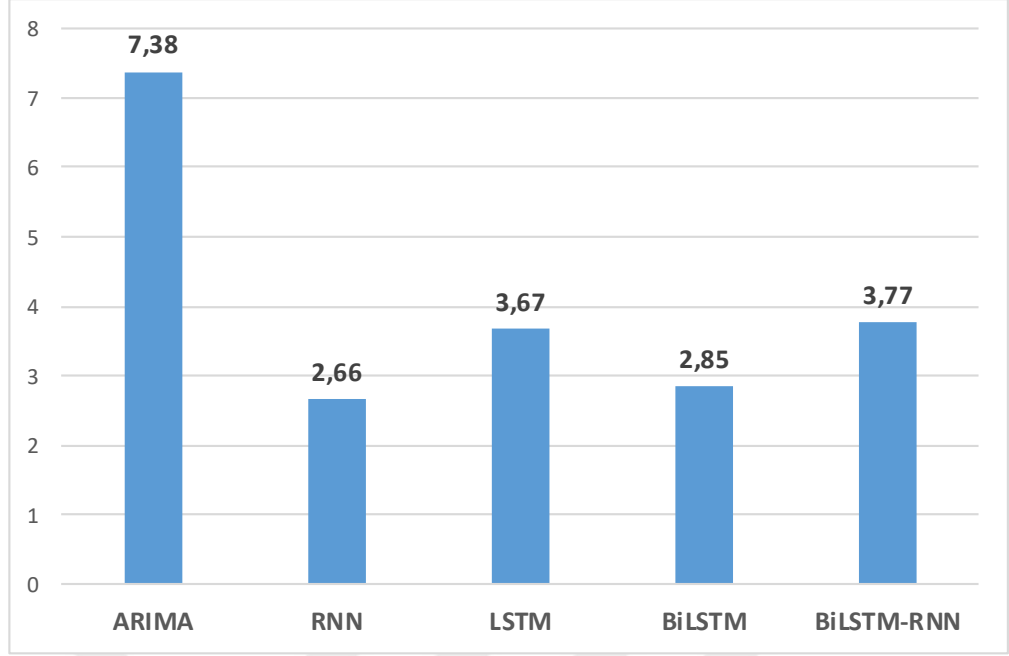
Şekil 5.7: Küme bazında ayırma yöntemi RMSE sonuçları

Şekil 5.8, geliştirilen 6 ayrı modelin küme bazında ve veri bazında ayırma yöntemlerine göre elde edilen en başarılı ortalama RMSE değerlerini sunmaktadır. En yüksek başarı elde edilen RMSE değerleri incelendiğinde hem küme bazında ayırma yönteminde hem de veri bazında ayırma yönteminde en iyi sonuç RNN modeli ile elde edilmiştir.



Şekil 5.8: Veri ayrıştırma yöntemleri RMSE sonuçları.

Geliştirilen tüm modellerin, küme bazında ayırma metodunda k-katlamalı çapraz doğrulama yöntemi ile de değerlendirmesi yapılarak deneysel sonuçların teyit edilmesi sağlanmıştır. k değeri 5 alınarak, her bir kümenin hem eğitim hem de test için kullanılması sağlanmıştır. Şekil 5.9, her bir modelin tüm turlarda elde edilen RMSE değerlerinin ortalaması alındığında elde edilen değerleri göstermektedir. RMSE değerleri ARIMA için 7,38 RNN için 2,66 LSTM için 3,67 BiLSTM için 2,85 ve BiLSTM-RNN modeli için 3,77 olarak elde edilmiştir. Tablo 5.7’de ise her turda model bazında elde edilen kümelerin ortalama RMSE değerlerini gösterilmektedir.

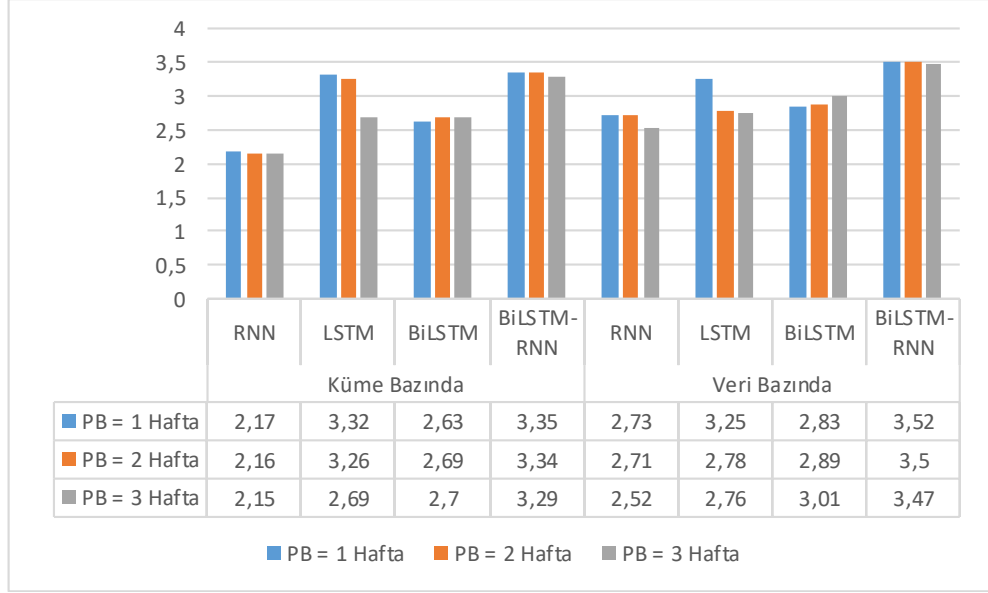


Şekil 5.9: Küme bazında ayırma yöntemi RMSE sonuçları (k-katlamalı çapraz doğrulama).

Tablo 5.7: Küme bazında ayırma yöntemi (k-katlamalı çapraz doğrulama) tur bazında RMSE sonuçları

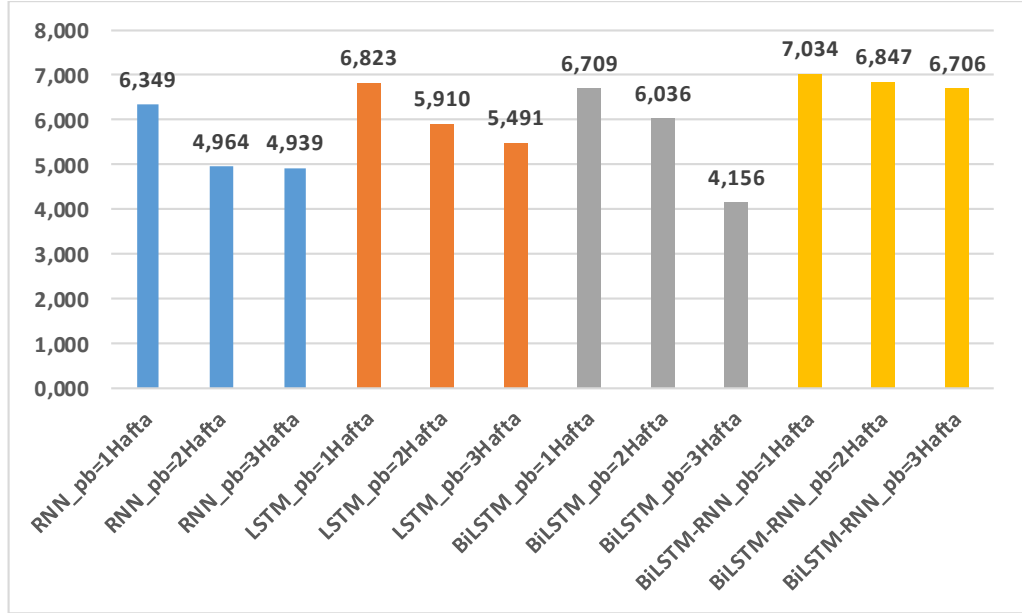
	ARIMA	RNN	LSTM	BiLSTM	BiLSTM-RNN
k=1	6,71	2,20	3,34	2,66	3,4
k=2	6,32	2,41	4,15	2,38	3,96
k=3	8,32	2,23	3,51	3,25	3,68
k=4	7,43	3,18	3,12	2,79	4,24
k=5	8,13	3,27	4,22	3,18	3,56

Küme bazında ve veri bazında ayırma yöntemlerinin her ikisinde derin öğrenme teknikleri ile geliştirilen 4 ayrı model, farklı pencere boyutları ile test edilmiştir. Pencere boyutları sırası 1 hafta, 2 hafta ve 3 hafta olarak seçilmiştir ve bunlar sırası ile 28, 56 ve 84 zaman adımına denk gelmektedir. Şekil 5.10, geliştirilen modellerin $t+1$ zaman adımı tahmini için pencere boyutlarına göre elde edilen RMSE değerlerini göstermektedir. Elde edilen RMSE değerlerini incelediğimizde en başarılı sonucun küme bazında ayırma yöntemi ile pencere boyutu 3 hafta olarak ayarlandığında RNN modeli ile elde edildiği gözlemlenmiştir. Pencere boyutu genişledikçe, RMSE değerinin azaldığı yani modelin başarısının arttığı görülmektedir.



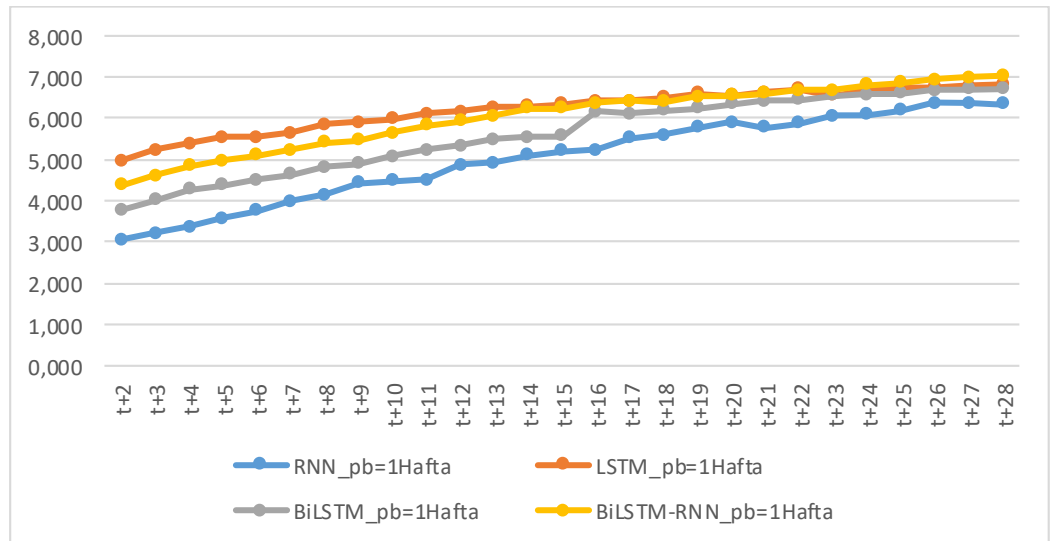
Şekil 5.10: Pencere boyutlarına göre model bazında RMSE sonuçları.

Son olarak, gerçekleştirilen çalışmada geliştirilen her bir model ile çok adımlı zaman serisi tahmini üzerine deneysel çalışmalar gerçekleştirilmiştir. Her bir model ile sonraki 28 zaman adımının tahmini yapılarak her bir zaman adımı için RMSE değerleri hesaplanmıştır. Çok adımlı zaman serisi tahmini yapılırken her bir modelde pencere boyutları sırasıyla 1 hafta, 2 hafta ve 3 hafta olarak ayarlanarak test edilmiştir. Şekil 5.11’de $t+28$ zaman adımının tahmini için kullanılan modellerde pencere boyutunun artmasının modelin başarısına olumlu etki sağladığı görülmektedir. En düşük RMSE değerleri her modelde pencere boyutu 3 hafta yani 84 zaman adımı olarak seçildiğinde, BiLSTM modeli ile elde edilmiştir.

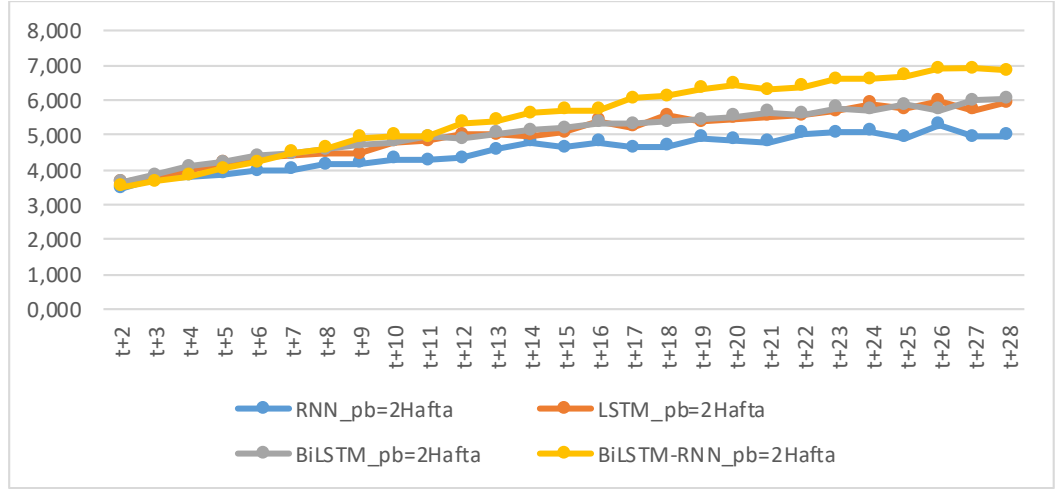


Şekil 5.11: “t+28” zaman adımı tahmini için pencere boyutlarına göre model bazında RMSE sonuçları.

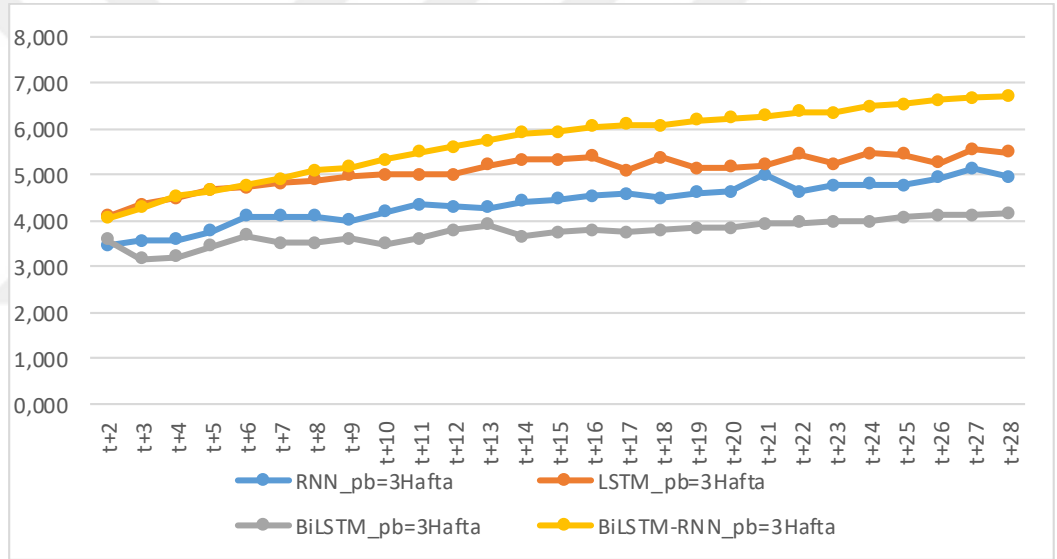
Şekil 5.12 Şekil 5.13 ve Şekil 5.14 ise sırası ile pencere boyutu 1 hafta, 2 hafta ve 3 hafta seçilerek derin öğrenme modellerine verildiği takdirde tahmin edilecek zaman adımı arttıkça RMSE değerinin arttığını göstermektedir. Yani bellek kullanım değerinin daha uzak bir zaman adımında tahmin edilmesi modelin başarısını olumsuz etkilemektedir.



Şekil 5.12: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=1 Hafta).



Şekil 5.13: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=2 Hafta).



Şekil 5.14: Çok adımlı zaman serisi tahmini için RMSE değerleri (pb=3 Hafta).

6. DEĞERLENDİRME

Günümüzde zaman serisi analizi oldukça popüler olup, birçok alanda uygulanabilmektedir. Tez çalışması kapsamında derin öğrenme ve istatistiksel tahmin modelleri kullanılarak zaman serisi analizinin nasıl gerçekleştirildiğine dair teorik bilgiler sunulmuş ve birtakım deneyler gerçekleştirilmiştir. Gerçekleştirilen çalışmada aktif faaliyet gösteren, birçok kurumsal müşteriye hizmet sunan bir bulut servis sağlayıcısına ait veri merkezlerinde barındırılan 53 adet kümeye ait 3 yılın zaman serisi verilerinin kullanılarak bellek kaynak kullanımının tahminine yönelik deneyler gerçekleştirilmiştir. Oluşturulan veri setinde yedekleme, versiyon güncelleme gibi çalışma yapılan günlere ait raporlarda veri tutarsızlığı ve eksikliği oluşabildiğinden bazı ön işleme aşamalarından geçmesi gerekmiştir. İlk olarak boş oluşan ve eksik değer içeren raporlara enterpolasyon uygulanarak eksik veriden kaynaklanacak hata toleransının azaltılması sağlanmıştır. Aynı nedenden ötürü, veri setinin oluşturulduğu tarihler arasında hizmet vermeyi durduran kümeler de veri setinden çıkarılmıştır. 2 farklı teknik altında geliştirilen 6 farklı model ise 2 adet veri ayrıştırma yöntemi ile test edilmiştir.

Zaman serilerini tek değişkenli ve çok değişkenli olarak sınıflandırmak mümkündür. İlk olarak veri setinin tek değişkenli ve çok değişkenli zaman serisi olarak tahmin başarısı değerlendirilmiştir. Bellek kullanım değeri birçok faktörden etkilenmektedir. Bu nedenle veri seti çok değişkenli zaman serisi olarak değerlendirildiğinde tahmin başarısı daha iyi sonuç vermiştir ve yeni geliştirilen modeller çok değişkenli zaman serisine uygun olacak şekilde değerlendirilmiştir. Veri seti küme bazında ve veri bazında olmak üzere iki ayrı yöntem ile eğitim – test seti olarak ayrılmıştır. Değerlendirme metriği olarak RMSE kullanılmıştır. Derin öğrenme teknikleri ile geliştirilen modellere tahmin için kullanılacak olan geçmiş zaman adımı noktası sayısı 1 hafta, 2 hafta ve 3 hafta olacak şekilde verilmiştir ve gelecek 28 zaman adımı için çok adımlı zaman serisi tahmini yapılmıştır. Deneysel sonuçlar incelendiğinde pencere boyutu büyüdükçe modelin başarısı artarken tahmini yapılan zaman adımı arttıkça modelin başarısı azalmaktadır. “t+1” anı için en başarılı sonuç küme bazında ayırma yönteminde, 2,175 RMSE değeri ile RNN modelinde elde edilmiştir. Veri bazında ayırma yönteminde ise 6 model test edildiğinde en başarılı sonuç 2,728 RMSE değeri ile RNN modelinde sağlanmıştır. Çok adımlı zaman serisi

tahmininde ise en başarılı sonuç, küme bazında ayırma yöntemi ile pencere boyutu 3 hafta seçildiğinde, “t+28” zaman adımı için 4,156 RMSE değeri ile BiLSTM modelinde elde edilmiştir.



KAYNAKLAR

- [1] Esling P., Agon C., (2012), “Time-Series Data Mining”, ACM Computing Surveys (CSUR), 45 (1), 12:1-12:34.
- [2] Shen Z., Zhang Y., Lu J., Xu J., Xiao G., (2019), “A novel time series forecasting model with deep learning”, Neurocomputing, 396 (10), 302-313.
- [3] Yaşar H., (2019), “Zaman Serileri Analizi ve Derin Öğrenme Modelleri Kullanarak Amerikan Doları/Türk Lirası Döviz Kuru İçin Hibrid Tahmin Modeli”, Yüksek Lisans Tezi, Doğu üniversitesi.
- [4] Janardhanan D., Barrett E., (2017), “CPU Workload forecasting of Machines in Data Centers using LSTM Recurrent Neural Networks and ARIMA Models”, International Conference for Internet Technology and Secured Transactions, Cambridge, United Kingdom, 11-14 December.
- [5] Kaya E., (2019), “Zaman Serileri Analizinde Box-Jenkins Yöntemi ile Savunma Sanayi Verileri Üzerine Bir Uygulama”, Yüksek Lisans Tezi, Karamanoğlu Mehmetbey üniversitesi.
- [6] Akyüz H. E., (2018), “Vektör Otoregresyon (VAR) Modeli ile İklimsel Değişkenlerin İstatistiksel Analizi”, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 10 (2), 183-192.
- [7] Sevinç A., Kaya B., (2021) “Derin Öğrenme ve İstatistiksel Modelleme Yöntemiyle Sıcaklık Tahmini ve Karşılaştırılması”, Avrupa Bilim ve Teknoloji Dergisi, (28), 1222-1228.
- [8] Güdelek M. U., (2019), “Zaman Serisi Analiz ve Tahmini: Derin Öğrenme Yaklaşımı”, Yüksek Lisans Tezi, TOBB ekonomi ve teknoloji üniversitesi.
- [9] Kumar J., Goomer R., Singh A. K., (2018), “LSTM-RNN Based Workload Forecasting Model For Cloud Datacenters”, 6th International Conference on Smart Computing and Communications, 676-682, Kurukshetra, India, 7-8 December.
- [10] Okcu B., Kalkan, H. (2022), “Estimation of Memory Resource Utilization with Time Series Analysis”, Innovations in Intelligent Systems and Applications Conference (ASYU), 1-5, Antalya, Turkey, 07-09 September.
- [11] Malhotra P., Vig L., Shroff G., Agarwal P., (2015), “Long short term memory networks for anomaly detection in time series”, European Symposium on Artificial Neural Networks, Bruges, Belgium, 22-24 April.
- [12] Ellefsen A.L., Bjørlykhaug E., Æsøy V., Ushakov S., Zhang H., (2019), “Remaining useful life predictions for turbofan engine degradation using semi-

supervised deep architecture”, Reliability Engineering & System Safety, (183), 240-251.

- [13] Web 1, (2010), <https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq>, (Eriřim Tarihi: 3/12/2022).
- [14] Wang J., Wen G., Yang S., Liu Y., (2018), “Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network”, Prognostics and System Health Management Conference, Chongqing, China, 26-28 October.
- [15] Kızrak M.A., Bolat B., (2019), “Predictive Maintenance of Aircraft Motor Health with Long-Short Term Memory Method”, Biliřim Teknolojileri Dergisi, 12 (2), 103-109.
- [16] Jain A., Satish B., (2009), “Short Term Load Forecasting by Clustering Technique Based on Loads”, IEEE Power & Energy Society General Meeting, Calgary, AB, Canada, 26-30 July.
- [17] Bi J., Li S., Yuan H., Zhou M., (2021), “Integrated Deep Learning Metod for Workload and Resource Prediction in Cloud Systems”, Neurocomputing, 424, 35-48.
- [18] Mehmood T., Latif S., Malik S., (2018), “Prediction of Cloud Computing Resource Utilization”, International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)), Islamabad, Pakistan, 08-10 October.
- [19] Web 2, (2011), https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md, (Eriřim Tarihi: 3/12/2022).
- [20] Borkowski M., Schulte S., Hochreiter C., (2016), “Prediction Cloud Resource Utilization”, 9th IEEE/ACM International Conference on Utility and Cloud Computing, Shanghai China, 06-09 December.
- [21] Gupta S., Dinesh D. A., (2017), “Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks”, IEEE International Conference on Advanced Networks and Telecommunications Systems, 1-6, Bhubaneswar, India, 17-20 December.
- [22] Zhang Z., Li B., Zhao D., Gong F., Lu Q., (2016), “Workload Prediction For Cloud Cluster Using a Recurrent Neural Network”, International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, China, 20-21 October.
- [23] Gao J., Wang H., Shen H., (2020), “Machine Learning BAsed Workload Prediction in Cloud Computing”, 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 03-06 August.

- [24] Malik S., Tahir M., Sardaraz M., Alourani A., (2022), “A Resource Utilization Prediction Model for Cloud Data Centers Using Evolutionary Algorithms and Machine Learning Techniques”, *Applied Sciences*, 12 (4), 1-15.
- [25] Al-Asaly M.S., Bencherif M.A., Hassan M.M, Alsanad A., (2022), “A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment”, *Neural Computing and Applications*, 34, 10211–10228.
- [26] Anupama K.C., Shivakumar B.R., Nagaraja R., (2021), “Resource Utilization Prediction in Cloud Computing using Hybrid Mod”, *International Journal of Advanced Computer Science and Applications*, 12 (4), 373-381.
- [27] Bao L., Yang J., Zhang Z., Liu W., Chen J., Wu J., (2023), “On accurate prediction of cloud workloads with adaptive pattern mining”, *The Journal of Supercomputing*, 79, 160–187.
- [28] Saxena D., Singh A.K., (2020), “Auto-adaptive learning-based workload forecasting in dynamic cloud environment”, *International Journal of Computers and Applications*, 44 (6), 541-551.
- [29] Xu M., Song C., Wu H., Gill S.S., Ye K., Xu C., (2022), “esDNN: Deep Neural Network Based Multivariate Workload Prediction in Cloud Computing Environments”, *ACM Transactions on Internet Technology Journals*, 22 (3), 1-24.
- [30] Yu Y., Jindal V., Bastani F., Li F., Yen I., (2018), “Improving the Smartness of Cloud Management via Machine Learning Based Workload Prediction”, 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23-27 July.
- [31] Brockwell P. J., Davis R. A., (2002), “Introduction to Time Series and Forecasting”, 2nd Edition, Springer.
- [32] Karakoyun E. Ş., (2018), “Derin Öğrenme ile Zaman Serilerinin Gerçek Zamanlı Tahmini”, Yüksek Lisans Tezi, Necmettin Erbakan üniversitesi.
- [33] Web 3, (2022), <https://kardelennerdem.medium.com/>, (Erişim Tarihi: 5/12/2022).
- [34] Falk M., (2012), “A First Course on Time Series Analysis — Examples with SAS”, 1st Edition, Chair of Statistics, University of Würzburg.
- [35] Web 4, (2022), <https://www.itl.nist.gov/>, (Erişim Tarihi: 7/12/ 2022).
- [36] Petropoulos F., Spiliotis E., (2021), “The Wisdom of the Data: Getting the Most Out of Univariate Time Series Forecasting”, *Forecasting*, 3 (3), 478-497.
- [37] Duan Z., Xu H., Huang Y., Feng J., Wang Y., (2021), “Multivariate Time Series Forecasting with Transfer Entropy Graph”, *Tsinghua Science and Technology*, 28 (1), 141-149.

- [38] Web 5, (2022), <http://www.veridefteri.com/>, (Eriřim Tarihi: 13/12/2022).
- [39] Rizwan M., (2011), “Augmented Dickey Fuller Test”, SSRN.
- [40] Web 6, (2022), <https://en.wikipedia.org/>, (Eriřim Tarihi: 20/12/2022).
- [41] Web 7, (2020), <https://www.scribbr.com/>, (Eriřim Tarihi: 20/12/2022)
- [42] Bozdogan H., (1987), “Model Selection and Akaike's Information Criterion (AIC): The General Theory And Its Analytical Extensions”, Psychometrika, 52 (3), 345-370.
- [43] Web 8, (2019), <https://medium.com/>, (Eriřim Tarihi: 20/12/2022).
- [44] Web 9, (2020), <https://www.capitalone.com/tech/cloud/what-is-a-cluster/>, (Eriřim Tarihi: 20/12/2022).
- [45] Web 10, (2020), <https://serverlift.com/blog/topserver-chassis-procurement-variables/>, (Eriřim Tarihi: 23/12/2022).
- [46] Web 11, (2020), <https://www.juniper.net/documentation/>, (Eriřim Tarihi: 23/12/2022)
- [47] Ersever B., Doğru A. B., Dörteller M., (2017), “Büyük ölçekli veri merkezleri için bulut biliřim kullanarak sunucu sanallařtırma”, Gazi Mühendislik Bilimleri Dergisi, 3 (1), 20-26.
- [48] Web 12, (2022), <https://www.vmware.com/products/vrealizeoperations.html/>, (Eriřim Tarihi: 23/12/2022).
- [49] Web 13, (2022), <https://www.mshowto.org/>, (Eriřim Tarihi: 23/12/2022).
- [50] Yüce H., Özkan A. O., (2021), “Normalizasyon Yöntemlerinin Biyomedikal Verilerde Sınıflandırma Performansına Etkisi”, Avrupa Bilim ve Teknoloji Dergisi, (30), 35-43.
- [51] Panigrahi S., Karali,Y., Behera H.S., (2013), “Normalize Time Series and Forecast using Evolutionary Neural Network”, International Journal of Engineering Research & Technology (IJERT), 2 (9), 2518-2522.
- [52] Uyar H., (2018), “Otoregresif Ve Otoregresif Hareketli Ortalamalar Modelleri Kullanılarak Çelik Model Bir Yapının Dinamik Karakteristiklerinin Belirlenmesi”, Yüksek Lisans Tezi, Sakarya üniversitesi.
- [53] Web 14, (2022), <https://www.section.io/engineering-education/multivariate-time-series-using-auto-arima/>, (Eriřim Tarihi: 31/12/2022).

- [54] Web 15, (2022), <https://avys.omu.edu.tr/storage/app/public/vceyhan/109840/VAR.pdf>, (Erişim Tarihi: 01/01/2023).
- [55] Web 16, (2021), <https://www.aptech.com/blog/introduction-to-the-fundamentals-of-vector-autoregressive-models/>, (Erişim Tarihi: 01/01/2023).
- [56] Çınar R., (2019), “Derin Öğrenme Yöntemleri ile Lisans Öğrencilerinin Akademik Performanslarına Dayalı Mezuniyet Tahmini”, Yüksek Lisans Tezi, Osmangazi üniversitesi.
- [57] Yavuz E., (2020), “Konutlarda Enerji Tüketimi Kestirimi İçin Derin Öğrenme Ve Makine Öğrenme Yöntemlerinin Karşılaştırılması”, Yüksek Lisans Tezi, İstanbul üniversitesi.
- [58] Web 17, (2021), <https://mfakca.medium.com/lstm-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-326866fd8869>, (Erişim Tarihi: 02/01/2023).
- [59] Yalman H.İ., Tüfekçi Z., (2022), “Konuşma Tanıma Uygulanan BiRNN, BiLSTM ve BiGRU Modellerinin Performans Değerlendirmesi”, Avrupa Bilim ve Teknoloji Dergisi Özel Sayı 36, 121-127.
- [60] Karakuş B.A., (2018), “Derin Öğrenme Ve Büyük Veri Yaklaşımları İle Metin Analizi”, Doktora Tezi, Fırat üniversitesi.

ÖZGEÇMİŞ

Büşra OKCU Gazi Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği ve Elektrik Elektronik Mühendisliği bölümlerinden mezundur. Yüksek lisans eğitimini Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında tamamlamıştır. Halen TURKCELL’de Altyapı Operasyonları biriminde Bulut Platformları Yöneticisi olarak çalışmaktadır. Araştırma ilgi alanları bulut ve konteyner teknolojileri ve yapay zekadır.



EKLER

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Okcu B., Kalkan H., (2022), “Estimation of Memory Resource Utilization with Time Series Analysis”, Innovations in Intelligent Systems and Applications Conference (ASYU), 1-5, Antalya, Turkey, 07-09 September.

