

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**UZUN KISA SÜRELİ HAFIZA VE GEÇİTLİ YİNELENEN BİRİM İLE**  
**BORSA İSTANBUL 100 ENDEKS DEĞERİ TAHMİNİ ÜZERİNE**  
**BİR UYGULAMA**

**Serdar TURAN**

YÜKSEK LİSANS TEZİ

İstatistik Anabilim Dalı

İstatistik Programı

Danışman

Doç. Dr. İbrahim DEMİR

Temmuz, 2019

**TC**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**UZUN KISA SÜRELİ HAFIZA VE GEÇİTLİ YİNELENEN BİRİM İLE  
BIST100 ENDEKS DEĞERİ TAHMİNİ ÜZERİNE BİR UYGULAMA**

Serdar TURAN tarafından hazırlanan tez çalışması 16.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen bilimleri Enstitüsü İstatistik Anabilim Dalı İstatistik Programı tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Tez Danışmanı**

Doç. İbrahim Demir

Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Doç. İbrahim Demir, Danışman

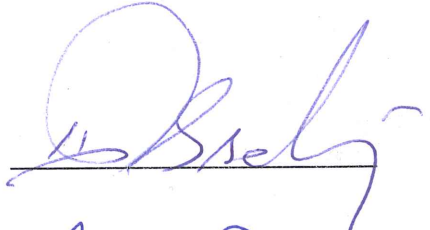
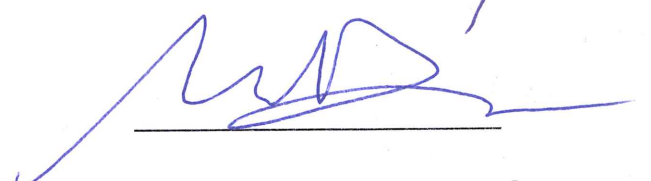
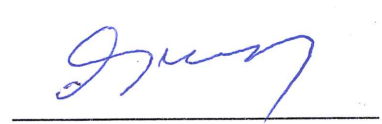
Yıldız Teknik Üniversitesi

Prof. Dr. Mehmet Hakan SATMAN, Üye

İstanbul Üniversitesi

Doç. Dr. Reşit ÇELİK, Üye

Yıldız Teknik Üniversitesi

Danışmanım Doç. Dr. İbrahim DEMİR sorumluluğunda tarafımda hazırlanan Finansal Varlıklarda Fiyat Tahmini başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Serdar TURAN

İmza



*Sevgili kız kardeşim*

*Hatice'ye*

## TEŐEKKÜR

---

Çalıőmalarımnda sürekli bana destek olan ve verdiđim kararlarda arkamda duran aileme teőekkür ediyorum.

Çalıőmamın her aőamasında yardımcı olan tez danıőmanım Doç. Dr. İbrahim DEMİR'e ve yine desteđini hiçbir zaman esirgemeyen Arő. Gör. Hasan Aykut KARABOĐA'ya sonsuz teőekkür ediyorum.

Tanıőmamızın üzerinden kısa bir süre geçmesine rađmen akademik ve profesyonel fikirleri ile bakıő açımını deđiőtiren Sayın Hossein SADR 'a teőekkür ediyorum.

Bu çalıőmayı sevgili anneme ve kardeőlerime ithaf ediyorum.

Serdar TURAN

## İÇİNDEKİLER

<b>SİMGE LİSTESİ</b>	<b>vii</b>
<b>KISALTMA LİSTESİ</b>	<b>viii</b>
<b>ŞEKİL LİSTESİ</b>	<b>ix</b>
<b>TABLO LİSTESİ</b>	<b>x</b>
<b>ÖZET</b>	<b>xi</b>
<b>ABSTRACT</b>	<b>xiii</b>
<b>1 Giriş</b>	<b>1</b>
1.1 Literatür Özeti .....	1
1.2 Tezin Amacı .....	5
1.3 Hipotez .....	5
<b>2 Finansal Piyasalarda Fiyat Tahmini</b>	<b>7</b>
2.1 Finansal Piyasalarda Tahmin İçin Kullanılan Yöntemler .....	7
<b>3 Yapay Sinir Ağları</b>	<b>9</b>
3.1 Yapay Sinir Hücresi Yapısı .....	9
3.2 Yapay Sinir Ağı Öğrenme Yöntemleri .....	12
3.3 Danışmanlı Öğrenme .....	12
3.4 Danışmansız Öğrenme .....	12
3.5 Pekiştirmeli Öğrenme .....	13
3.6 Veri Ölçeklendirme .....	13
3.7 Aktivasyon Fonksiyonu .....	14
3.8 Kayıp Fonksiyonu .....	15
3.9 ADAM Optimizasyonu .....	16
<b>4 Derin Öğrenme</b>	<b>17</b>

4.1 Derin Öğrenme.....	17
4.2 Yinelenen Sinir Ağı (YSA).....	17
4.3 Uzun Kısa Süreli Hafıza.....	19
4.4 Geçitli Yinelenen Birim.....	22
<b>5 Uygulama</b>	<b>24</b>
5.1 Analiz Araçları.....	24
5.2 Model Adımları.....	24
5.3 Veri Ön İşleme.....	26
5.4 Uzun Kısa Süreli Hafıza Bulgular.....	27
5.5 Geçitli Yinelenen Birim Bulgular.....	29
<b>6 Sonuç ve Öneriler</b>	<b>31</b>
<b>A Grafikler</b>	<b>33</b>
<b>B Ağırlıklar</b>	<b>35</b>
<b>Kaynakça</b>	<b>43</b>
<b>Tezden Üretilmiş Yayınlar</b>	<b>47</b>

## SİMGE LİSTESİ

---

$w_{km}$	Ağırlıklar
$\varphi(\cdot)$	Aktivasyon fonksiyonu
$b_k$	Bias değeri
$z_t$	GYB güncelleme fonksiyonu
$r_t$	GYB sınırlama fonksiyonu
$e$	Hata terimi
$x_m$	Modele girdi sinyali
$f_t$	UKSH unutma fonksiyonu
$C_t$	UKSH aday vektörü
$o_t$	UKSH sigmoid fonksiyon çıktısı
$h_t$	YSA çıktı değeri



## KISALTMA LİSTESİ

---

BİST 100	Borsa İstanbul 100 Endeksi
BİST	Borsa İstanbul
ÇG-UKSH	Çok Girdili Uzun Kısa Süreli Hafıza Yöntemi
GYB	Geçitli Yinelenen Birim
KOKH	Kök Ortalama Kare Hata
S&P 500	Standard & Poor's 500
UKSH	Uzun Kısa Süreli Hafıza
YISA	Yinelenen Sinir Ağı
YSA	Yapay Sinir Ağı

## ŞEKİL LİSTESİ

<b>Şekil 3.1</b> Biyolojik Sinir Hücresi .....	9
<b>Şekil 3.2</b> Tek Bir Nörona ile Kurulabilecek Lineer Olmayan Model .....	10
<b>Şekil 3.3</b> Aktivasyon Fonksiyonu Türleri.....	14
<b>Şekil 3.4</b> Tanh Fonksiyonu ve Türevi.....	14
<b>Şekil 4.1</b> Yinelenen Sinir Ağı Yapısı .....	16
<b>Şekil 4.2</b> Yinelenen Sinir Ağı Modeli.....	16
<b>Şekil 4.3</b> Uzun Kısa Frekanslı Model Yapısı .....	20
<b>Şekil 4.4</b> Uzun Kısa Frekanslı Hafıza Modeli .....	20
<b>Şekil 4.5</b> Geçitli Yinelenen Birim Modeli.....	20
<b>Şekil 5.1</b> Ölçeklendirme.....	25
<b>Şekil 5.2</b> Eğitim ve Test Veri Setlerinin Ayrılması.....	25
<b>Şekil 5.3</b> Uzun Kısa Süreli Hafıza Modeli .....	25
<b>Şekil 5.4</b> Geçitli Yinelenen Birim.....	25
<b>Şekil 5.5</b> Kapanış Verisi Histogram Grafiği .....	26
<b>Şekil 5.6</b> Uzun Kısa Süreli Hafıza Fiyat Tahmini .....	28
<b>Şekil 5.7</b> Uzun Kısa Süreli Hafıza Kayıp Değeri.....	28
<b>Şekil 5.8</b> Geçitli Yinelenen Birim ile Fiyat Tahmini .....	30
<b>Şekil 5.9</b> Geçitli Yinelenen Birim Kayıp Değeri.....	30

## TABLO LİSTESİ

---

<b>Tablo 5.1</b> Kapanış Tanımlayıcı İstatistikler .....	26
<b>Tablo 5.2</b> Ölçeklendirilmiş Veri.....	27



# UZUN KISA SÜRELİ HAFIZA VE GEÇİTLİ YİNELENEN BİRİM İLE BORSA İSTANBUL 100 ENDEKS DEĞERİ TAHMİNİ ÜZERİNE BİR UYGULAMA

Serdar TURAN

İstatistik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. İbrahim DEMİR

Fiyat tahmini finansal piyasalar için oldukça önem arz etmektedir. Piyasa profesyonelleri ve araştırmacıların her zaman ilgisini çekmiştir. Bu kapsamda araştırmacılar çeşitlik modeller geliştirmişlerdir. Fiyat tahminde temel analiz, teknik analiz gibi çeşitli yöntemler aktif olarak kullanılıyor olsa da finansal varlıkların doğası gereği tutarlı sonuçlar elde etmek oldukça zordur. Gelişen teknoloji ile beraber artan bilgisayarların işlem güçleri de artmış, buda finansal piyasalarda fiyat tahmini için makine öğrenmesi algoritmalarının kullanılabilmesine olanak sağlamıştır.

Bu çalışmada makine öğrenmesinin bir alt dalı olan derin öğrenme yöntemleri ile finansal varlıklar için fiyat tahmininde bulunulması amaçlanmıştır. Çalışmada Borsa İstanbul 100 endeksine ait 2001-2018 yıllarına arasında günlük kapanış değerleri kullanılmıştır. Bu değerlerden yola çıkarak endeksin gelecekteki yönü ve değeri

tahmin edilmeye çalışılmıştır. Tahmin yöntemi olarak yinelenen sinir ağı algoritmaları olan uzun kısa süreli hafıza ve geçitli yinelenen birim yöntemleri kullanılmıştır. Elde edilen sonuçlar üzerinden bu iki yöntem birbiriyle karşılaştırılmış ve geçitli yinelenen birim yönteminin daha başarılı tahmin sonuçları verdiği gözlemlenmiştir.

**Anahtar Kelimeler:** Derin öğrenme, hisse senedi tahmin, uzun kısa süreli hafıza, geçitli yinelenen birim, makine öğrenmesi



---

YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

## **An Application on Prediction of BIST100 Index Value with LSTM and GRU**

Serdar TURAN

Department of Statistics

Master of Science Thesis

Advisor: Assoc. Prof. Dr. İbrahim DEMİR

Price forecasting is very important for financial markets. It has always attracted the attention of market professionals and researchers. In this context, researchers have developed various models. Such as basic analysis and technical analysis are used actively in price estimation, it is difficult to obtain consistent results due to the nature of financial assets. The computing power of computers increased with the technology, which enabled the use of machine learning algorithms for price forecasting in financial markets.

In this study, it is aimed to forecast prices for financial assets by using deep learning methods which is a sub-branch of machine learning. Daily closing values of Borsa İstanbul 100 Index between 2001-2018 were used in this study. Based on these values, the future direction and value of the index was tried to be estimated. Long – short term memory and gated recurrent unit methods which are recurrent neural network algorithms are used as estimation method. These two methods were compared with each other and it was observed that the gated recurrent unit method yielded more successful estimation results.

**Keywords:** Machine learning, deep learning, stock prediction, lstm, gru



---

YILDIZ TECHNICAL UNIVERSITY  
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

### 1.1 Literatür Özeti

Finansal zaman serilerinde tahmin yapmak gürültülü ve yarı-etkin piyasa biçimi sebebiyle oldukça zordur (Fama, Efficient capital markets: A review of theory and empirical work, 1970). Hisse senetleri, endeksler, vadeli işlem sözleşmeleri gibi finansal varlıkların fiyatları lineer olmayan davranışsal özelliklere sahiptir. Bu sebeple bu varlıkların fiyat tahminlerinde sıklıkla makine öğrenmesi yöntemlerine başvurulur (Karaboğa & Demir, 2013). Son yıllarda yapılan çalışmalar makine öğrenmesi tekniklerinin lineer olmayan finansal veriler üzerinde etkili tahmin yapabileceklerini göstermiştir (G. Zhang, 1998).

Yapay sinir ağları insanın öğrenme yöntemini nöronların model alınarak kopyalanması ile oluşturulan tahmin yöntemleridir. Bu yöntemler makine öğrenmesinin temelini oluştururlar (Mohammed, Khan, & Bashier, 2016). Derin öğrenme ise makine öğrenmesinin spesifik bir alt dalıdır. Derin öğrenme yöntemleri bir bütün olarak portföy yönetiminde kullanılabilirler. Heaton ve arkadaşları derin öğrenmenin portföy yönetiminde kullanımının standart yöntemlere nazaran daha etkin olduğu görüşünü savunmuşlardır. Portföy oluşturma, risk yönetimi ve finansal varlıkların fiyatlanmasında bu yöntemlerin kullanılmasının daha yararlı olacağını ortaya sürmüşlerdir (Heaton, Polson, & White, 2016).

Finansal zaman serilerinin tahmininde makine öğrenmesi ve derin öğrenme ağları farklı metotlar ve farklı varyasyonlar ile kullanılmış ve geleneksel metotlarla veya birbirleriyle karşılaştırmalar yapılmıştır. Bu çalışmalar ile metotların başarıları ölçülmeye çalışılmıştır (Kim, 2003) (Akira, Yoshihara, Matsubara, & Uehara, 2016) (Lavrenko, Schmill, Lawrie, & Ogilvie, 2000) (Bick, Kraft, & Munk, 2013).

Hassan ve arkadaşları (2007) hisse senedi fiyat tahmini için Gizli Markov Modeli, Genetik Algoritma ve Yapay Sinir ağır içeren bir model geliştirmişlerdir. Bu modelin



başarısını test etmek içinse ARIMA gibi geleneksel modellerle karşılaştırma yapmışlardır. Bu karşılaştırma sonucunda önerdikleri modelin geleneksel modeller kadar iyi performans gösterdiği sonucuna ulaşmışlardır.

Ghosh (2012) çalışmasında finansal zaman serilerini Gradient Descent Learning algoritması ile tahmin etmeye çalışmıştır. Gradient Descent Learning algoritmasını Geri yayılım yöntemi ile geliştirerek oluşturduğu modelin başarılı olduğu sonucuna ulaşmıştır.

Fischer ve Krauss (2017) çalışmalarında Uzun Kısa Süreli Hafıza modelini kullanarak 1992-2015 tarihleri arasında S&P 500 hisselerinin verileri üzerinden bir model oluşturmuşlardır. Oluşturdukları modelde sonucunda UKSH modelinin, Random Forest, Deep Neural Net ve Lojistik Regresyon modellerine göre daha başarılı olduğu sonucuna ulaşmışlardır.

Shen ve arkadaşları (2018) Geçitli Yinelenebilir Birim algoritmasının finansal zaman serisi tahmininde kullanımı üzerine yaptıkları çalışmada S&P 500, Hang Seng Indexes (HSI) ve Deutscher Aktienindex(DAX) endeksi hareketlerini tahmin etmeye çalışmışlardır. Bu çalışma için endekslere ilişkin 1991-2007 arası fiyat verilerini kullanmışlardır. Çalışma sonucunda Geçitli Yinelenebilir Birim algoritması ile oluşturulan modellerin kıyaslama değeri olarak kullandıkları destek vektör makineleri yöntemine göre daha etkili olduğu sonucuna ulaşmışlardır.

Ding ve arkadaşları (2015) yaptıkları çalışmada hisse senetlerinin bilgi ve fiyat hareketlerine duyarlılıklarını vurgulamış ve bu verileri kullanarak bir derin öğrenme ile S&P 500 endeksi fiyat hareketlerini tahmin etmeye çalışmıştır. Çalışmalarında haberleri ve olayları kullanarak uzun ve kısa dönemli olayları ayrı ayrı modellemişler ve fiyat hareketleri üzerinden yapılan tahminle karşılaştırmışlardır. Bu karşılaştırma sonucunda endeksi %6 oranında daha doğru tahmin edebildikleri sonucuna ulaşmışlardır.

Patel ve arkadaşları (2015) makine öğrenmesi ile hisse senetleri ve hisse fiyat endekslerini tahmin etmek için trend deterministik veri kullanmışlardır. Çalışmalarında iki hisse senedi ve CNX Nifty ve S&P Bombay Stock Exchange (BSE) Sensex fiyat endekslerini tahmin etmek için Yapay Sinir Ağı (ANN), Destek Vektör Makineleri, Random Forest ve Naive-Bayes algoritmalarını kullanmışlardır. Çalışma

sonucunda Random Forest ile kurdukları modelin diğer üç modele göre daha başarılı sonuç verdiklerini gözlemlemişlerdir.

Chong ve arkadaşları hisse senedi piyasasını derin öğrenme ile analiz etmek ve tahmin etmek için 3 katmanlı bir derin sinir ağı oluşturmuşlardır. Bu ağı oluştururken özellik çıkartma yöntemi olarak temel bileşen analizi, autoencoder ve Boltzmann makinesini kullanarak 7 farklı dizi çıkartmışlardır. Daha sonra modeli Kore hisse senetleri piyasasında test etmiş ve derin yapay ağın finansal piyasaları tahminde kullanılabileceğini önermişlerdir.

Qiu ve arkadaşları (2016) yapay sinir ağları kullanarak Japon hisse senedi piyasası üzerinde hisse senedi getiri tahmininde bulunmuşlardır. Genetik algoritma ve benzetilmiş tavlama (simulated annealing) yöntemlerini kapsaya hibrit bir model oluşturmuşlardır. Bu yöntemi geri yayılım öğrenme algoritması ile karşılaştırmışlar ve önerdikleri modelin daha etkin tahminde bulunduğu sonucuna ulaşmışlardır.

Egeli ve arkadaşları (2003) Türkiye hisse senetleri piyasasını tahmin için makine öğrenmesi yöntemlerinden çok katmanlı algılayıcı ve genel ileri doğru ağı kullanmışlar ve elde ettikleri sonuçları hareketli ortalamalar ile karşılaştırmışlardır. Bu çalışma sonucunda makine öğrenmesi yöntemlerinin hareketli ortalamalara göre daha iyi bir performans gösterdikleri sonucunu elde etmişlerdir.

Moghaddam ve arkadaşları (2016) NADAQ hisse senedi endeksini tahmin etmek için geri yayılım algoritması ile ileri beslemeli yapay sinir ağlarını kullanmışlardır. Tek Adım Sekant yönteminin başarılı sonuç verdiğini belirtmişlerdir.

Hiransha ve arkadaşları (2018) çok katmanlı algılayıcı, yinelemeli yapay sinir ağı, uzun kısa süreli hafıza ve geleneksel sinir ağı ile Hindistan Ulusal Hisse Senetleri Borsası ve New York Borsasında işlem gören hisse senetlerinin fiyatlarını tahmin etmeye çalışmışlardır. Oluşturdukları modelleri lineer bir yöntem olan ARIMA ile karşılaştırmışlardır. Karşılaştırma sonucunda makine öğrenmesi yöntemlerinin tahminde daha başarılı olduğu sonucuna ulaşmışlardır.

Kara ve arkadaşları (2011) BIST 100 endeksini tahmin etmek için yapay sinir ağı ve destek vektör makinesi metotlarını kullanmış ve bu metotların tahmin etme başarılarını karşılaştırmışlardır. Çalışma kapsamında 1997 ve 2007 yılları arasında gerçekleşen kapanış fiyatlarını kullanmışlardır. Çalışma sonucunda yapay sinir

ağlarının, destek vektör makinesi yöntemine göre daha başarılı tahminde bulunduğu sonucuna ulaşmışlardır.

Rather ve arkadaşları (2015) hisse senetlerinin getirilerini tahmin etmek için yinelenen sinir ağlarına dayalı bir hibrit model önermişlerdir. Hibrit model exponential smooting ve autoregressive moving reference lineer modellerini kapsamaktadır. Rather ve arkadaşları bu hibrit modeli aynı yöntemler kullanılarak oluşturulmuş lineer modellerle karşılaştırıldığında daha etkin sonuç verdiğini belirtmişlerdir

Huy ve arkadaşları (2017) yaptıkları çalışmada hisse senedi hareketlerini yapay sinir ağları ile tespit etmişlerdir. Çalışmalarında S&P 500 endeks değerini tahmin etmişlerdir. Bu çalışma sonunda ise iki yönlü geçitli yinelenen birim tahmin modelini tanıtmışlardır.

Roondiwala ve arkadaşları (2017) NIFTY 50 endeksinin tahmini için 01.01.2011 ve 31.12.2016 tarihleri arasında günlük bazda endeks verilerini endeksin gelecekteki fiyatını tahmin etmek için kullanmıştır. Tahmin modeli olarak yinelenen yapay sinir ağı ve uzun kısa süreli hafıza yöntemini kullanmış ve sonuç olarak bu modelin tahmin için analistlere, yatırımcılara veya araştırmacılara gelecek için iyi bir bilgi kaynağı olacağını belirtmişlerdir.

Shah (2007) çalışmasında hisse senetleri fiyatlarının düşüş ve yükselişlerini makine öğrenmesi yöntemleri ile tahmin etmeye çalışmıştır. Uygulamasında destek vektör makineleri ile birleştirilmiş boosting yönteminin tahmin edici sonuçlar verdiği sonucuna ulaşmıştır. Choudhry ve Garg (2008) destek vektör makineleri ile Genetik Algoritmaları birleştirerek hisse senedi fiyat tahmini yapmışlardır. Çalışmalarında destek vektör makinesinin tek başına kullanılmasına nazaran genetik algoritma ile birleştirilmesinin daha etkin sonuçlar verdiğini belirtmişlerdir.

Li ve arkadaşları (2008) çalışmalarında çoklu girdiği uzun kısa süreli hafıza yöntemini tanıtmışlardır. Çin borsasında işlem gören hisse senetleri üzerinde yaptıkları çalışma da hisse senedi fiyatlarına etki eden faktörleri bulmaya çalışmışlardır. Kullandıkları ÇG-UKSH yöntem ile bu orijinal UKSH yöntemine göre gürültüleri ve etken faktörleri bulmakta başarılı olduklarını belirtmişlerdir.

Niaki ve Hoseinzade (2013) yaptıkları çalışmada S&P 500 endeksinin yönünü yapay sinir ağları ile tahmin etmeye çalışmışlardır. Bu çalışma için 27 farklı ekonomik göstergeden yararlanmışlar ve yapay sinir ağları yönteminin logit modellere göre daha başarılı olduğu sonucuna ulaşmışlardır.

Bildirici ve Ersin (2009) Borsa İstanbul'da makine öğrenmesi, otoregresif modellerle birleştirilerek fiyat tahmininde bulunmuştur. Boyacıoğlu ve Avcı (2010) bir bulanık mantık ve makine öğrenmesi yöntemi olan yapay sinir-bulanık çıkarım sistemi ile Bist100 endeksinin dönüş hareketlerini tahmin etmiştir.

Li ve arkadaşları (2017), uzun kısa süreli hafıza yöntemini kullanarak endeks değeri tahmininde bulunmuşlardır. Modelde yatırımcıların olası tepkilerini analiz edebilmek için forumlardan elde ettikleri yatırımcı görüşlerini Naive Bayes'i ile sentimental analize tabi tutmuşlardır. Önerdikleri modeli diğer modeller ile karşılaştırmış ve modelin diğerlerine göre başarılı olduğu sonucuna ulaşmışlardır.

Tayyar ve Tekin (2013) çalışmalarında Borsa İstanbul 100 endeksinin hareket yönünü tahmin etmek için destek vektör makinelerini kullanmışlar. 1995-2013 yılları arasında endekse ait günlük, haftalık ve aylık verileri kullanmış ve bu veri setleri üzerinde 4 model oluşturmuşlardır. Yaptıkları kıyaslamada ilk oluşturdukları modelin yaklaşık %83 başarı ile endeksin artış yönünü tahmin ettiğini gözlemlemişlerdir.

Literatürde son yıllarda makine öğrenmesi ve özellikle derin öğrenmenin finansal analizde sıklıkla kullanıldığı gözlemlenmiştir.

## **1.2 Tezin Amacı**

Bu çalışma ile derin öğrenme yöntemlerinin finansal varlıkların fiyatlarını tahmin etmede etkili olup olmadıklarının araştırılması amacıyla yapılmıştır. Araştırmada Borsa İstanbul 100 endeksinin fiyat tahmini için derin öğrenme yöntemlerinden uzun kısa süreli hafıza ve geçitli yinelenen birim modellerinin efektif olup olmadığı araştırılmak istenilmiştir.

## **1.3 Hipotez**

Arařtırmada endeks fiyat tahmini için UKSH ve GYB yöntemleri kullanılmıřtır. Uygulama kapsamında metotlar ile ayrı ayrı fiyat tahminleri yapılarak bu sonuçlar karşılařtırılmıř ve başarıları incelenmiřtir.

Bu alıřmada ařađıdaki sorulara yanıt aranmıřtır

- UKSH modeli ile Borsa İstanbul Bist 100 endeks fiyatı tahmin edilebilir mi?
- UKSH modeli ile Borsa İstanbul Bist 100 endeks fiyatı tahmin edilebilir mi?
- UKSH ve GYB modellerinden hangisi Bist 100 endeks fiyatı arařtırma için belirlenen veriler üzerinden daha etkili bir řekilde tahmin edebilir?



### 2.1 Finansal Piyasalarda Tahmin İçin Kullanılan Yöntemler

Finansal piyasalarda fiyat tahmini için farklı yöntem kullanılmaktadır. Bu yöntemler;

- Temel Analiz
- Teknik Analiz
  - İstatistiksel Öğrenme Teorisi
  - Momentum Etkisi
- Makine Öğrenmesi
- Ekonometrik Analiz

başlıkları altında toplanabilirler.

Temel analiz hisse senetleri fiyat tahminlerinde piyasa şartlarına göre yoğunlukla kullanılan yöntemlerden bir tanesidir. Bu yöntemde hisse senetlerine ilişkin piyasada oluşan veriler toplanır, çeşitli analiz yöntemleri ile analiz edilir ve yorumlanır. Bu yöntemde hisse senetlerini etkileyecek tüm faktörler ele alınır. Bunlar genel ekonomik, politik, sektörle veya şirkete yönelik her türlü faktör olabilir. Temel analiz ile şirkete ilişkin piyasa değeri tahmini yapılır. Hisse senedi ile yapılacak yatırım türünü de baz alarak son fiyatın bu piyasa değerinden düşük veya yüksek olması durumuna göre işlem yapılır (Karslı, 2004).

Hisse senetlerinin piyasa değerlerinin belirlenebilmesi için ekonomi, sektör ve firma analizi olmak üzere 3 farklı aşamada çeşitli analizler yapılmaktadır (Afşar & Afşar, 2010).

Etkin piyasa hipotezine göre finansal varlığın fiyat hareketi, bu fiyat hareketini tahmin etmede kullanılabilir (Fama, 1970).

Teknik analiz hisse senetlerinin geçmişteki fiyat hareketlerini tekrarlayacağı varsayımına dayanmaktadır. Bu yöntemde fiyat hareketlerini tahmin etmek için

fiyat ve fiyatla ilişkili veriler kullanılarak ortalamalar, göstergeler ve endeksler ile tahminde bulunulur (Perşembe, 2001).

İstatistiksel öğrenme teorisi, verilerin incelenmesi ve tahmin edilmesi için istatistiksel yöntemleri kullanır. Bu teorinin teknik analizi teknik analizin bir alt başlığı gibi değerlendirmemizin amacı oluşacak fiyat verilerinin gelecekte oluşacak fiyat verileriyle ilişkili olduğunu varsaymasıdır. Teori olasılık modelleri ile gelecek hakkında fiyat tahmininde bulunulması temeline dayanmaktadır (Bousquet, Boucheron, & Lugosi, 2004).

Momentum etkisi, kısa ve orta vadede finansal varlıkların getirilerinin gelecekteki hareketlerinin yönünün bir göstergesi olduğu temeline dayanmaktadır (Hong & Stein, 1999). Bu teoriye göre varlıkların 3-12 ay aralığında pozitif değişim göstermesi ileriki 3-12 aylık periyotta da bu yönde bir eğilim göstereceğinin, negatif değişim göstermesi negatif eğiliminin devam edeceğinin göstergesidir.

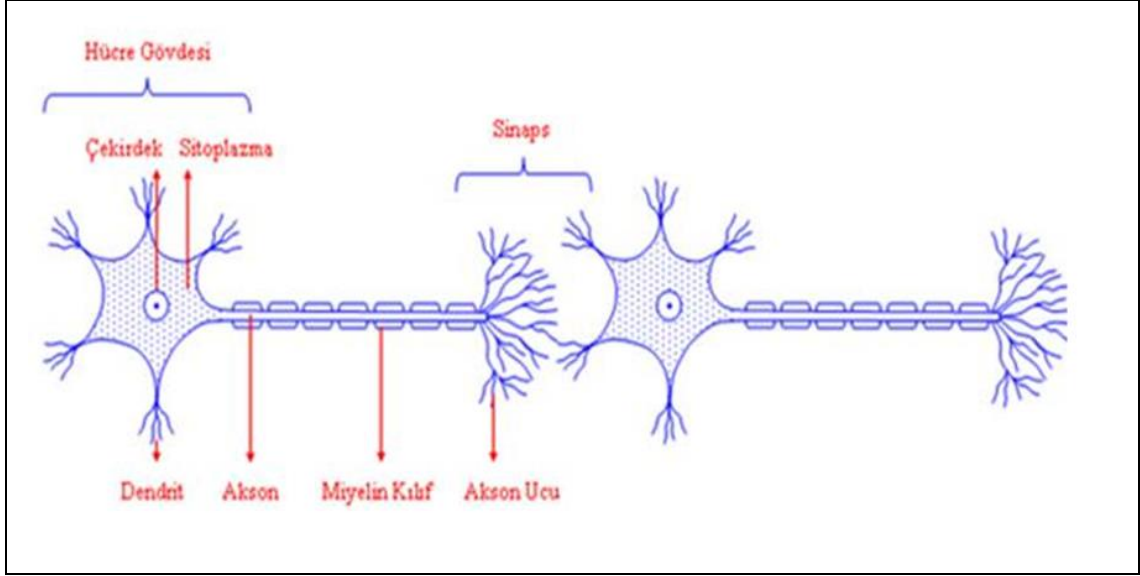
Makine öğrenmesi hisse senetleri fiyat tahmini ve alım satım işlemlerinde makine öğrenmesi yöntemlerinin kullanılması temeline dayanmaktadır. Makine öğrenmesi insanın öğrenme yöntemini taklit ederek modele giren verilerden anlamlı sonuçlar çıkartmayı hedeflemektedir. Algoritmaya dayalı alım satımlar ve buna bağlı olarak makine öğrenmesi yöntemleri ile fiyat tahmini yöntemleri 1990'larda Amerika Hisse Senetleri piyasasında kullanılmaya başlanmıştır (Boehmer, Fong, & Wu, 2012).

Yapay sinir ağları ile ilgili ilk çalışma McCulloch ve Pitts tarafından 1943 yılında gerçekleştirilmiştir. McCulloch ve Pitts çalışmasında sinir ağlarını temel alarak bir hesaplama yöntemi geliştirmiştir (McCulloch & Pitts, 1943). Clark ve Farley 1954 yılında yaptıkları bir çalışmada 1949 yılında psikolog Hebb'in ortaya attığı sinirsel öğrenme mekanizmasına dayanan Hebbian Öğrenme Kuralını taklit eden bir algoritma oluşturmuşlardır (Farley & Clark, 1954) (Hebb, 1949). Daha sonra Hebb öğrenme kuralına ek olarak Hopfield Kuralı, Delta Öğrenme Kuralı, Kohonen Öğrenme Kuralı bunlara örnek gösterilebilir (Karaboğa & Demir, 2013).

### **3.1 Yapay Sinir Hücresi Yapısı**

Temel olarak bir yapay sinir ağının çalışma mantığının anlaşılabilmesi için biyolojik sinir hücresinin yapısının anlaşılması gerekmektedir. Biyolojik sinir ağı dendrit, akson ve hücre gövdesi olmak üzere 3 ana bölümden oluşmaktadır. Biyolojik sinir hücresinde dendritlerin ve aksonların görevi elektrik sinyallerinin hücreye giriş çıkışını sağlamaktır. Hücrede dendritler bilgiyi gövdeye aktarmaktadır. Aksonlar ise gövdede işlenen bilgileri diğer hücrelere aktarmaktadır. Biyolojik sinir hücresi şekilde incelenebilmektedir.





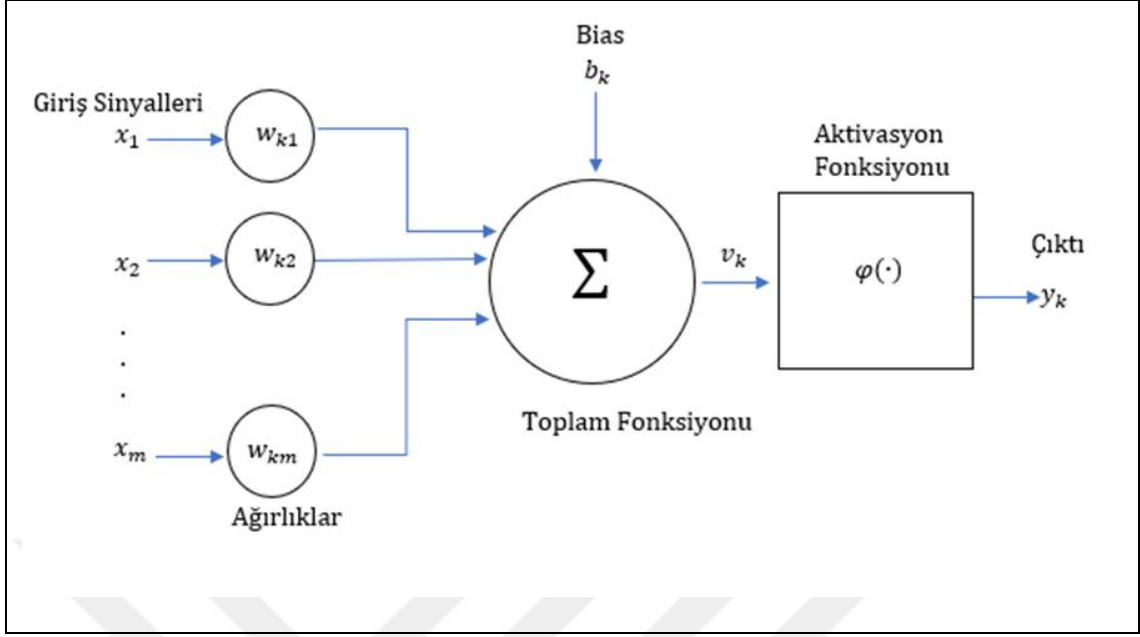
**Şekil 3.1** Biyolojik Sinir Hücresi

Yapay sinir ağları temel olarak biyolojik sinir ağlarının bilgi işleme sistemlerini baz alarak oluşturulmuştur. Temel olarak aşağıdaki süreçleri kapsamaktadır:

1. Bilgi işleme nöronlar üzerinden gerçekleşir.
2. Sinyaller nöronlar arasındaki bağlantılar aracılığıyla birbirine aktarılır.
3. Her bağlantının aktarılan sinyalin değerini belirleyen bir ağırlığı vardır.
4. Her nöron bir sonuç elde etmek için girdiye bir aktivasyon fonksiyonu uygular.

Bu bileşenlere göre bir yapay sinir ağı nöronlar arası bağlantı yapısı (mimari), bağlantıların ağırlıklarının belirlenmesi (eğitim, öğrenme veya algoritma) ve aktivasyon fonksiyonu ile oluşturulmaktadır (Fausett, 1994).

Bir nöron, bir sinir ağının bilgi işleme birimidir (Haykin, 2009). Tek bir nörona ait yapay sinir ağı modeli Şekil 2.'de incelenebilmektedir. Lineer olmayan model tek katmanlı McCulloch-Pitts modeline yanmaktadır.



**Şekil 3.2** Tek Bir Nörona ile Kurulabilecek Lineer Olmayan Model

Şekil 3.2.'de yer alan nöronu matematiksel olarak aşağıdaki eşitlikler ile ifade edebiliriz.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.1)$$

$$y_k = \varphi(u_k + b_k) \quad (3.2)$$

Denklemden  $x_1, x_2, \dots, x_m$  modele giren sinyallerdir,  $w_{k1}, w_{k2}, \dots, w_{km}$  ise sinyallerin ağırlıklarını vermektedir.  $\varphi(\cdot)$  aktivasyon fonksiyonudur.  $y_k$  model sonucu elde edilen çıktıyı vermektedir. Burada modele etki eden bir başka değişken ise bias ( $b_k$ ) değeridir.

Bu modelin çalışma sistemini incelersek;

$x_1, x_2, \dots, x_m$  ham verileri modele alınır. Bu değerler  $w_{k1}, w_{k2}, \dots, w_{km}$  ağırlıkları ile çarpılır. Toplama işlevi ile bir araya getirilir ve toplam değer aktivasyon fonksiyonundan geçirilerek çıktı üretilir. Modelin yapısına bias değeri modelde belirli hücreleri aktif veya pasif edebilir.

Modelde aktivasyon fonksiyonu ağırlıklandırılmış girdiler üzerinden çıktı üreten temel fonksiyondur. Bu ağırlıklandırma modele dahil olan girdilerin katsayıları

olarak düşünülebilir. Bu ağırlıkları yorumlamak mümkün olmasa da regresyon analizindeki  $\beta$  katsayısı gibi düşünülebilir (Elmas, 2016). Modele dahil edilen bir diğer etken bias (eşik) değeridir. Bias değeri modele dahil edilen önsel bilgidir (Shalev-Shwartz & Ben-David, 2014). Hücrenin aktif veya pasif olmasını belirler (Karaboğa & Demir, 2013).

Yukarıda tek nöronlu bir yapay sinir ağının yapısını ve bu yapının bileşenlerine ilişkin tanımlar verilmiştir. Sinir ağları tahmin edilmek istenen değerlere göre tek çıktı veya daha çok çıktı verebilirler. Buna ek olarak bu ağlar tek katmanlı veya çok katmanlı olabilir. Çok katmanlı sinir ağları yukarıdaki modele ek olarak gizli katmanlar içermektedir.

### **3.2 Yapay Sinir Ağı Öğrenme Yöntemleri**

Öğrenme yöntemlerine göre yapay sinir ağları danışmanlı öğrenme, danışmansız öğrenme, pekiştirmeli öğrenme olarak 3 ana başlık altında incelenebilir. Bir yapay sinir ağında öğrenme yöntemini daha çok verinin özellikleri belirler.

### **3.3 Danışmanlı Öğrenme**

Danışmanlı öğrenme yönteminde etiketli veri üzerinden model kurulmaktadır. Bu yöntemde girdi ve çıktılar modele veri seti içerisinde aynı anda verilmektedir. Burada ve diğer makine öğrenmesi modellerinde beklenen ileride olası bir girdi için olası bir çıktının maksimum başarı ile tahmin edilebilmesidir. Model hali hazırda gerçekleşmiş veriler üzerinden tahminlerde bulunarak doğru çıktıyı elde etmek için ağırlıkları günceller. Model ile ağa verilen veriler üzerinden bir genelleme yapılması ve bunun olası problem uzayında yer alan veya alabilecek diğer girdiler için kullanılması amacını taşımaktadır. Burada modele danışmanlı öğrenme adının verilmesinin sebebi veri setinin etiketli olmasından kaynaklanmaktadır. Bir başka deyişle eşitlik oluşturulurken tahmin edilen çıktılar gerçekleşmiş çıktılar üzerinden kontrol edilmekte ve ağırlıklar güncellenmektedir.

### **3.4 Danışmansız Öğrenme**

Danışmansız öğrenmede, danışmanlı öğrenmeye nazaran kullanılan veri setinde hedef çıktılar almamaktadır. Burada ağ birbirine benzeyen veri grupları üzerinden

sınıflandırmalar ve gruplandırmalar yapmaktadır. Ağdan elde edilen her çıktı aynı zamanda bir girdi olarak kullanılmaktadır.

Danışmansız öğrenme daha çok sınıflandırma ve kümeleme problemlerinin çözümünde kullanılmaktadır.

Adaptif Rezonans Teorisi, Grossberg Öğrenme, Öz Örgütlemeli Harita Ağları, danışmansız öğrenme yöntemlerine örnek olarak gösterilebilir.

### 3.5 Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmede ağ deneme yanılma yoluyla öğrenmektedir. Ağa giren veriler olumlu veya olumsuz bir sinyal ile değerlendirilir ve ağ bu şekilde öğrenir.

### 3.6 Veri Ölçeklendirme

Yapay sinir ağları ölçeklendirilmemiş veri setlerine duyarlıdır (Hochreiter & Schmidhuber, 1997). Verileri ölçeklendirme ile verilerin normalleştirilmesi verilerin eğitilmesi işlemini hızlandırma ve geliştirmektedir (Ciaburro & Venkateswaran, 2017). Chen ve arkadaşları yaptıkları çalışmada normalleştirmenin tahmin yapmada etkili olduğunu göstermiştir (Chen, Zhou, & Dai, 2015). Z-skor normalleştirme, Minimum Maksimum Normalleştirme, Medyan ve Medyan Mutlak Sapma yöntemleri normalleştirme için kullanılan yöntemlerden bazılarıdır. Çalışmanın uygulama bölümünde Minimum Maksimum Ölçeklendirme kullanılmıştır.

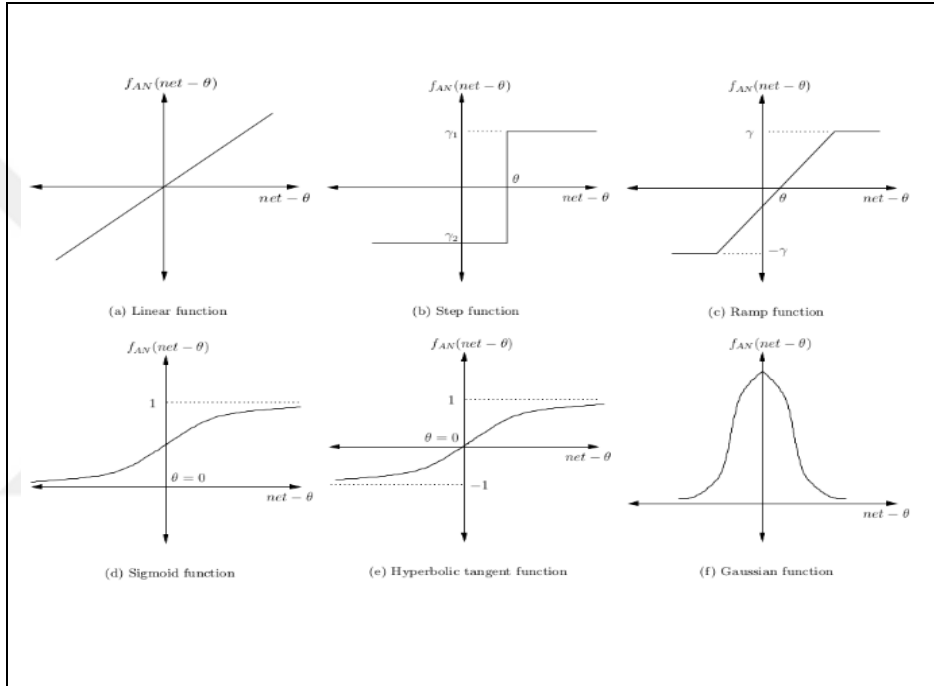
Minimum Maksimum Normalleştirme veri setindeki her veriden veri setinde yer alan minimum değer farkının veri setindeki minimum ve maksimum değer farkına oranlanması ile hesaplanmaktadır. Minimum maksimum ölçeklendirmeyi aşağıdaki eşitlikle ifade edebiliriz.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3.3)$$

Burada  $x_i$  veri setinde yer alan her bir veriyi,  $\min(x)$  veri setindeki minimum veriyi,  $\max(x)$  ise maksimum veriyi göstermektedir. Bu yöntem veri setinde yer alan her bir veriyi 0 ve 1 aralığında bir değere dönüştürecektir.

### 3.7 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu hücreye gelen bilgileri işleyerek bu bilgilere karşılık gelen sonuçları üretmektedir. Aktivasyon fonksiyonu uygulanmadığı durumda model lineer bir model gibi davranacaktır. Aktivasyon fonksiyonu lineer veya lineer olmayan bir fonksiyon olabilir. Aktivasyon fonksiyonu aynı zamanda transfer fonksiyonu olarak da bilinmekte ve çıkış boğumunda yer almaktadır. Bu fonksiyon 0 ve 1 veya -1 ve 1 aralığında değerler üretmektedir.



**Şekil 3.3** Aktivasyon Fonksiyonu Türleri(Turing Finance,2019)

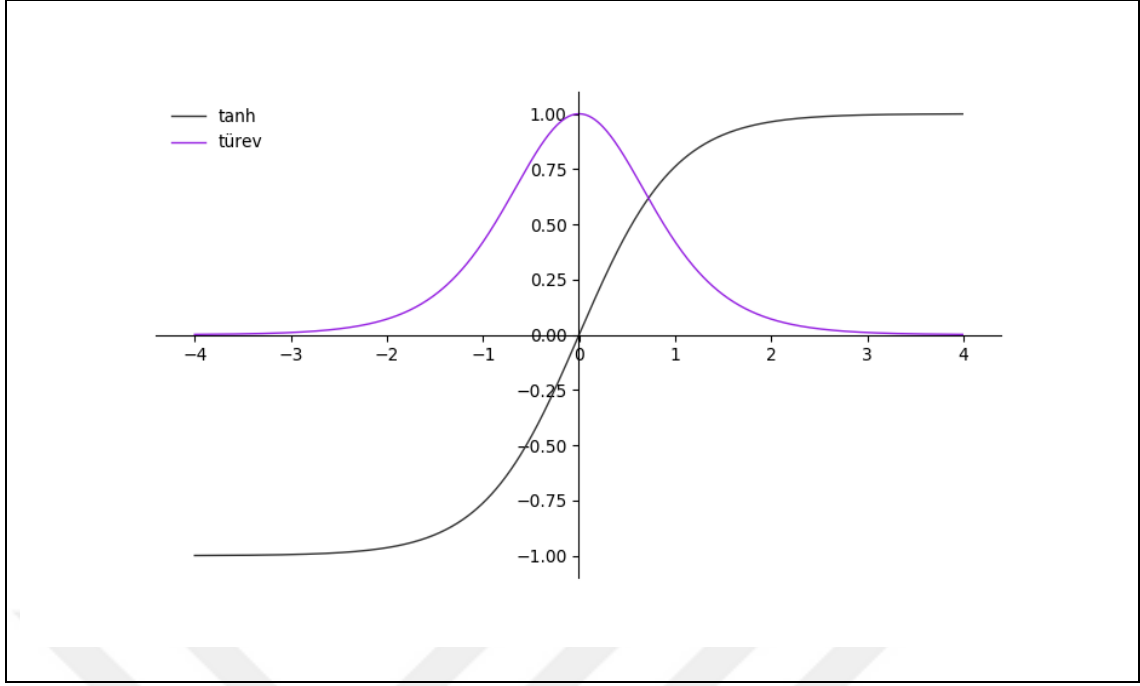
Uzun Kısa Süreli Hafıza ve Geçitli Yinelenen Birim modellerinde tanh aktivasyon fonksiyonu kullanılmaktadır.

$$\tanh(x) = 2\sigma(2x) - 1 \quad (3.4)$$

Eşitlik 3.4 'te sigmoid fonksiyonu eşitliğe eklediğimizde;

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (3.5)$$

eşitliği ile tanımlanabilir. tanh fonksiyonu ve türevi şekilde yer almaktadır.



**Şekil 3.4** Tanh Fonksiyonu ve Türevi

tanh fonksiyonu sigmoid fonksiyonun ölçeklendirilmesi ile elde edilmiştir. Gradyan değeri sigmoid fonksiyondan büyüktür. Buda gradyan yok olması problemi ile daha az karşılaşılmamasına katkı sağlamaktadır.

### 3.8 Kayıp Fonksiyonu

Kayıp fonksiyonu aynı zamanda hata fonksiyonu olarak adlandırılır. Hata fonksiyonu ile model ile tahmin edilen sonuçların gerçek sonuçlara göre farklılığını vermektedir. Kayıp fonksiyonu model performansının bir göstergesidir. Aşağıdaki eşitlik ile gösterilebilir.

$$e = y - \hat{y} \quad (3.5)$$

Burada e hata değeri, y gerçek sonucu,  $\hat{y}$  ise tahmin değerini vermektedir. Makine öğrenmesinde hata kareler ortalaması, ortalama mutlak hata, ortalama eşik değeri hata, hata kareleri toplamı, ortalama yüzde hata fonksiyonları sıklıkla kullanılmaktadır.

Ortalama Hata

$$OH = \frac{1}{n} \cdot \sum_{i=1}^n e_i \quad (3.6)$$

Ortalama Mutlak Hata

$$OMH = \frac{1}{n} \cdot \sum_{i=1}^n |e_i| \quad (3.7)$$

Ortalama Yüzde Hata

$$OYH = \frac{1}{n} \cdot \sum_{i=1}^n \frac{e_t}{Y_t} \cdot 100 \quad (3.8)$$

Ortalama Mutlak Yüzde Hata

$$OMYH = \frac{1}{n} \cdot \sum_{i=1}^n \left| \frac{e_t}{Y_t} \right| \cdot 100 \quad (3.9)$$

Hata Kareler Ortalaması

$$HKO = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad (3.10)$$

Araştırmada hata kareler ortalaması kayıp fonksiyonu olarak kullanılmıştır.

### **3.9 ADAM Optimizasyonu**

Adam optimizasyonu stokastik gradyan yerine kullanılacak ağırlıklı iteratif yollarla güncellemek için kullanılan bir algoritmadır. Adam algoritması birinci dereceden gradyan tabanlı optimizasyon sağlar. Kingma ve arkadaşları tarafından 2014 yılında tanıtılmıştır. Adam optimizasyonunun uygulaması basittir, daha az belleğe ihtiyaç duyar, durağan olmayan, gürültülü ve seyrek veriler için uygundur. (Kingma, Diederik, & Ba, 2014)

Çalışmada ağırlıkların hesaplamasında adam optimizasyonu kullanılmıştır.

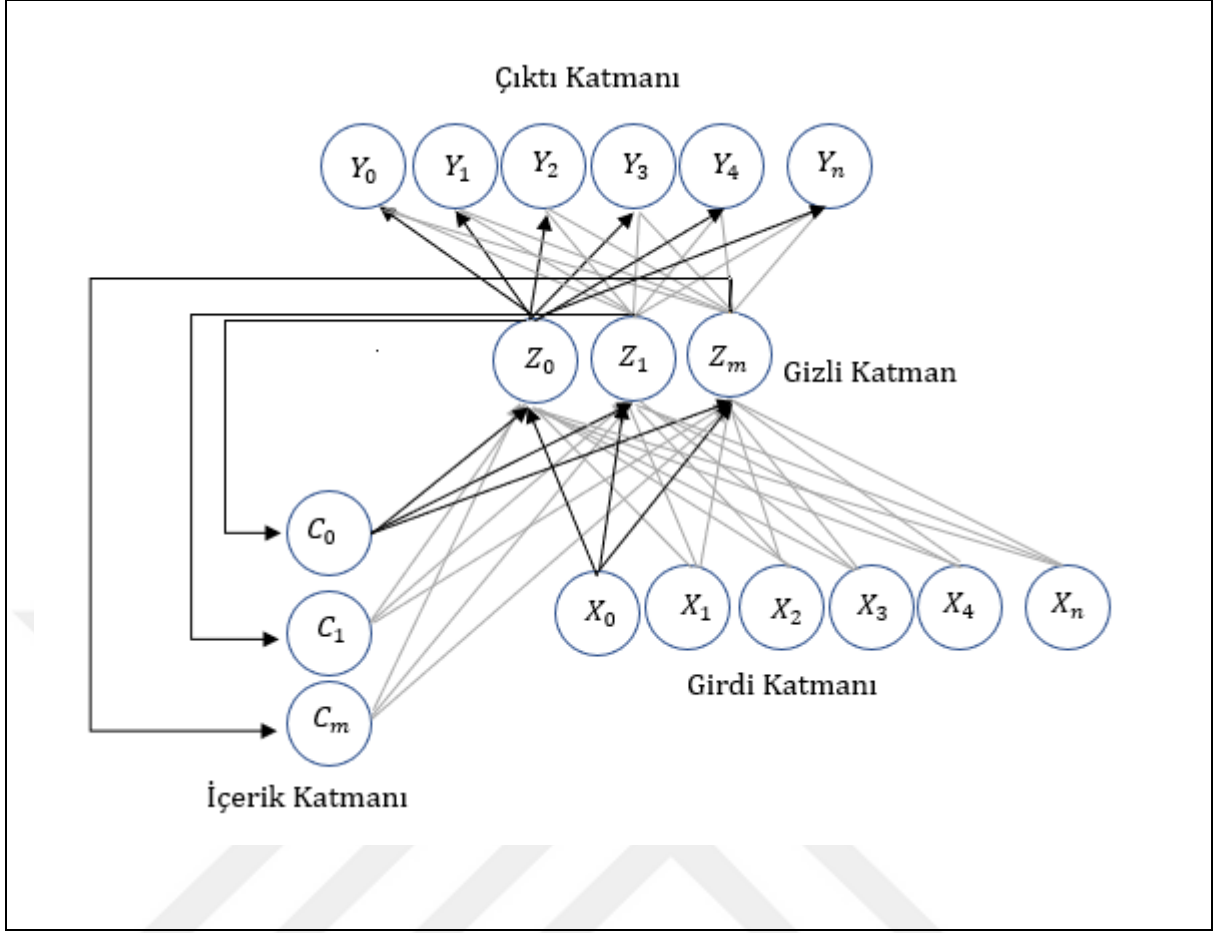
### **4.1 Derin Öğrenme**

Geleneksel makine öğrenmesi yöntemleri derin öğrenme yöntemleri ile karşılaştırıldığında yapıları gereği veri işlemede sınırlı kalmaktadır. Derin öğrenme yöntemleri ise çoklu işlem katmanlarının hesaplamalarda kullanılmasına olanak sağlamaktadır. Bu yöntemler geri yayılım (backpropagation) algoritmasını kullanarak veri kümelerindeki yapıyı keşfedebilmektedir (LeCun, Bengio, & Hinton, 2015).

### **4.2 Yinelenen Sinir Ağı (YSA)**

İnsan nezdinde bilginin sürekliliği vardır. Bir bilginin  $t$  anındaki algısı geçmişte kazanılmış bilgilere dayalı bir süreklilik sağlamaktadır. Örnek olarak bir filmi izlediğimizde bir önceki sahneye göre şu anki sahne arasında bir bağlantı kurarız. Geleneksel yapay sinir ağları bir ön bilgi olmadan bu çıkarımı yapabilmesi oldukça zordur. Yinelenen sinir ağları ise bir dizi içerisinde belirli döngülerle ile bilginin sürekliliğini sağlamaktadır. Yinelenen sinir ağları önceki çıktıyı veya gizlenmiş durumu girdi olarak alabilir. Bir başka deyişle yinelenen sinir ağı tahmin edilmek istenen olayı tahmin edebilmek için modele verilen girdileri yine modele bağımlı olarak ilerleyen adımlarda tekrar tekrar kullanabilir. Yinelenen sinir ağları girdi katmanı, içerik katmanı gizli katmanlar ve çıkış katmanından oluşmaktadır.



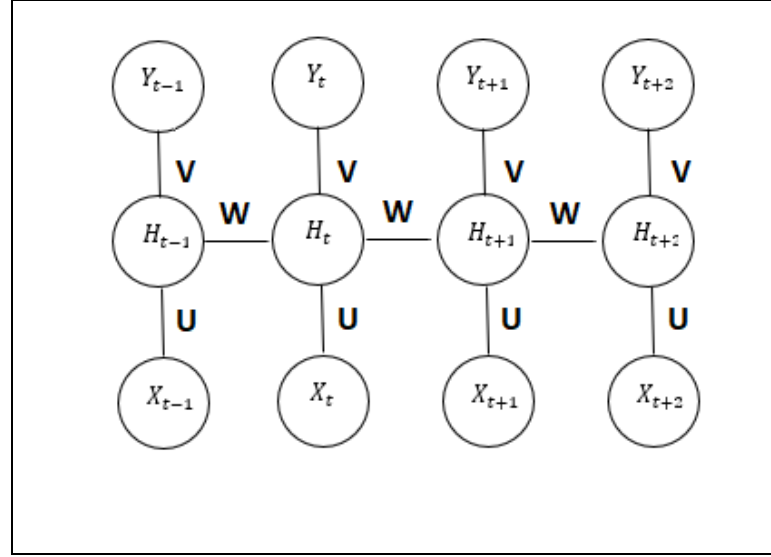


**Şekil 4.1** Yinelenen Sinir Ağı Yapısı

Şekilde yer alan yinelenen sinir ağı çeşitli uzunluktaki dizileri davranış şekillerini öğrenebilir (Elman, 1990).

Yinelenen sinir ağları her biriminde ağırlıklandırılmış ve yine bir ağırlığı olan bağlantıları ile birbirine bağlıdır. Bu birimlerin her biri aktivasyon fonksiyonu ile güncellenmektedir.

Yinelenen sinir ağlarının algoritmasını daha detaylı olarak aşağıda ki şekilde inceleyebiliriz.



**Şekil 4.2** Yinelenen Sinir Ağı Modeli

$$H_t = \sigma(U * X_t + W * H_{t-1}) \quad (4.1)$$

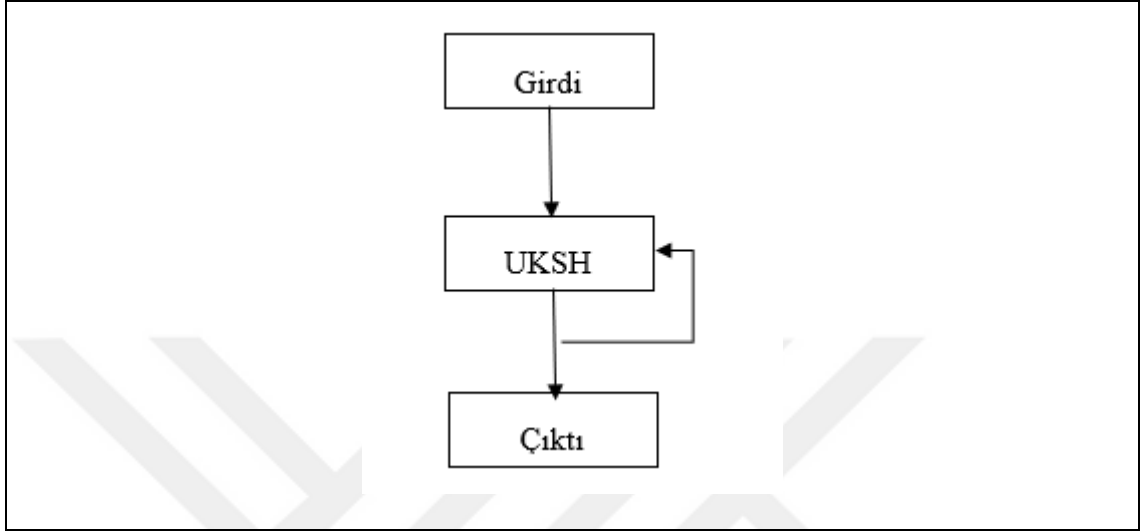
Denklemden  $U$  gizli katmanlara ait ağırlık vektörü,  $V$  çıktı katmanına ait ağırlık vektörü,  $W$  farklı zaman aralıklarında kullanılan aynı ağırlık vektörü bir başka deyişle geçmişten alınan bilgidir.

### 4.3 Uzun Kısa Süreli Hafıza

Uzun kısa süreli hafıza (UKSH), yinelenen ağı mimarisine sahip bir makine öğrenme algoritmasıdır (Hochreiter & Schmidhuber, 1997). İlk olarak Schmidhuber ve Hochreiter tarafından ortaya atılmıştır. Model olarak kısa periyotta öğrenilen bilgiyi depolamaktadır ve bunu uzun periyotta öğrenim için kullanmaktadır. Bu nedenle uzun kısa süreli hafıza gizli katmanlarında "hafıza blokları" adı verilen birimler içermektedir. Bu hafıza blokları geleneksel yinelenen sinir ağlarında yer alan gizli birimler olarak tanımlanabilirler (Gers F., 2001). Hafıza blokları içerisinde bir veya birden fazla hafıza hücreleri içermektedir. Her bir hafıza bloğu bilgi akışını kontrol etmek için giriş ve çıkış kapıları içermektedir. Giriş kapısı hafıza hücresindeki giriş aktivasyon bilgilerinin akışını kontrol ederken, çıkış kapıları çıktı aktivasyon bilgilerinin akışını kontrol eder. Daha sonra hafıza bloklarına "unutma kapısı" eklenmiştir. Unutma kapısı hücrenin tekrarlayan bağlantısı yoluyla girdi aktivasyonu olmadan önce hücrenin iç durumunu ölçeklendirir, hücrenin hafızasını

sıfırlar (Gers, Schmidhuber, & Cumming, 2000). UKSH ağı bu 3 kapıdan oluşmaktadır.

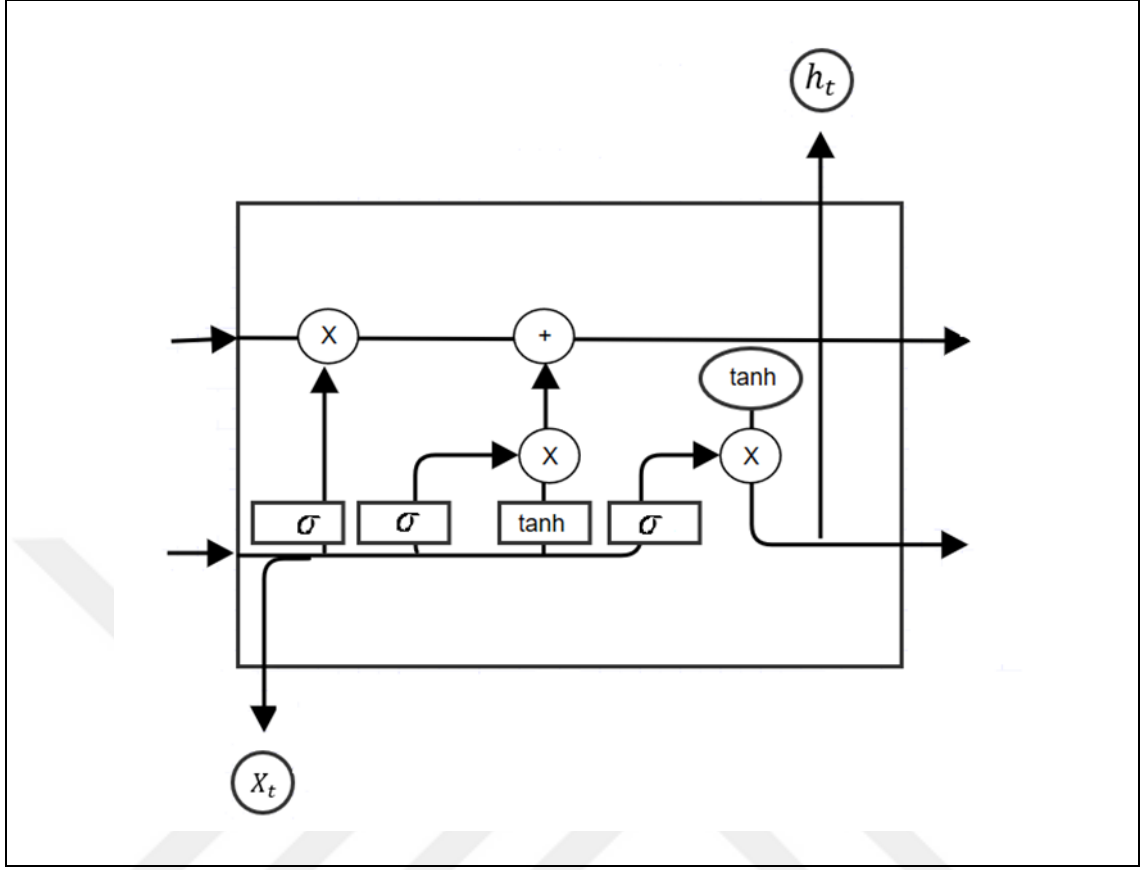
Uzun kısa süreli hafıza modeli yapısı aşağıda yer almaktadır.



**Şekil 4.3** Uzun Kısa Frekanslı Hafıza Model Yapısı

Şekil 4.3'te genel olarak UKSH modelinin genel yapısı bulunmaktadır. Her bir adımda modelden çıktı olarak elde edilen verilerin modele tekrar girebileceği gösterilmektedir.

Uzun kısa süreli hafıza mimarisi, tekrar eden bölümlerden oluşan birbirine bağlı bir ağ şeklindedir.



Şekil 4.4 Uzun Kısa Frekanslı Hafıza Modeli

UKSH modelinin daha iyi anlaşılabilmesi için modelin adımlarının incelenmesi gerekmektedir. UKSH modelin t anında model girdisine  $x_t$ , model çıktısına  $h_t$  dersek oluşturacağımız ağ t anında öncelikle bir önceki modelden gelen çıktıyı sıfırlamalıdır.

$$f_t = \sigma(W_f[h_{t-1}, x_t]) \quad (4.2)$$

Model daha sonra modelde hangi bilginin depolanacağına karar verilmelidir. Bu işlem iki bölümden oluşmaktadır. Önce girdi kapısı katmanı hangi değerlerin güncelleneceğine karar verir. Daha sonra sigmoid katmanı olası yeni değerleri içeren bir vektör oluşturur. Bu işlemlerin sonunda ise bu iki adım birleştirilir ve girdi güncellenir.

$$i_t = \sigma(W_i.[h_{t-1}, x_t]) \quad (4.3)$$

$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t]) \quad (4.4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4.5)$$

Son olarak ađın ıktısına karar verilmektedir. Burada ıktı olarak verilecek sonu sigmoid kapısının karar verilen blmnden oluřturulmaktadır.

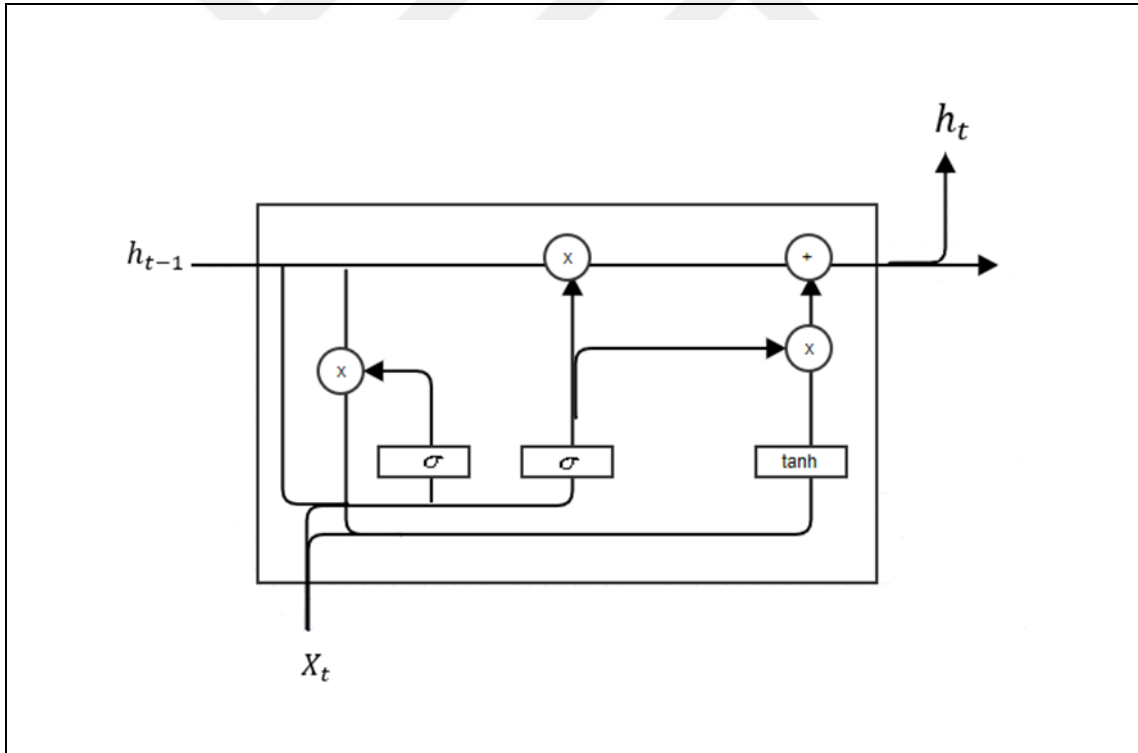
$$o_t = \sigma(W_0[h_{t-1}, x_t]) \quad (4.6)$$

$$h_t = o_t * \tanh(C_t) \quad (4.7)$$

Modelde oluřan ıktı model hcre durumunu temel alan filtrelenmiř ıktıdır.

#### 4.4. Geitli Yinelenen Birim

Geitli Yinelenen Birim Cho ve arkadaşları tarafından ilk olarak 2014 yılında tanıtıldı. Model kaybolan gradient problemini ozmek iin oluřturulmuřtur (Cho, ve diđerleri, 2014). Geitli yinelenen birim UKSH ađının bir varyasyonudur. UKSH aynı metodu izler fakat UKSH modelinden farklı olarak modeldeki kapı sayısı 2'dir. Bu kapılar sıfırlama ve gncelleme kapılarıdır.



**řekil 4.5** Geitli Yinelenen Birim Modeli

UKSH modelinde yer alan girdilerle GYB modelini oluřturacak olursak ncelikle bir gncelleme kapısı oluřurmamız gerekmektedir.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (4.8)$$

GYB modelinden daha sonra sıfırlama kapısı UKSH modelinde yer alan unutma ve girdi kapısını birleştirerek bir sıfırlama kapısı oluşturur.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (4.9)$$

GYB sıfırlama ve güncelleme kapısının modeli sırasıyla model çıktısı için kullanmakta geçmişteki bilgilerin kullanılması sıfırlama kapısı modele uygulanır. Nihai çıktıyı elde etmek için ise modele güncelleme kapısı uygulanmaktadır.

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \quad (4.10)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4.11)$$

GYB modelinde eğitim süreçleri UKSH modeline göre daha hızlıdır.

### 5.1 Analiz Araçları

Çalışmanın analiz kısmında verilerin analizi için Python programlama dili kullanılmıştır. Python programlama dili son yıllarda popülaritesini arttıran, yüksek seviyeli bir dildir. Bilimsel çalışmalar için yoğunlukla kullanılmaktadır (Millman & Aivazis, 2011).

Analizde kullanılacak verilerin yüklenmesi, düzenlenmesi ve temizlenmesi işlemlerini yerine getirmek için Python Pandas, Numpy ve Math kütüphaneleri kullanılmıştır. Bu kütüphaneler veri işleme için sıklıkla başvuru kütüphanelerdir (McKinney, 2013).

Verilerin ölçeklendirilebilmesi için Scikit-learn (sklearn) Python kütüphanesine başvurulmuştur. Scikit-learn, bir arabirim olarak tutarlı ve görev odaklı birçok makine öğrenmesi metodu sunmaktadır (Pedregosa, Varoquaux, Gramfort, Michel, & Thirion, 2011).

Veriler üzerinden model kurma işlemi ise yine Python'ın Keras kütüphanesi ile gerçekleştirilmiştir. Keras, TensorFlow, CNTK veya Theano üzerinde çalışan yüksek seviye bir yapay sinir ağı programlama arayüzüdür (Chollet, 2015).

### 5.2 Model Adımları

Çalışmada aynı veri seti üzerinde UKSH ve GYB modelleri kullanılmıştır.

Araştırmanın adımları özetle aşağıdaki gibidir.

1. Veri seti okunduktan sonra bir takım ön düzenleme sürecinden geçirilmiş ve daha sonra 0 ve 1 aralığında ölçeklendirilmiştir. Ölçeklenen veri seti eğitim ve test veri setlerine ayrılmıştır.

```
olcek = MinMaxScaler(feature_range=(0,1))
bistData = olcek.fit_transform(bistData)
```

**Şekil 5.1** Ölçeklendirme

2. Eğitim veri seti, verinin ilk %80'ini test veri seti ise geriye alan bölümü kapsamaktadır.

```
egitim_buyukluk = int(len(bistData)*0.80)
egitim, test = bistData[0:egitim_buyukluk, :],
                bistData[egitim_buyukluk:len(bistData), :]
```

**Şekil 5.2** Eğitim ve Test Veri Setlerinin Ayrılması

3. Model parametreleri ve eğitim veri seti ile uzun kısa süreli hafıza ve geçitli yinelenen birim modelleri oluşturulmuştur.

```
model = Sequential()
model.add(LSTM(50, input_shape=(1, geçmiş_kontrol)))
model.add(Dense(1))
model.fit(egitimX, egitimY, epochs=10000, batch_size=64, verbose=1)
```

**Şekil 5.3** Uzun Kısa Süreli Hafıza Modeli

```
model = Sequential()
model.add(GRU(50, input_shape=(1, geçmiş_kontrol)))
model.add(Dense(1))
model.fit(egitimX, egitimY, epochs=10000, batch_size=64, verbose=1)
```

**Şekil 5.4** Geçitli Yinelenen Birim



Son olarak modeller test veri seti üzerinden tahminden bulunmuş ve modelin tahmin başarısı ölçülmüş, model ile ilgili sonuçlar alınmış ve model görselleştirilmiştir.

### 5.3 Veri Ön İşleme

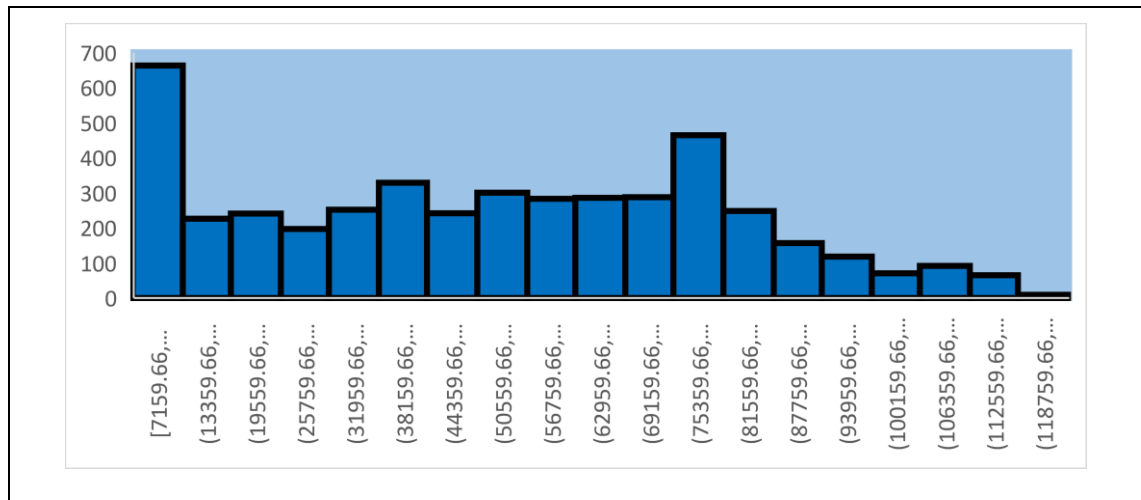
Araştırma kapsamında fiyat tahmini yapmak için öncelikle veri setini normalleştirmemiz gereklidir. Veri setini normalleştirmek için minimum maksimum ölçeklendirme uygulanmıştır. Ölçeklendirme sonrasında değerler 0-1 aralığında yer almaktadır. Bu normalleştirmeden önceki ham kapanış zaman serisinin tanımlayıcı istatistiklerini aşağıdaki tabloda inceleyebiliriz.

**Tablo 5.1** Kapanış Tanımlayıcı İstatistikler

Tanımlayıcı İstatistikler	Değer
Ortalama	52,425.05
Medyan	53,296.54
Standart Sapma	29,242.19
Basıklık	-0.99
Çarpıklık	0.13

2001-2018 yılları arası BIST100 endeksinin kapanış değerlerinin ortalaması 52,425, medyanı 53,296, standart sapma değeri 29,242.19, basıklık değeri -0.99, çarpıklık değeri ise 0.13 olarak hesaplanmıştır.

Kapanış verisine ilişkin histogram grafiği aşağıda incelenebilmektedir.



**Şekil 5.5** Kapanış Verisi Histogram Grafiği

Histogram grafiğine göre fiyatlar çoğunlukla 7,159 ve 13,359 arasında yoğunlaşmıştır.

Analizde kullanılmadan önce veri ölçeklendirilmiştir. Ölçeklendirme işlemi için Minimum Maksimum ölçeklendirme yöntemi kullanılmıştır. Bu yöntem ile yeni oluşan veride yer alan her bir değer 0 ile 1 arasındadır. Ölçeklendirme sonrası verilere ilişkin örneklem aşağıdaki şekilde incelenebilmektedir.

**Tablo 5.2** Ölçeklendirilmiş Veri

Tarih	Veri
2/1/2001	0.020
3/1/2001	0.017
4/1/2001	0.026
.	.
.	.
27.12.2018	0.737
28.12.2018	0.732
31.12.2018	0.739

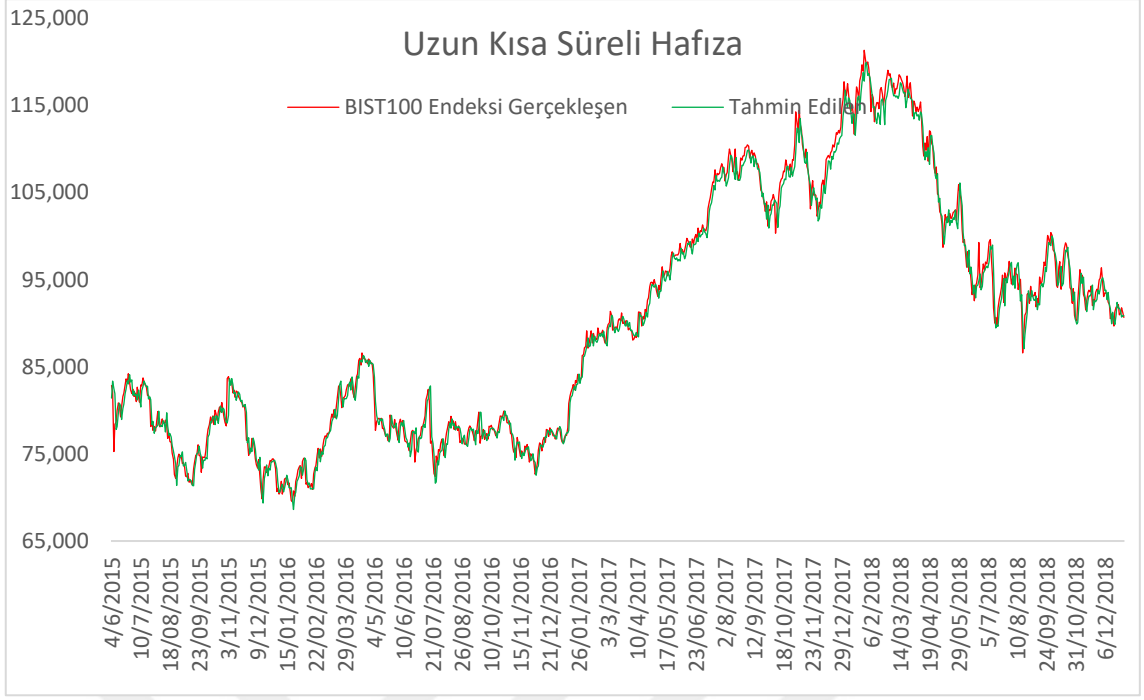
Ölçeklendirme işleminden sonra veri seti eğitim verisi ve test verisi olarak ikiye ayrılmıştır. Bunun için veri seti %80 eğitim, %20 test olarak iki yeni veri seti elde edilmiştir.

Yinelenen sinir ağlarını diğer sinir ağlarından ayırt eden özellik bir hafızaya sahip olmalarıdır. Bunun sağlanabilmesi için oluşturulan veri setleri bir önceki veriyi de girdi alacak şekilde eğitim seti ve test seti için ileri doğru birer tane kaydırılmış yeni veri setleri oluşturulmuştur.

#### **5.4 Uzun Kısa Süreli Hafıza Bulgular**

UKSH modeli için mini küme büyüklüğü kullanılmıştır. Kullanılan küme büyüklüğü değeri 64'tür. Model eğitim seti üzerinde 10000 iterasyon (epochs) ile eğitilmiştir. Modelin kayıp fonksiyonu hata kareler ortalaması ile bulunmuştur.

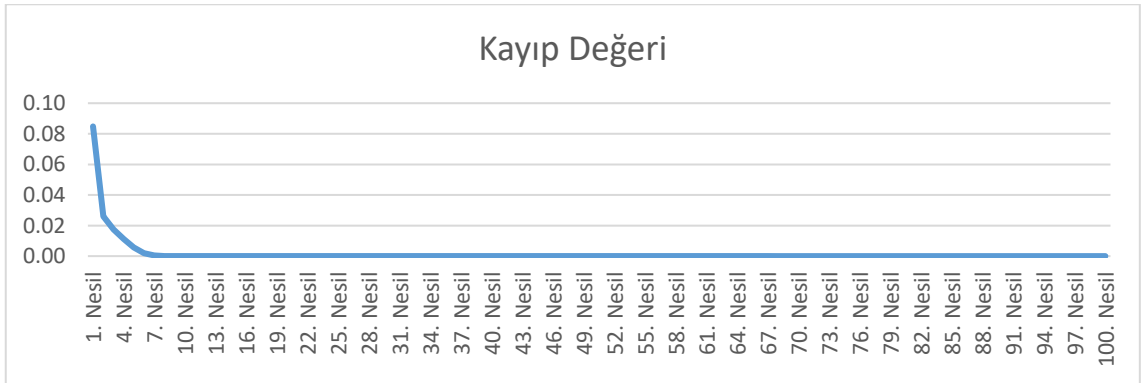
Analizde ilk olarak eğitim verileri ile elde edilen model test verileri ile karşılaştırılmıştır. Bu karşılaştırma Şekil 7. 'de incelenebilmektedir.



**Şekil 5.6** UKSH ile Fiyat Tahmini

Şekil 7. 'de 04.06.2015 ve 12.12.2018 tarih aralığında tahmin verileri ve endeks verileri ile paralel hareket etmektedir. Uzun kısa süreli hafıza yöntemi ile yapılan analizde elde edilen kapanış verileri gerçekleşen tahmin verilerine göre ortalama olarak %1.109 farklılık göstermektedir.

Bunun yanı sıra gerçekleştirilen analiz ile kayıp değerleri hesaplanmıştır. Kayıpları değerlerine ilişkin ilk 100 nesil Şekil 8. 'de incelenebilmektedir.



**Şekil 5.7** Uzun Kısa Süreli Hafıza Kayıp Değeri

Şekil 8.'de yer alan Uzun Kısa Süreli Hafıza kayıp değeri ilk nesilde 0.0848 olarak hesaplanmıştır. Her yeni nesilde kayıp değeri kademe kademe düşmüştür. 100. Nesilde 0.00008 değerine ulaşmıştır.

Analizde kapsamında uzun kısa süreli hafıza yöntemi ile endeks fiyatı yönü tahmin edilmiştir. Burada model test verisinin %83.57 'unu doğru bir şekilde tahmin etmiştir.

## 5.5 Geçitli Yinelenen Birim Bulgular

Geçitli Yinelenen Birim modeli için mini küme büyüklüğü kullanılmıştır. Kullanılan küme büyüklüğü değeri 64'tür. Model eğitim seti üzerinde 10000 iterasyon(epochs) ile eğitilmiştir. Modelin kayıp fonksiyonu hata kareler ortalaması ile bulunmuştur.

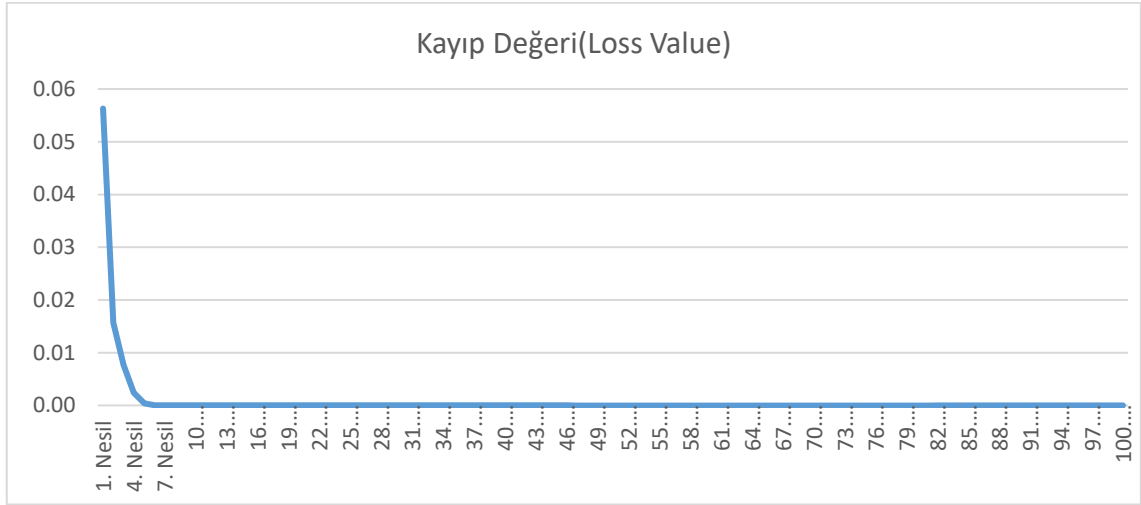
Analizde ilk olarak eğitim verileri ile elde edilen model test verileri ile karşılaştırılmıştır. Bu karşılaştırma şekil. 'de yer almaktadır.



Şekil 5.8 Geçitli Yinelenen Birim ile Fiyat Tahmini

Şekil 9. 'de 04.06.2015 ve 12.12.2018 tarih aralığında tahmin verileri ve endeks verileri ile paralel hareket etmektedir. Uzun kısa süreli hafıza yöntemi ile yapılan analizde elde edilen kapanış verileri gerçekleşen tahmin verilerine göre ortalama olarak %0.4811 farklılık göstermektedir.

Bunun yanı sıra gerçekleştirilen analiz ile kayıp değerleri hesaplanmıştır. Kayıpları değerlerine ilişkin ilk 100 nesil Şekil 10. 'de incelenebilmektedir.



**Şekil 5.9** Geçitli Yinelenen Birim Kayıp Değeri

Şekil 8.'de yer alan Uzun Kısa Süreli Hafıza kayıp değeri ilk nesilde 0. 0.0563 olarak hesaplanmıştır. Her yeni nesilde kayıp değeri kademe kademe düşmüştür. 100. Nesilde 0.0000015 değerine ulaşmıştır.

Analizde kapsamında uzun kısa süreli hafıza yöntemi ile endeks fiyatı yönü tahmin edilmiştir. Burada model test verisinin %83.79 'unu doğru bir şekilde tahmin etmiştir.

Araştırma kapsamında Borsa İstanbul 100 endeksi kapanış verileri kullanarak endeksin bir sonraki gün ne olacağı tahmin edilmeye çalışılmıştır. Çalışmada zaman serisi olarak alınan kapanış verileri derin öğrenme yöntemleri olan uzun kısa süreli hafıza ve geçitli yinelenen birim ile modellenmiştir.

Çalışmada hisse senetleri fiyatlarının ve fiyat hareketlerinin tahmininde uzun kısa süreli hafıza ve geçitli yinelenen birim yöntemlerinin etkili birer tahmin edici olduğu önerilmiştir. Bu kapsamda veriler eğitim ve test verileri olarak iki veri seti olarak bölünmüş ve test verileri tahmin edilmiştir. Elde edilen sonuçlarla araştırma problemlerine yanıt aranmıştır. Uygulama sonucunda uzun kısa süreli hafıza modelinin Borsa İstanbul Bist 100 endeks fiyatı tahmin etmekte etkili bir yöntem olduğu sonucuna ulaşılmıştır. Uzun kısa süreli hafıza yöntemi endeksin yönünü %83.57 başarı oranı ile tahmin etmiştir. GYB modeli ile Borsa İstanbul Bist 100 endeks fiyat yönünü tahmin etmekteki başarısı aynı veri seti için %83.79'dur. Buda geçitli yinelenen birim modelinin uzun kısa süreli hafıza yöntemine göre daha başarılı bir tahmin edici olduğu sonucunu vermektedir.

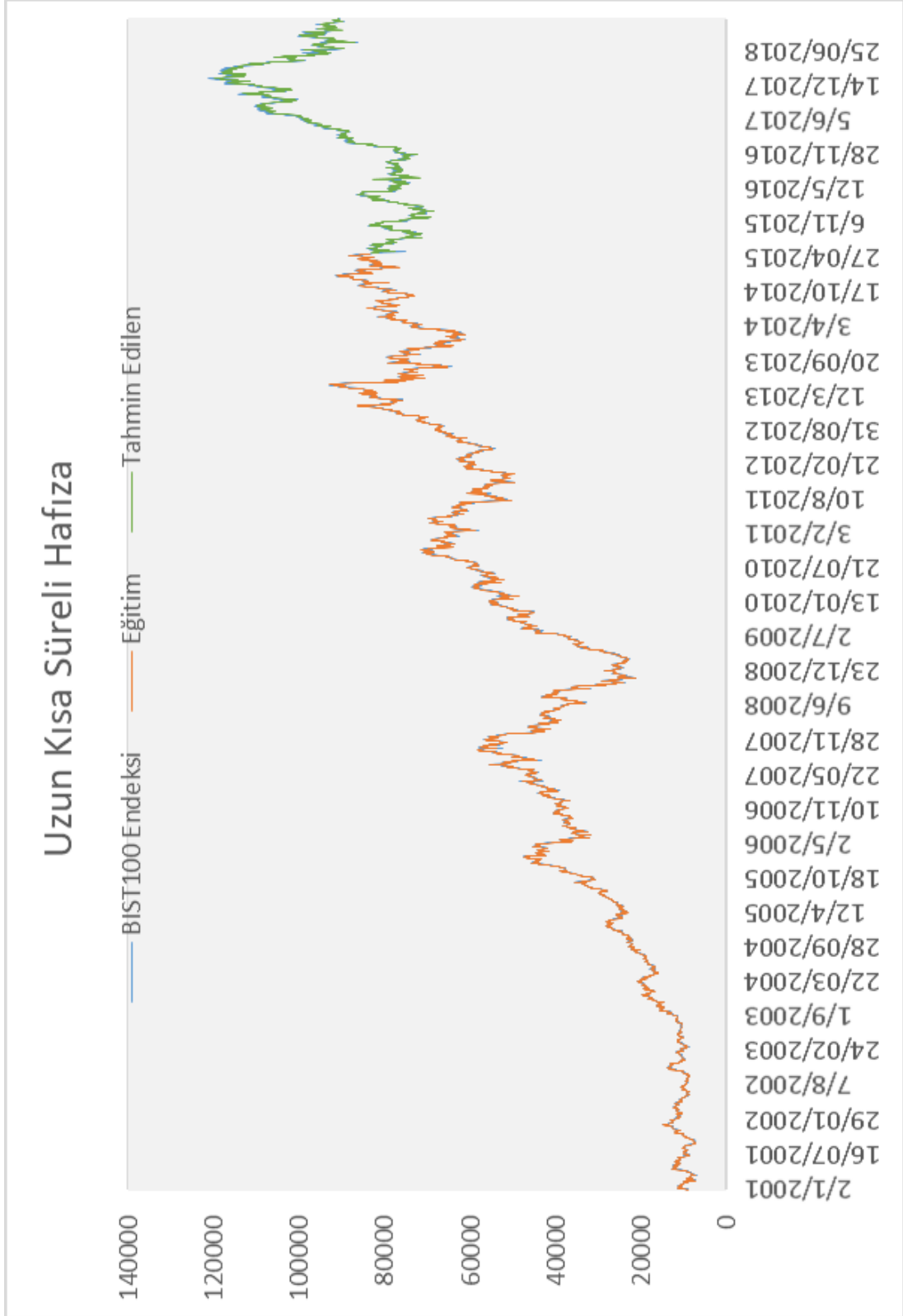
Araştırmada tahmin modellerine ilişkin hata değerleri de hesaplanmıştır. Buna göre kurulan modellerde uzun kısa süreli hafızaya ait ortalama kare hata 23.9, ortalama mutlak hata ise 102.47 olarak hesaplanmıştır. Geçitli yinelenen birime ilişkin hata değerleri ise ortalama kare hata 18.35 ve ortalama mutlak hata 77.83 olarak bulunmuştur.

Araştırma kapsamında elde edilen sonuçlar doğrultusunda bu iki verinin ilgili zaman aralığında Borsa İstanbul 100 endeksinin verilerini tahmin etmedeki başarı oranlarının birbirine yakın olması farklı zaman dilimlerinde ve farklı finansal varlıklarda daha farklı sonuçlar verebileceğini göstermektedir. Hangi modelin daha etkin olduğunun tam olarak elde edilebilmesi için modellerin farklı finansal varlıklarla, farklı zaman dilimlerinde test edilmesi gerekmedir. İlerleyen

dönemlerde arařtırmacılar bu modellerin etkinliđini farklı zaman serilerini kullanarak test edebilirler.

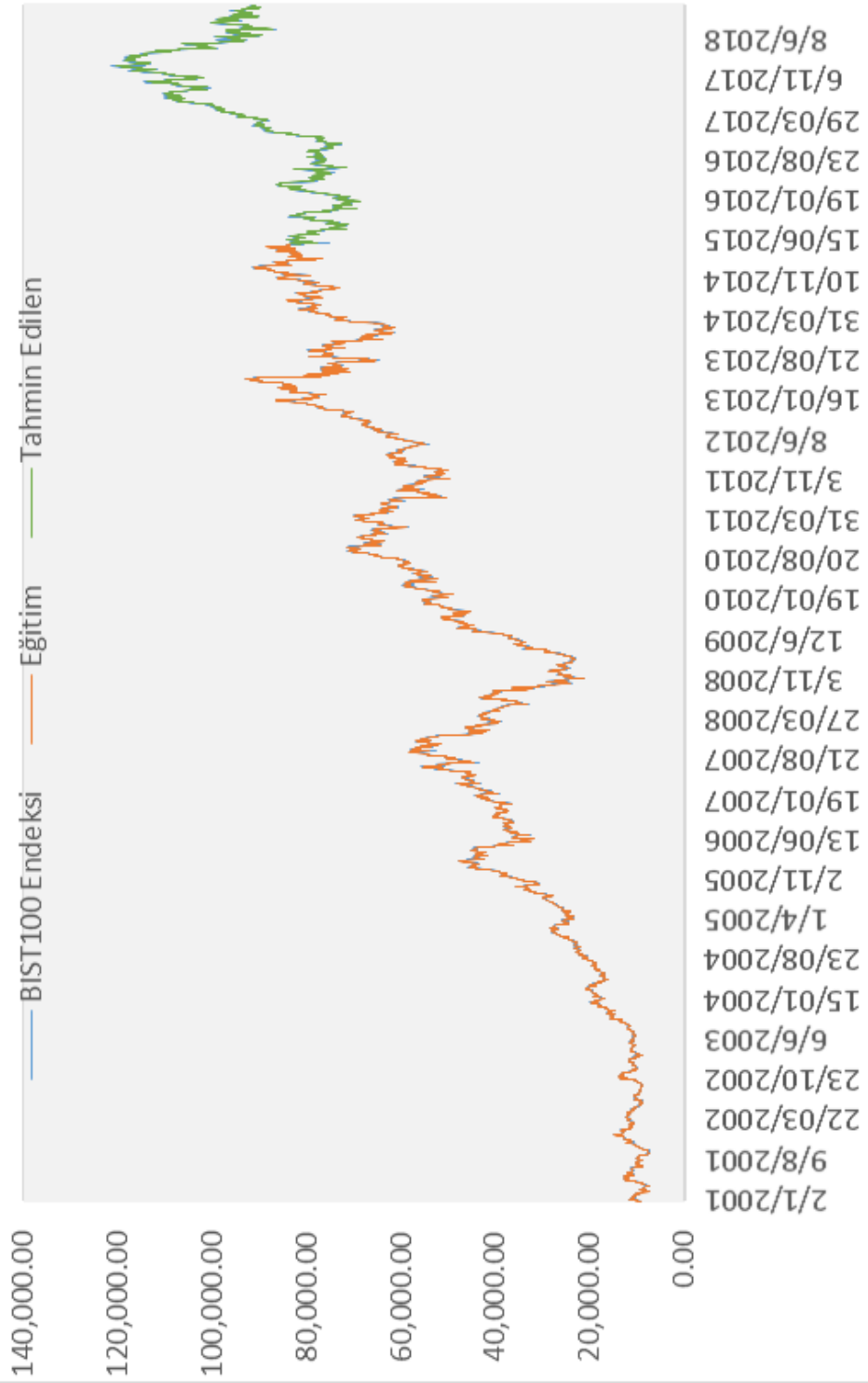
Buna ek olarak model ile oluşturulan betik sürekli çalıştırılarak uzun dönemde başarı oranları kaydedilebilir ve bu başarı oranları modellerin etkinliđinin arařtırılmasında bir karşılaştırma kıstası olarak kullanılabilir.







## Geçitli Yinelenen Birim



### GYB Model Ağırlıkları

Param <tf.Variable 'gru\_1/kernel:0' shape=(1, 150) dtype=float32\_ref>:

```
[[ 0.07531255 -0.01815681 0.69815886 -0.18673706 -0.0479346 0.01427572
 0.11222216 0.05318572 -0.00727765 0.0314401 -0.20080712 -0.0014492
-0.12604947 0.09882112 -0.15241003 0.15187521 -0.16034906 -0.13730215
-0.01451063 0.05961315 0.79182035 0.31429663 0.19207314 -0.04840828
 0.2525242 0.00438332 0.02586333 0.22742005 0.15076481 0.12424108
-0.02848392 -0.21207541 0.15632367 0.01094815 0.1730339 0.20268592
 0.0054297 0.02978343 -0.15775341 -0.07306291 -0.19626217 -0.03611717
-0.02763462 0.02696831 0.2800777 0.2661921 0.05493268 -0.2644313
-0.13264617 -0.29028162 -0.18277656 0.1582382 -0.11158369 0.10072634
 0.14640757 -0.0144673 -0.19240162 -0.05640562 -0.10494804 -0.02216488
-0.11575402 -0.12003688 0.18954468 -0.16503493 -0.05382702 -0.13627157
-0.07157087 -0.10342845 0.10807613 -0.00870611 0.14004508 -0.14278331
-0.15808499 -0.1132892 0.02394646 -0.17267005 -0.16609128 0.09000558
-0.07644676 -0.08975245 0.07888335 -0.17628676 -0.13668433 -0.01057719
 0.05994964 -0.13722417 0.14152583 0.07471853 0.11571172 -0.09132155
 0.08278778 -0.06974053 -0.13143513 0.15986991 -0.11140604 -0.18541574
 0.04221244 0.04625759 0.08410844 0.03561024 0.25548285 -0.24310263
-0.04860272 -0.23743366 -0.05260415 -0.2431451 -0.24558394 -0.25127548
 0.25389332 -0.04933659 -0.24358295 0.05887535 0.2455564 0.18539286
 0.24998662 0.2501263 0.24816321 -0.24478182 -0.23285463 -0.05205545
-0.03896697 -0.05956791 -0.24547246 -0.04832788 0.04758786 -0.24271755
-0.24246459 0.05251903 -0.2492142 -0.25976476 0.24660417 0.24477018
```

```
0.26102084 -0.24850562 -0.05321066 -0.05988514 -0.05351104 -0.23812996
0.21597628 0.05726401 -0.24953836 -0.13390298 0.2436222 0.0591137
-0.05593697 0.05273117 -0.05209075 -0.24548663 0.24366589 -0.2385454 ]]
```

Param <tf.Variable 'gru\_1/recurrent\_kernel:0' shape=(50, 150)  
dtype=float32\_ref>:

```
[[[-0.03649655 0.00021029 0.02517265 ... -0.11844117 0.0594194
0.08561199]
[ 0.02944985 -0.05803664 -0.13304888 ... 0.01399756 -0.0902507
-0.07862375]
[ 0.07111562 -0.12467149 0.07201726 ... 0.09241515 0.17488566
0.04372003]
...
[-0.10048652 -0.0414236 0.07966437 ... 0.04470905 -0.07648795
-0.00135601]
[ 0.02592353 -0.04607272 -0.06184913 ... -0.12624194 0.0772213
-0.02685159]
[ 0.02571239 0.02287751 -0.02451006 ... 0.0725955 0.20992304
-0.07494676]]
```

Param <tf.Variable 'gru\_1/bias:0' shape=(150,) dtype=float32\_ref>:

```
[-1.30889922e-01 -1.00973740e-01 8.74704063e-01 -1.05171882e-01
2.93758899e-01 -1.64647907e-01 -1.61918953e-01 -1.33493125e-01
-1.29558161e-01 3.47613782e-01 -1.36476159e-01 4.01789546e-01
-1.49487987e-01 3.15053850e-01 -1.28549576e-01 -1.61558837e-01
-1.31466180e-01 -1.28220066e-01 -7.44095072e-02 2.87368804e-01
9.07455266e-01 4.07014161e-01 -1.97874606e-01 2.89178997e-01
3.98466498e-01 -1.94897681e-01 -1.45958364e-01 4.02647227e-01
```

-1.56536803e-01 -1.28720880e-01 -1.33317888e-01 -1.32072836e-01  
-1.27904803e-01 -1.34390205e-01 2.68932313e-01 4.10704583e-01  
3.09116304e-01 -1.38756126e-01 1.60069823e-01 3.85347724e-01  
-1.28498092e-01 4.74981219e-01 -9.39900428e-02 5.23767948e-01  
4.28160012e-01 4.59692776e-01 3.02603573e-01 -1.29302815e-01  
-1.65274829e-01 -1.16621844e-01 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00  
-6.49535796e-04 4.17980074e-04 -4.05873880e-02 1.55696354e-03  
-5.92922047e-02 2.64139306e-02 5.47956526e-02 5.69389248e-03  
4.39929310e-03 -4.91163954e-02 2.83705397e-03 6.96628019e-02  
-1.93704497e-02 -4.62488197e-02 -2.66995886e-03 -3.65734324e-02  
2.02183682e-03 -3.67380073e-03 2.77016629e-02 -5.43135814e-02  
-2.76874285e-02 -6.38099611e-02 5.13978191e-02 -5.27198575e-02  
4.66540121e-02 3.89628001e-02 1.92246083e-02 5.47211617e-02  
2.83703823e-02 -5.35227219e-03 -5.52542321e-03 3.19545856e-03

-6.02686126e-03 5.47255762e-03 -5.36215454e-02 -6.71505705e-02  
-5.64139336e-02 6.69730604e-02 -1.66251045e-02 6.79654106e-02  
-3.81590147e-03 4.05637734e-02 3.86480126e-03 7.02717602e-02  
-5.88187128e-02 4.81380112e-02 -5.55763356e-02 -1.23295886e-02  
-2.41519064e-02 -1.41683631e-02]

### **UZUN KISA FREKANSLI HAFIZA MODEL AĞIRLIKLARI**

Param <tf.Variable 'lstm\_1/kernel:0' shape=(1, 200) dtype=float32\_ref>:

[[ -7.24733397e-02 -3.60384560e-03 -8.27194154e-02 -3.36043946e-02  
-7.73211718e-02 1.18683733e-01 -5.69939539e-02 7.97628835e-02  
3.43972921e-01 3.10985632e-02 1.43550009e-01 2.49089807e-01  
1.31035317e-02 1.30785063e-01 1.62458748e-01 1.57615080e-01  
1.32638738e-01 2.16604874e-01 -2.38028741e+00 3.02047245e-02  
3.96216482e-01 5.49632534e-02 2.86194652e-01 1.92699701e-01  
-5.19092023e-01 3.00087750e-01 3.44671130e-01 3.74815971e-01  
-2.10332438e-01 -1.86312333e-01 2.45456621e-01 -1.63655698e-01  
3.93271223e-02 3.58596832e-01 -2.62853391e-02 -1.10539712e-01  
-3.04785252e-01 1.72387004e-01 2.88093418e-01 1.35134250e-01  
3.17336209e-02 8.22391957e-02 3.65985721e-01 1.19194008e-01  
3.62176836e-01 3.65855604e-01 2.58345336e-01 2.64082402e-02  
3.57030518e-03 1.84957683e-01 1.60358608e-01 1.49511069e-01  
-1.17973089e-02 6.99288994e-02 -1.55708030e-01 -1.11372471e-02  
1.23742133e-01 1.34110421e-01 1.10732317e-01 8.58107209e-02  
-1.60261065e-01 -1.59563109e-01 -2.80371159e-02 -1.49906114e-01  
4.34134305e-02 -7.26671517e-03 -1.46490395e-01 1.62932694e-01  
-6.71148300e-04 -1.47976577e-01 -4.39083129e-02 -1.12329200e-01  
-6.38052225e-02 -3.21730822e-02 -8.69597048e-02 3.54538113e-02

8.02309811e-03 5.16875088e-02 -1.11634277e-01 6.57126307e-03  
-1.69102624e-01 -9.65718850e-02 3.89633328e-02 -8.49339887e-02  
7.11744279e-02 -4.26363349e-02 6.87585324e-02 -1.41736209e-01  
8.59546661e-02 5.57856262e-02 -3.41370553e-02 -1.83309764e-02  
1.01264864e-01 2.74312496e-03 5.55001199e-02 5.99814504e-02  
8.87446105e-03 -6.56110048e-03 7.23202378e-02 -1.26405939e-01  
1.94398570e-04 2.47997910e-01 1.79758891e-01 2.97681370e-04  
1.13775830e-04 -5.55749774e-01 -1.75229259e-04 3.47231001e-01  
3.97245824e-01 -2.24381670e-01 -2.07357734e-01 3.06617141e-01  
-5.37156045e-01 7.47083686e-05 -2.78031737e-01 -3.69418502e-01  
-4.43254799e-01 3.01816493e-01 5.77628434e-01 -5.99845080e-05  
-3.93278211e-01 -3.48732263e-01 4.39862788e-01 4.42333251e-01  
-9.09976443e-05 -2.96527416e-01 4.02586222e-01 -3.64706337e-01  
-3.08478600e-04 1.96361143e-04 -4.10746157e-01 1.61467124e-05  
-3.95586848e-01 -4.38380837e-01 4.86629906e-05 6.69839283e-05  
-3.08815535e-04 1.88610986e-01 -5.44209421e-01 4.26635265e-01  
-1.00178426e-04 2.57469833e-01 2.97926188e-01 -3.25586200e-01  
3.60204458e-01 -3.35277915e-01 4.40381438e-01 -2.15957925e-01  
2.52999365e-01 3.04598570e-01 1.39259264e-01 6.85205534e-02  
-1.04318865e-01 5.07226400e-02 -4.09150384e-02 7.84663856e-02  
3.77155654e-02 8.69083107e-02 1.85366303e-01 9.63734761e-02  
2.50997186e-01 3.96257997e-01 1.84142262e-01 1.06579006e-01  
1.28114000e-01 3.45158219e-01 1.96875304e-01 1.98328152e-01  
-2.45184636e+00 -1.28074765e-01 2.87088096e-01 2.21342102e-01  
3.60783190e-01 7.53050968e-02 -3.52020502e-01 4.05394346e-01  
4.10712808e-01 2.23878130e-01 6.25149021e-03 -6.03451952e-02

```
3.99961889e-01 7.53392056e-02 1.56263500e-01 4.17757630e-01
-1.61514401e-01 -1.90687180e-01 -1.18574932e-01 -1.12563506e-01
2.74504930e-01 1.83941767e-01 7.12534264e-02 2.46358126e-01
1.31347090e-01 3.24351043e-01 1.17439024e-01 3.19050610e-01
3.70861769e-01 -1.90733261e-02 4.20302562e-02 3.81308585e-01]]
```

Param <tf.Variable 'lstm\_1/recurrent\_kernel:0' shape=(50, 200)  
dtype=float32\_ref>:

```
[[ 1.12038076e-01 -1.49258971e-02 2.94903871e-02 ... 6.37445180e-03
-7.21255168e-02 -1.04747160e-04]
[ 4.49219495e-02 1.57937244e-01 1.72566436e-02 ... 5.62891848e-02
7.28257839e-03 -1.71712562e-02]
[ 4.15489040e-02 -1.34137213e-01 -1.33929970e-02 ... 7.74409771e-02
-6.68363497e-02 -2.49808189e-02]
...
[-1.46320714e-02 1.04643129e-01 -1.16946697e-02 ... -3.77090387e-02
-3.79736274e-02 -1.82797667e-02]
[-5.39467856e-02 -5.81915528e-02 9.37483646e-03 ... 6.87094778e-02
7.42698386e-02 2.76523288e-02]
[ 6.13527521e-02 -7.52050653e-02 -6.16979823e-02 ... -7.17000514e-02
5.13328314e-02 9.30331126e-02]]
```

Param <tf.Variable 'lstm\_1/bias:0' shape=(200,) dtype=float32\_ref>:

```
[-1.38498485e-01 -1.78704653e-02 -2.34198317e-01 -1.07827462e-01
-3.27364475e-01 3.52956772e-01 -1.98416889e-01 1.63224548e-01
1.21202275e-01 -6.22610711e-02 -1.16838410e-01 9.78248045e-02
```

3.46764594e-01 -2.40638539e-01 5.31113781e-02 1.10796139e-01  
3.16688865e-01 8.11840594e-02 1.15497209e-01 -1.85672075e-01  
1.27163112e-01 1.18727766e-01 1.58981532e-01 1.62787020e-01  
-1.21869162e-01 8.40087831e-02 1.25053748e-01 1.15615726e-01  
-2.17002541e-01 -2.93131292e-01 1.62770003e-01 -1.50546297e-01  
2.31402501e-01 1.43058226e-01 -3.72490101e-02 -2.75759727e-01  
-2.90897131e-01 -1.93977356e-01 3.34981143e-01 1.26013637e-01  
-2.58801877e-01 1.48778306e-02 8.94134641e-02 1.10501826e-01  
1.10183284e-01 1.15371302e-01 1.63012743e-01 -1.07691497e-01  
-7.40585895e-03 9.25103575e-02 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00  
-5.62442801e-05 2.42245896e-03 -9.17129684e-03 -7.99199042e-05  
-3.66459244e-05 1.16415873e-01 5.57225721e-05 -4.73415107e-02  
1.30853737e-02 3.40620615e-02 -8.76718666e-03 1.55144995e-02  
1.14923671e-01 -2.49904042e-05 -9.80577245e-03 -1.97186787e-02



1.06746718e-01 1.06657511e-02 -1.88802168e-01 9.79665128e-06  
-1.47905760e-02 1.35395350e-02 -3.91367683e-03 -2.73382403e-02  
2.89479722e-05 -2.94546261e-02 2.62976717e-02 -1.29916975e-02  
9.30019087e-05 -3.73504045e-05 8.42206739e-03 -5.30829857e-06  
6.67351931e-02 -1.52246831e-02 -1.51125205e-05 -1.93316591e-05  
5.13733030e-05 -5.12499455e-03 8.35913494e-02 8.25497974e-03  
3.29049763e-05 1.01720216e-03 4.39548679e-03 1.54362689e-03  
9.06166434e-03 -1.20617980e-02 -6.47370657e-03 6.75178505e-03  
2.40183016e-03 1.51043171e-02 -1.42424747e-01 -2.04834137e-02  
-2.32984766e-01 -1.09302334e-01 -3.28740299e-01 3.67572874e-01  
-1.99307561e-01 1.65509537e-01 1.25560343e-01 -6.38279095e-02  
-1.22962192e-01 9.49298292e-02 3.73210877e-01 -2.39800498e-01  
5.42685166e-02 1.08299449e-01 3.33895326e-01 8.19787607e-02  
2.62362957e-01 -1.83190569e-01 1.30471036e-01 1.18146867e-01  
1.61225915e-01 1.66877747e-01 -1.03760809e-01 8.12586397e-02  
1.25506490e-01 1.19323723e-01 -2.18344137e-01 -2.92460114e-01  
1.64104015e-01 -1.48955986e-01 2.39613965e-01 1.44782826e-01  
-4.28303927e-02 -2.76273429e-01 -2.90176570e-01 -1.77377030e-01  
3.44366997e-01 1.27555370e-01 -2.59509027e-01 9.92456917e-03  
9.45689529e-02 1.08129606e-01 1.15566038e-01 1.16838388e-01  
1.65117994e-01 -1.05576493e-01 -8.67101178e-03 8.83758664e-02]

- Afşar, M., & Afşar, A. (2010). Finansal Ekonomi. *Ankara: DETAY YAYINCILIK*.
- Akira, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep Learning for Stock Prediction Using Numerical and Textual Information. *IEEE ICIS 2016 (s. 1-6)*. Okayama, Japan: 10.1109/ICIS.2016.7550882.
- Bick, B., Kraft, H., & Munk, C. (2013). Solving Constrained Consumption–Investment Problems by Simulation of Artificial Market Strategies. *Management Science vol. 59*, 485–503.
- Bildirici, M. and Ersin, Ö.Ö. (2009) Improving Forecasts of GARCH Family Models with the Artificial Neural Networks: An Application to the Daily Returns in Istanbul Stock Exchange. *Expert Systems with Applications, 36*, 7355–7362. <http://dx.doi.org/10.1016/j.eswa.2008.09.051>
- Boehmer, E., Fong, K., & Wu, J. (2012). Algorithmic Trading and Changes in firms' Equity Capital. *FIRN Research Paper: Algorithmic Trading and Changes in Firms' Equity Capital (November 12, 2012)*. FIRN Research Paper. Available at SSRN: <https://ssrn.com/abstract=2050856> or <http://dx.doi.org/10.2>.
- Bousquet, O., Boucheron, S., & Lugosi, G. (2004). Introduction to Statistical Learning Theory. Lecture Notes in Computer Science, *Advanced Lectures on Machine Learning(3176)*, 169–207.
- Boyacıoğlu, M.A. ve Avcı, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, *Expert Systems with Applications, 37(12)*, 7908–7912.
- Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *2015 IEEE International Conference on Big Data (Big Data)(s. 2823-2824)*. Santa Clara, CA, USA: IEEE.
- Cho, K., Dzmitry, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Encoder Decoder, Learning Phrase Representations using RNN Encoder Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (s. 1724-1734)*. Doha, Qatar: Association for Computational Linguistics.
- Chollet, F. (2015). Keras: The Python Deep Learning library. Keras: <https://keras.io> adresinden alındı
- Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications 83*, 187–205.
- Ciaburro, G., & Venkateswaran, B. (2017). Neural Networks with R. *Packt Publishing: Birmingham, UK*.

- Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). IJCAI'15 Proceedings of the 24<sup>th</sup> International Conference on Artificial Intelligence. *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence* (s. 2327-2333). Buenos Aires, Argentina:AAAI Press.
- Egeli, B., Özturan, M., & Badur, B. (2003) Stock Market Prediction Using Artificial Neural Networks. *Proceeding of the 3rd Hawaii International Conference on Business*. Hawaii, USA.
- Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science* 14, 179-211.
- Elmas, Ç. (2016). Yapay Zekâ Uygulamaları 3. Baskı. Seçkin Yayıncılık.
- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* Vol. 25, 383-417.
- Farley, B. G., & Clark, W. A. (1954). Simulation of Self-Organizing Systems by Digital Computer. *IEEE Transactions of the IRE Professional Group on Information Theory*, 4(4), 76-84.
- Fausett, L. (1994). Fundamentals of Neural Networks: Architectures, Algorithms and Applications. *Pearson Prentice Hall*.
- Fischer, T., & Krauss, C. (2017). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*,654-669.
- G. Zhang, B. E. (1998). Forecasting with Artificial Neural Networks: *The State of the Art. International Journal of Forecasting*, Vol. 14, 35-62.
- Gers, F. (2001). Long Short-Term Memory in Recurrent Neural Networks. *Doktora Tezi*.
- Gers, F. A., Schmidhuber, J., & Cumming, F. A. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 2451-2471.
- Ghosh, A. (2012). Comparative study of Financial Time Series Prediction by Artificial Neural Network with Gradient Descent Learning. *International Journal of Scientetific & Engineering Research*.
- Hassan, R., Natch, B., & Kirley, M.(2007). A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications*, 171-180.
- Haykin, S. (2009). Neural Networks and Learning Machines Third Edition. *New York: Pearson Prentice Hall*.
- Heaton, J. B., Polson, N. G., & White, J. H. (2016). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 3-11.
- Hebb, D. O. (1949). The Organization of Behavior. New York: OHN WILEY and SONS, Inc.

- Hiransha, M., Gopalakrishnan, E., Menon, V. K., & Soman, K. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science* 132, 1351-1362.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 1735 -1780.
- Hong, H., & Stein, J. C. (1999). A Unified Theory of Underreaction, Momentum Trading, and Overreaction in Asset Markets. *The Journal of Finance* Vol. No. 6, 2143-2184.
- Huynh, Huy D., Minh, Dang L.; A New Model for Stock Price Movements Prediction Using Deep Neural Network. *SoICT'17: Eight International Symposium on Information and Communication Technology* (s. 57-62). New York, NY, USA.
- Kara, Y., Boyacıoğlu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of Istanbul Stock Exchange. *Expert Systems with Applications* 38, 5311-5319.
- Karaboğa, H. A., & Demir, İ. (2013). Dalgacık Dönüşümü Kullanılarak Hisse Senedi Fiyat Tahmini Üzerine Bir Uygulama. *2nd International Eurasian Conference on Mathematical Sciences and Applications*. Saraybosna.
- Karlı, M. (2004). Sermaye Piyasası, Borsa, Menkul Kıymetler. İstanbul: Alfa Yayınları.
- Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing* 55, 307-319.
- Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.
- Lavrenko, V., Schmill, M., Lawrie, D., & Ogilvie, P. (2000). *Language Models for Financial News Recommendation*. 10.1145/354756.354845.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* (521), 436-444.
- Li, J., Bu, H. ve Wu, J. (2017). Sentiment-aware stock market prediction: A deep learning method, *Service Systems and Service Management (ICSSSM), 2017 International Conference on, IEEE*, s.1-6.
- McCulloch, W. S., & Pitts, W. H. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* (5), 115-133.
- McKinney, W. (2013). Python for Data Analysis. CA, USA: O'Reilly Media.
- Millman, K. J., & Aivazis, M. (2011). Scientific Python. *Volume 11 of Computing in Science & Engineering*, 9-12.

- Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock Market Index Prediction Using Artificial Neural Network. *Journal of Economics, Finance and Administrative Science* 21, 89-93
- Mohammed, M., Khan, M. B., & Bashier, E. (2016). Machine Learning: Algorithms and Applications. Boca Raton, FL,USA: CRC Press.
- Niaki, S., T., & Hoseinzade, S. (2013). Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, 1-9.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 259-268.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825-2830.
- Perşembe, A. (2001). Teknik analiz mi dedin? Hadi canım sen de! İstanbul: Scala Yayıncılık.
- Rather, A. M., Agarwal, A., & Sastry, V. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications* 42, 3234-3241.
- Roondiwala, M., Patel, H., & Varna, S. (2017). Predicting Stock Prices Using LSTM. *International Journal of Science and Research (IJSR)*, 1754-1756.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. New York: Cambridge University Press.
- Shah, Vatsal H., Machine Learning Techniques for Stock Prediction. *Foundations of Machine Learning*, 1-19.
- Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions. *Procedia Computer Science*, 895-903.
- Tayyar N., Tekin. S., (2013), "İMKB-100 Endeksinin Destek Vektör Makineleri İle Günlük, Haftalık Ve Aylık Veriler Kullanarak Tahmin Edilmesi", *Abant İzzet Baysal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 26(26), ss. 189-218.
- Turing Finance, 10 Misconceptions about Neural Networks, Erişim Tarihi:15 Haziran 2019, <http://www.turingfinance.com/misconceptions-about-neural-networks/>
- Qiu, M., Song, Y., & Akagi, F. (2016). Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos, Solitons and Fractals* 85, 1-7

## Tezden Üretilmiş Yayınlar

---

İletişim Bilgisi: [serdar.turan@live.com](mailto:serdar.turan@live.com)

### Konferans Bildirileri

1. Demir, İ., Turan, S. (2019) “Uzun Kısa Süreli Hafıza ve Geçitli Yinelenen Birim ile BIST100 Endeks Deęeri Tahmini Üzerine Bir Uygulama” *V. International Conference on Applied Economics and Finance & EXTENDED WITH SOCIAL SCIENCES (ICOAEF'19)*, Girne, KKTC