



**KTO KARATAY  
ÜNİVERSİTESİ**

**T.C.  
KTO Karatay Üniversitesi  
Fen Bilimleri Enstitüsü**

**ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
TEZLİ YÜKSEK LİSANS PROGRAMI**

**ZAMAN SERİLERİ MODELLERİ İÇİN YENİ BİR PERFORMANS  
DEĞERLENDİRME YÖNTEMİ**

**Onur ORAL**

**KONYA**

**Kasım 2019**

ZAMAN SERİLERİ MODELLERİ İÇİN YENİ BİR PERFORMANS  
DEĞERLENDİRME YÖNTEMİ

Onur ORAL

KTO Karatay Üniversitesi Fen Bilimleri Enstitüsü


Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalı  
Yüksek Lisans Programı

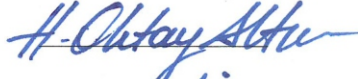
Yüksek Lisans Tezi

*Kasım, 2019*

## KABUL VE ONAY

Öğrenci Onur ORAL tarafından hazırlanan “Zaman Serileri Modelleri İçin Yeni Bir Performans Değerlendirme Yöntemi” başlıklı bu çalışma, 19 Kasım 2019 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Başkanı: Dr. Öğr. Üyesi Ali O. ÇIBIKDİKEN   
Necmettin Erbakan Üniversitesi

Tez Danışmanı: Dr. Öğr. Üyesi H. Oktay ALTUN   
KTO Karatay Üniversitesi

Jüri Üyesi: Dr. Öğr. Üyesi Şekip Engin MENDİ   
KTO Karatay Üniversitesi

Jüri tarafından kabul edilen bu çalışmanın Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.



**Prof. Dr. Hüseyin Bekir YILDIZ**  
Enstitü Müdürü

## BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini KTO Karatay Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan “Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge” kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve KTO Karatay Üniversitesi Açık Erişim Sisteminde erişime açılır.

- Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.<sup>1</sup>
- Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren ... ay ertelenmiştir.<sup>2</sup>
- Tezimle ilgili gizlilik kararı verilmiştir.<sup>34</sup>

19 Kasım 2019



Onur ORAL

<sup>1</sup> MADDE 6(1) Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

<sup>2</sup> MADDE 6(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ay aşmamak üzere tezin erişime açılması engellenebilir.

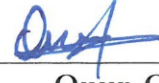
<sup>3</sup> MADDE 7(1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

<sup>4</sup> MADDE 7(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

## ETİK BEYAN

KTO Karatay Üniversitesi Fen Bilimleri Enstitüsü Tez Hazırlama ve Yazım Kurallarına uygun olarak Dr. Öğr. Üyesi H. Oktay ALTUN danışmanlığında tarafımdan üretilen bu tez çalışmasında; sunduğum tüm veri, enformasyon, bilgi ve belgeleri bilimsel etik kuralları çerçevesinde elde ettiğimi, tüm değerlendirme, analiz, bulgu ve sonuçları bilimsel usullere uygun olarak sunduğumu, tez çalışmasında yararlandığım kaynakların tümüne bilimsel normlara uygun biçimde atıfta bulunarak kaynak gösterdiğimi, tezimin kaynak gösterilen durumlar dışında özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarımı kabullendiğimi beyan ederim.

19 Kasım 2019



---

Onur ORAL

## TEŐEKKÜR

Tez alıőmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan deęerli danıőman hocam sayın Dr. Öğr. Üyesi Hüseyin Oktay ALTUN'a, ilgisini ve önerilerini göstermekten kaçınmayan deęerli hocam sayın Dr. Öğr. Üyesi Semih YUMUŐAK'a sonsuz teőekkür ve saygılarımı sunarım. alıőmalarım boyunca maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan anneme de sonsuz teőekkürler ederim.

Onur ORAL  
Kasım-2019

# ÖZET

## ZAMAN SERİLERİ MODELLERİ İÇİN YENİ BİR PERFORMANS DEĞERLENDİRME YÖNTEMİ

Onur ORAL

KTO Karatay Üniversitesi,

Fen Bilimleri Enstitüsü,

Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi H. Oktay ALTUN

Kasım 2019

Bu tezde zaman serilerini tahmin etmeye yönelik geliştirilen modellerin başarısını ölçmede geleneksel metriklerin yetersizliğini tartışarak, yeni bir zaman serisi tahmini hata hesaplama tekniği ve model değerlendirme yöntemi önermekteyiz. Konvansiyonel hata hesaplama metriklerinin zaman serileri modellerinin değerlendirilmesinde yetersiz kalmasının en önemli nedenlerinden biri; tahmin için kullanılan modellerin, bir önceki zamanda gerçekleşen değeri bir sonraki dönemin tahmin değeri olarak aynen veya çok yakın vermesi durumunda MSE, RMSE ve AE gibi yaygın olarak kullanılan metriklerin, bu şekildeki bir yaklaşımın geleceği öngörme yetisinin çok sınırlı olmasına rağmen, güzel sonuçlar veriyor görünmesinde yatmaktadır. Biz, zaman serilerinin performansının klasik yaklaşımlarla elde edilen performans metriklerinin yanında, bu bahsettiğimiz naif ve etkisiz yaklaşıma uzak olması ölçüsünde değerlendirme gereğinden yola çıkarak ikili bir metrik ve model kıyaslama tekniği öneriyoruz. Bu çalışmada, hisse senetleri fiyatlarının tahmin edilmesinde LSTM, GRU, dense ve simpleRNN gibi standart katmanlarla oluşturulmuş derin öğrenme mimarilerinin yanısıra, kendi oluşturduğumuz melez mimarilerin tahmin güçlerini de hesapladık. Bu modeller içinde, kendi önerdiğimiz kıyaslama tekniğini kullanarak en iyi modelin hangisi olduğunu belirledik.

**Anahtar kelimeler:** Zaman serisi modelleri için performans kıyası, hata ölçümü metrikleri, makine öğrenmesi, hisse senedi tahmini

# ABSTRACT

## A NEW PERFORMANCE EVALUATION METHOD FOR TIME SERIES MODELS

Onur ORAL

KTO Karatay University,  
The Graduate School of Natural and Applied Sciences,  
Master of Science Thesis in Electrical and Computer Engineering

Advisor: Asst. Prof. H. Oktay ALTUN

November 2019

In this thesis, we propose a new time series prediction error calculation technique and model assessment method. We discuss inadequacy of traditional metrics in evaluating success of predictive time series models. One naive but ineffective way of predicting the future is giving out today's realized values as tomorrow's prediction. However, conventional error metrics such as RMSE, MSE and AE etc. seem to give very good results when we employ this kind of a naive approach. Many new models based on machine learning techniques end up mimicking this incompetent behaviour, but earning high marks from conventional metrics. In addition to the performance metrics obtained by classical approaches for the performance of time series, we propose a new metric and model comparison technique, based on the need for an approach which is far from the naive and ineffective approach we have just mentioned. In this study, we calculated the predictive powers of LSTM, GRU, dense, simpleRNN and the hybrid layered architectures that we have created in addition to the standard deep learning architectures such as LSTM, GRU, dense and simpleRNN for estimating stock prices. Among these models, we determined the best model according to our suggested evaluation technique.

**Keywords:** Performance benchmark proposal for time series models, error measurement metrics, machine learning, stock market prediction



# İçindekiler

Kabul ve Onay	iii
Bildirim	iv
Etik Beyan	v
Teşekkür	vi
Özet	vii
Abstract	viii
Tablo Listesi	xi
Şekil Listesi	xii
Simgeler Dizini	xv
Kısaltmalar Dizini	xvi
<b>1 Giriş</b>	<b>1</b>
<b>2 İlgili Çalışmalar</b>	<b>4</b>
<b>3 Derin Öğrenme</b>	<b>7</b>
3.1 LSTM Ağ Yapısı	9
3.2 GRU Ağ Yapısı	11
3.3 Dense Ağ Yapısı	12
3.4 Derin Öğrenmenin Tarihçesi	12
3.5 Zaman-Serisi Tahmini için Derin Öğrenme	14

<b>4</b>	<b>Metodoloji</b>	<b>15</b>
4.1	Veri Temini	15
4.2	Veri Setinin Hazırlanması	16
4.2.1	Yahoo Finance'den Verilerin Çekilmesi	16
4.2.2	Verilerin Ön-İşlemesi	16
4.3	Derin Sinir Ağının Eğitilmesi	17
4.3.1	Melez Mimarinin Yapısı	17
4.3.2	Parametre Optimizasyonu	18
4.4	Derin Sinir Ağının Test Edilmesi	19
4.5	Geliştirilen Yeni Metrik: TRTP-RMSE	19
4.5.1	Geleneksel Performans Metrikleri	21
4.6	TRTP-RMSE ve Melez Model Sonuçları	23
<b>5</b>	<b>Sonuç ve Değerlendirme</b>	<b>26</b>
<b>6</b>	<b>Ekler</b>	<b>28</b>
	<b>Kaynaklar</b>	<b>53</b>
	<b>Özgeçmiş</b>	<b>56</b>

# Tablo Listesi

3.1	Aktivasyon Fonksiyonları	9
4.1	İş Bankasına Ait Örnek Veri	15
4.2	Geleneksel Performans Metrikleri Denklemleri	22
4.3	Derin öğrenme ile finansal verilerin tahmini için kullanılan farklı modellerin başarı tablosu	24

# Şekil Listesi

3.1	Derin öğrenme katmanları	8
3.2	LSTM yapısı	10
3.3	GRU yapay sinir ağ yapısı	12
4.1	Pareto Optimum Grafiği	20
4.2	RNN Zaman Serisi Tahmininde Zaman Kayması	22
4.3	Modellerin TRTP-RMSE ile kıyaslanması	25
4.4	Model Performans Karşılaştırması Grafiğindeki GRU-LSTM Melez Modelinden Alınan Sonuçlar	25
6.1	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	28
6.2	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	29
6.3	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	29
6.4	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	30
6.5	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	30
6.6	Tablo 4.3’de Bulunan Sonuçların TRTP-RMSE Grafiği	31
6.7	Dense 20 çevrim sayısı ve 5 pencere boyutu	31
6.8	Dense 40 çevrim sayısı ve 5 pencere boyutu	32
6.9	Dense 60 çevrim sayısı ve 5 pencere boyutu	32
6.10	GRU+LSTM 20 çevrim sayısı ve 5 pencere boyutu	32
6.11	GRU+LSTM 40 çevrim sayısı ve 5 pencere boyutu	33
6.12	GRU+LSTM 60 çevrim sayısı ve 5 pencere boyutu	33
6.13	GRU 20 çevrim sayısı ve 5 pencere boyutu	33
6.14	GRU 40 çevrim sayısı ve 5 pencere boyutu	34

6.15 GRU 60 çevrim sayısı ve 5 pencere boyutu	34
6.16 LSTM 20 çevrim sayısı ve 5 pencere boyutu	34
6.17 LSTM 40 çevrim sayısı ve 5 pencere boyutu	35
6.18 LSTM 60 çevrim sayısı ve 5 pencere boyutu	35
6.19 SimpleRNN 20 çevrim sayısı ve 5 pencere boyutu	35
6.20 SimpleRNN 40 çevrim sayısı ve 5 pencere boyutu	36
6.21 SimpleRNN 60 çevrim sayısı ve 5 pencere boyutu	36
6.22 Dense 20 çevrim sayısı ve 10 pencere boyutu	36
6.23 Dense 40 çevrim sayısı ve 10 pencere boyutu	37
6.24 Dense 60 çevrim sayısı ve 10 pencere boyutu	37
6.25 GRU+LSTM 20 çevrim sayısı ve 10 pencere boyutu	37
6.26 GRU+LSTM 40 çevrim sayısı ve 10 pencere boyutu	38
6.27 GRU+LSTM 60 çevrim sayısı ve 10 pencere boyutu	38
6.28 GRU 20 çevrim sayısı ve 10 pencere boyutu	38
6.29 GRU 40 çevrim sayısı ve 10 pencere boyutu	39
6.30 GRU 60 çevrim sayısı ve 10 pencere boyutu	39
6.31 LSTM 20 çevrim sayısı ve 10 pencere boyutu	39
6.32 LSTM 40 çevrim sayısı ve 10 pencere boyutu	40
6.33 LSTM 60 çevrim sayısı ve 10 pencere boyutu	40
6.34 SimpleRNN 20 çevrim sayısı ve 10 pencere boyutu	40
6.35 SimpleRNN 40 çevrim sayısı ve 10 pencere boyutu	41
6.36 SimpleRNN 60 çevrim sayısı ve 10 pencere boyutu	41
6.37 Dense 20 çevrim sayısı ve 30 pencere boyutu	41
6.38 Dense 40 çevrim sayısı ve 30 pencere boyutu	42
6.39 Dense 60 çevrim sayısı ve 30 pencere boyutu	42
6.40 GRU+LSTM 20 çevrim sayısı ve 30 pencere boyutu	42
6.41 GRU+LSTM 40 çevrim sayısı ve 30 pencere boyutu	43
6.42 GRU+LSTM 60 çevrim sayısı ve 30 pencere boyutu	43
6.43 GRU 20 çevrim sayısı ve 30 pencere boyutu	43
6.44 GRU 40 çevrim sayısı ve 30 pencere boyutu	44

6.45 GRU 60 çevrim sayısı ve 30 pencere boyutu	44
6.46 LSTM 20 çevrim sayısı ve 30 pencere boyutu	44
6.47 LSTM 40 çevrim sayısı ve 30 pencere boyutu	45
6.48 LSTM 60 çevrim sayısı ve 30 pencere boyutu	45
6.49 SimpleRNN 20 çevrim sayısı ve 30 pencere boyutu	45
6.50 SimpleRNN 40 çevrim sayısı ve 30 pencere boyutu	46
6.51 SimpleRNN 60 çevrim sayısı ve 30 pencere boyutu	46
6.52 Model performans karşılaştırması grafiğindeki dense modelinden alınan sonuçlar	46
6.53 Model performans karşılaştırması grafiğindeki simpleRNN modelinden alınan sonuçlar	47
6.54 Model performans karşılaştırması grafiğindeki GRU modelinden alınan sonuçlar	47
6.55 Model performans karşılaştırması grafiğindeki LSTM modelinden alınan sonuçlar	47
6.56 Model performans karşılaştırması grafiğindeki GRU ve LSTM melez modelinden alınan sonuçlar	48

# Simgeler Dizini

Simgeler Açıklama

$C_t$  Hücre durumu

$h_t$  Gizli durum

$X_t$  Girdi



# Kısaltmalar Dizini

Kısaltmalar	Açıklama
<i>ANN</i>	Yapay sinir ağları (Artificial neural networks)
<i>API</i>	Uygulama programlama arayüzü (Application programming interface)
<i>ARIMA</i>	Otoregresif entegre hareketli ortalama (Autoregressive integrated moving average)
<i>BIST100</i>	Borsa İstanbul 100 endeksi
<i>CNN</i>	Evrışimli sinir ağları (Convolutional neural networks)
<i>GA</i>	Genetik algoritma (Genetic algorithm)
<i>GRU</i>	Geçitli tekrarlayan ünite (Gated recurrent unit)
<i>HMM</i>	Gizli markov modeli (Hidden Markov model)
<i>LSTM</i>	Uzun-kısa süreli hafıza (Long-short term memory)
<i>MACD</i>	Ortalama yakınsama ve uzaklaşma hareketi (Moving average convergence divergence)
<i>MAPE</i>	Ortalama mutlak yüzde hatası (Mean absolute percentage error)
<i>MLP</i>	Çok katmanlı algılayıcı (Multi-layer perceptron)
<i>MSE</i>	Ortalama hata karesi (Mean-squared error)
<i>ReLU</i>	Doğrultulmuş lineer birim (Rectified linear unit)
<i>RMSE</i>	Kök ortalama kare hatası (Root-mean-square error)
<i>RMSLE</i>	Kök ortalama kare logaritmik hata (Root-mean-square logarithmic error)
<i>RNN</i>	Tekrarlayan sinir ağ yapısı (Recurrent neural network)
<i>RSI</i>	Görelî güç endeksi (Relative strength index)
<i>SGD</i>	Stokastik gradyan inişi (Stochastic gradient descent)
<i>SRN</i>	Basit tekrarlayan sinir ağı (Simple recurrent network)
<i>SVM</i>	Destek vektör makineleri (Support-vector machine)



# 1 Giriş

Finans sektöründe hisse senetlerinin ve döviz kurlarının gelecekte hangi değerlerde olacağını öngörebilmek önemli bir motivasyondur. Kişi ve kuruluşlar yatırımlarını gelecekte değer kazanacağı öngörülen hisseler veya para birimlerine yaparlar. Finans piyasasında bilgi kirliliği oldukça fazladır, ve kişi veya kuruluşların yatırım düşüncelerini olumsuz yönde etkileyebilmektedir. Hisse senedi ve döviz kurlarının değişim yönünün rastgele tahmin seviyesi olan yüzde elli oranından daha yüksek bir oranda tahmin edilebilmesi bile finansal yatırımcının uzun vadede kazançlı çıkmasına sebep olacaktır. Bilgi kirliliğinden arındırılmış, geçmiş verilerin iyi analiz edilmesiyle ortaya çıkarılan finansal veri tahminleri, kişi veya kuruluşların doğru yatırımı yapmasına olanak sağlar. Bir hisse senedinin veya döviz kurunun geçmiş fiyatlandırmaları, ilgili finansal verinin gelecekte hangi noktada olacağı hakkında bilgi verir niteliktedir. Bu sebeple, finansal verilerin geçmişteki hareketlerine bakılarak gelecekteki fiyatlamaları analitik metodlarla veya makine öğrenmesi teknikleriyle tahmin edilebilmektedir. Spekülatif söylemler içermeyen, geçmiş yıllara ait finansal verilerin analiz edilmesinden elde edilen tahminler kişi ve kuruluşların doğru yatırım yapmalarını sağlar ve piyasa mekanizmalarının daha da verimli şekilde çalışmasına sebep olur.

Menkul kıymetler borsaları dünyadaki bütün ülkelerin ekonomileri için önemli bir role sahiptir. Ülke bünyesinde bulunan tarım, sanayi ve teknoloji gibi alanlardaki şirketlerin ve bankaların durumlarının bir göstergesi de bu şirketlere ve bankalara ait hisselerin fiyatlanmasındaki davranışlardır. Bir ülkenin ekonomisinin bozulması veya şirketlerin finansal durumlarının kötüleşmesi gibi senaryolarda, o ülkedeki borsa endeksi düşüşe geçecektir. Ancak hisse senetlerinin fiyatlamasındaki tek rol bu gibi durumlar değildir. Spekülatif söylemler de hisse senetleri üzerinde gerçekçi olmayan fiyatlamalar oluşturabilir ve bu spekülasyonlardan dolayı bireysel veya kurumsal yatırımcılar yanlış yatırımlar yapabilmektedir.

Daha ciddi spekülasyonlar ise ekonomik kriz haline gelebilmektedir. 1857 yılındaki küresel kriz, spekülasyonların ne derece etkili olduğuna iyi bir örnektir [1]. 1848 yılında başlayan Kaliforniya altın hücumu, Amerika Birleşik Devletleri'nin çeşitli

yerlerinde yaşıyan yaklaşık üç yüz bin insanın Kaliforniya'ya taşınmasına sebep olmuştur. Çok sayıda insanın göç etmesiyle beraber, demiryolları endüstrisi kazançlı hale gelmiştir. Dönemin bankaları ise durumu fırsat olarak algılayıp fiziksel olarak demiryolu işletemeyecek şirketlere bile demiryolu işletmeleri için yüklü miktarda krediler vermiştir. Bu sebeple, demiryolu endüstrisine ait hisselerin borsada artacağı kanısı oluşarak spekülasyon bir şekilde fiyatlamalar yükselmiştir. 1850'li yılların başında Kaliforniya'da bulunan altınlarla beraber ülkenin para arzı oldukça genişlemiştir. Ancak 1850'li yılların ortasında bulunan altın miktarında ciddi bir düşüş yaşanmasıyla, batılı bankalara kredi verilmemeye başlanmış ve yatırımcılar kaçmaya başlamışlardır. Arkasından ise ekonomik kriz gelmiştir ve bu kriz küresel piyasaları etkilemiştir [2].

Spekülasyonların finans dünyasında derin problemlere sebep olmasından dolayı, kuruluşların finansal açıdan ileriye daha iyi görebilmelerini ve yatırımlarını daha emin adımlarla yapmalarını sağlamak, yatırımcı risklerinin mümkün olduğunca en aza indirilmesini sağlamak ve finans sektöründeki spekülasyonların önüne geçerek, spekülasyon söylemlerinin güvenilirliğini azaltıp yanlış yatırımların önüne geçmek amacıyla geçmiş yıllara dayalı finansal verilerin makine öğrenmesi ile analiz edilip, gelecekteki fiyatlamaların makine öğrenmesi altında yer alan derin öğrenme ile tahmin edilmesi bu çalışmada amaçlanmıştır.

Mayıs 2011 tarihinde yayınlanmış bir çalışmada, Borsa İstanbul 100 endeksinin (BIST100) fiyatlamaya yönü (artış/azalış) sınıflandırma metodları kullanılarak yapay sinir ağları (ANN) ve destek vektör makineleri (SVM) ile tahmin edilmiştir [3]. Elde edilen sonuçlara göre ANN modeli ile başarı oranı %75.74, SVM ile %71.52 tahmin edilebilmiştir. Bu sebeple bu çalışmada RNN modelleri kullanılmıştır. ARIMA ve LSTM modellerini karşılaştıran [4] ve [5] çalışmalarında kullanılan derin öğrenme modelinde bütün katmanlarının LSTM olmasının aksine, bu çalışmada GRU ve LSTM modelleri ikişer katman olmak üzere kullanılmıştır. Bu iki modelin tek başına kullanıldığında alınan sonuçlara göre RMSE metriğine göre ortalama %20 daha iyi sonuçlar elde edilmiştir. Derin öğrenme modelleri oluşturulmadan önce modelin giriş ve çıkış verileri şekillendirilmelidir. Bu çalışmada oluşturulan modelde giriş verileri ilgili hisse senedi fiyatlarının geçmiş dokuz yıla ait günlük kapanış fiyatlarıdır. Çıkış ise bir sonraki kapanış değerini barındırmaktadır. Giriş ve çıkış verilerinin seçilmesinden sonra modelin yapısı, modeldeki katman sayısı, gizli katmanların ağırlıkları ve öğrenme metodları seçilmelidir.

Çalışmalarımız sırasında, derin öğrenme modellerinin performans kıyaslamalarının yapılması esnasında, geleneksel hata ölçüm metriklerinin zaman serisi modellerinin performansını değerlendirmede tek başlarına yeterli olmadığı görülmüştür. Bazı derin öğrenme modellerinin, geleceğe dair fiyat tahminleri yapılırken bir önceki güne ait piyasa verisini bir sonraki günün tahmini olarak kaydırma eğiliminde oldukları

gözlemlenmiştir. Bu naif ve etkisiz yaklaşımın sonucu olarak elde edilen tahminlerin RMSE gibi geleneksel hata ölçüm metrikleriyle değerlendirilmesi, bu şekilde çalışmaya meyilli modellerin çok iyi başarımlar gösterdiğine dair yanıltıcı performans sonuçları vermektedir. Mamafih bu eğilimde olan derin öğrenme modelleri sadece zamanda kaydırma yapmaktadırlar. Derin öğrenme modellerinin performansının doğru kıyaslanabilmesi için yeni bir metrik önerisinde bulunmaktayız. Kök ortalama kare hatasının düşük olduğu her durumda modellerin iyi çalıştığı düşünülmemelidir. Çünkü kök ortalama kare hatasının düşük olduğu koşullarda, derin öğrenme modelleri ilgili hisseye ait önceki günün piyasa değerini bir sonraki günün tahmini olarak aktarma eğilimleri bulunabilmektedir ve sadece kök ortalama kare hatasına dayanan başarı ölçümleri gerçeği yansıtmayabilmektedir. Bu yüzden kök ortalama kare hatası metriği yetersiz görülmüş olup, çalışma kapsamında yeni bir metrik geliştirilmiştir. Nitekim, çalışmada yeni geliştirilen metrik sayesinde tahmin yapan derin öğrenme modelinin önceki günün gerçek piyasa verisini ertesi günün tahmini olarak değerlendirme durumu ve bu metriğe bağlı olarak derin öğrenme modellerin başarı durumları incelenmiş olup geleneksel hata ölçüm metriklerinin tek başlarına kullanımlarında yetersiz oldukları ispatlanmıştır.

Bu çalışma; literatür taraması, metodoloji ve sonuç bölümlerini içermektedir. Bölüm 2’de bu tez çalışmasında çalışılmış olan probleme ait geçmişte nelelerin yapıldığına dair örnek çalışmalar anlatılmış, Bölüm 3’te derin öğrenme hakkında bilgi verilmiş, Bölüm 4’te bu çalışmadaki modellerin oluşturulması, derin öğrenmede kullanılan verilerin temini ve hazırlanışı, performans ölçümü için yeni bir metriğin geliştirilmesi ve alınan sonuçların açıklanması, Bölüm 5’de ise çalışma sonucu elde edilen sonuçların değerlendirilmesi yapılmış olup, bu çalışmanın katkıları anlatılmıştır.

## 2 İlgili Çalışmalar

Borsadaki hisse senetleri gelecek fiyatlamaları tahmini borsa varolduğundan bu zamana kadar gelen bir problemdir. Hisse senetleri fiyatlarının tam olarak tahmin edilmesi insan davranışlarının da fiyatlama üzerinde etkisi olduğundan dolayı mümkün olmayacağından finans sektöründe sürekli var olan bir problemdir ve bu konuda çeşitli çalışmalar yapılmıştır ve yapılmaya devam etmektedir. Finansal verilerin derin öğrenme ile analizi konusunda yapılan geçmiş çalışmalarda genellikle lineer bir model olan otoregresif entegre hareketli ortalama ARIMA modeli kullanılmıştır. Bu modelin başarı oranı güncel LSTM ve GRU modellerine göre daha düşük olup, daha yüksek hatalara sahip sonuçlar elde edilebilmektedir [6].

Derin öğrenme ile finansal verilerin tahmini ile ilgili yapılan bir çalışmada, bilgisayarın işlem güçlerinin artmasıyla derin öğrenme gibi daha karmaşık yapıların geliştirilmesiyle ortaya çıkan yeni algoritmaların geleneksel algoritmalara karşı başarı oranı ve hata payında ne kadar gelişmiş oldukları ortaya çıkarılmıştır. Yapılan bu çalışmada ortaya çıkan sonuç ise LSTM modelinin ARIMA modeline göre ortalama %85 oranında daha düşük hata payı ile çıkış katmanından sonuç alındığı saptanmıştır [7].

2003 yılında Peter Zhang tarafından yapılan finansal verilerin derin öğrenme ile analizinin de içinde bulunduğu zaman-serisi tahmini problemi çalışmasında ARIMA modeli ile yapay sinir ağı modelinin birlikte kullanılmasıyla elde edilen bir model tasarlanmıştır. Bu modelin kullanım amacı ise lineer ARIMA modelinin ve lineer olmayan yapay sinir ağı modelinin kendilerine ait avantajlarından yararlanmak istenmesidir. Çalışmada alınan sonuçlara göre, iki modelin bir arada kullanılmasıyla daha yüksek doğrulukta sonuçlar elde edilmiştir [8].

Borsadaki hisse senetleri fiyatlarının tahmini için yapılan bir başka çalışmada çok katmanlı algılayıcı MLP ile elman tekrarlayan sinir ağı SRN ve lineer regresyon modelleri karşılaştırılmıştır. Yapılan bu çalışmada giriş verileri olarak bir hisse senetinin eski fiyatları kullanılmıştır ve gelecekteki fiyatlanma yönü ve fiyatlar tahmin edilmiştir. Çalışmanın sonucuna göre MLP modeli fiyat değişikliklerini daha

iyi tahmin edebilirken, SRN ve lineer regresyon modeli fiyat deęişim yönünü daha iyi tahmin ettiği ortaya çıkarılmıştır [9].

Apple hisse senedi ve Bitcoin fiyatlarının tahmini için 2018 yılında yapılan bu çalışmada, derin öğrenme mimarisi olarak LSTM ve ARIMA kullanılmış ve bu iki mimarinin performans karşılaştırması yapılmıştır. Apple hisse senedinin için 7 Eylül 1984 ile 28 Şubat 2018 tarihleri arasındaki fiyatlama verileri alınmıştır. Toplam 8428 adet Apple hisse senedi fiyatları içeren veri kümesindeki veriler ilgili hissenin günlük kapanış fiyatlarına aittir. Bitcoin tahmininde derin öğrenme modellerinin eğitimi için kullanılan veri kümesi 28 Nisan 2013 ve 29 Ekim 2017 tarihleri arasındaki Bitcoin fiyatlarını içermektedir. Toplam 1646 adet fiyat verisi toplanan çalışmada, derin öğrenme modelinin gelecek 30 günlük fiyatlama tahmini yapması istenmiştir. ARIMA modeli ile Apple hisse senedi fiyat tahmini yapıldığında RMSE metriğinde alınan sonuç 34.04762 olmuştur. Aynı veri kümesi kullanılarak LSTM modeli ile tahmin yapmak için giriş katmanında 64 nöronluk LSTM, ikinci katmanda 128 nöronluk LSTM, üçüncü katmanda 32 nöronluk MLP ve çıkış katmanında 1 nöronluk MLP yapısı kullanılmıştır. Aynı verilerle alınan LSTM ile alınan sonuçlarda ise RMSE metriğinde alınan deęer 9.61 olmuştur. Daha düşük RMSE, daha başarılı model demektir ve LSTM modeli Apple hisse senedi fiyatlarının tahmininde daha başarılı olmuştur. Bitcoin fiyatlarının tahmininde ise ARIMA modeli 1146.067 RMSE almıştır. Bitcoin fiyatlarının LSTM ile tahmininde giriş katmanında 64 nöronluk LSTM, ikinci ve üçüncü katmanda 128 nöronluk LSTM, dördüncü ve beşinci katmanda 64 nöronluk MLP ve çıkış katmanında 1 nöronluk MLP kullanılmış olup, 93.27 RMSE alınmıştır. Elde edilen bu sonuçlar ile birlikte LSTM modelinin geleneksel istatistiksel modellere göre daha başarılı olduğu ispatlanmıştır [10].

Zaman-serisi tahmin çalışmaları olarak 2018 yılında Bitcoin/USD fiyatlamalarının tahmin edilmesi üzerine bir çalışma yapılmıştır. Çalışmada ARIMA modeli ve LSTM modeli kullanılmıştır. Elde edilen sonuçlarda LSTM modelinin, ARIMA'ya göre daha yüksek doğruluk ve daha düşük kök ortalama kare hatası (RMSE) aldığı gözlemlenmiştir [4].

Amazon'un spot fiyatlarını tahmin etmek üzere bir çalışma da 2018 yılında yapılmıştır. Borsa fiyatları tahmini ile benzer olmasının sebebi zaman-serisi tahmini olmasıdır. Amazon spot fiyatları çok deęişken olup tahmin edilmesi zor olmasına rağmen, bu çalışmada ARIMA ile LSTM modellerinin kıyaslaması yapılmış olup, LSTM modelinin eğitimdeki hatasının %95 oranında daha düşük olduğu gözlemlenmiştir [5].

Lineer ARIMA modeli ile SVM, lineer olmayan regresyon tahmin problemlerinin çözümü için melez olarak kullanıldığı bir çalışma 2005 yılında yapılmıştır. Bu

çalıřmanın asıl amacı ise, lineer olan ARIMA modelinin lineer olmayan regresyon problemlerinde yetersiz kalmasıdır. İki modelin de kendine ait avantajlarının kullanılmıřtır, ve başarılı sonuçlar elde edilmiřtir [11]

Hisse senedi tahmini ile ilgili gizli Markov modeli (HMM) kullanılan bir çalıřma da 2005 senesinde yapılmıřtır. alıřmada havayolları ile ilgili hisse senetlerine ait gemiř fiyatlamalar, modelin eđitimi iin kullanılmıřtır. Veri setindeki komřu deđerlerin interpolasyonu ile tahminler yapılmıřtır. alıřmanın sonucunda HMM modelinin, hisse senedi tahmininde yeni bir paradigma sunduđu ve gelecek alıřmalara temel oluřturduđu ortaya ıkarılmıřtır [12].

2008 yılında yapılan, hisse senedi tahmini alıřmasında genetik algoritma (GA) ve SVM'nin melez olarak kullanıldıđı bir model üzerine alıřılmıřtır. alıřmada, genetik algoritma tm teknik gstergeler arasında en bilgilendirici girdi zellik setini semek iin kullanılmıřtır. alıřma sonucunda, sadece SVM kullanımına gre bu melez sistem daha iyi başarı gstermiřtir [13].

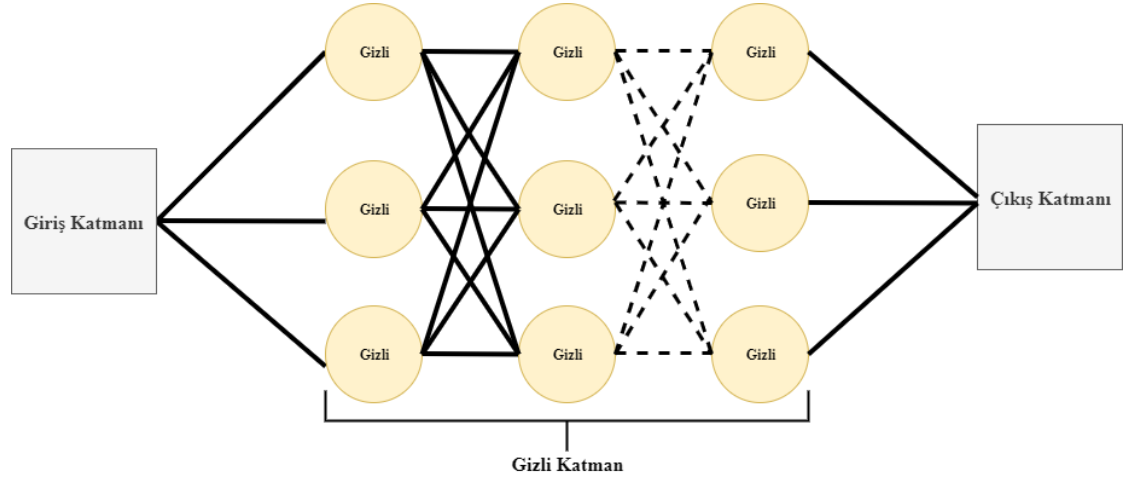
Yapay sinir ađları (ANN) ile karar ađaları (DT) kullanılarak melez bir model oluřturulan bu alıřmada, sadece ANN ve sadece DT modellerinin kullanılmasına gre daha yksek başarıml alınmıřtır. ANN borsa endeks fiyat tahmininde başarılı performans gsterirken, DT tahmin kararlarını tanımlamak iin bazı kurallar oluřturmaktadır. Bu melez yapı ile %77 oranında başarıml alınmıřtır [14].

## 3 Derin Öğrenme

Derin öğrenme günümüzde çok farklı alanlarda kullanılan bir makine öğrenmesi varyasyonudur. Görüntü sınıflandırma, finansal veri tahmini, ses tanıma, hava durumu tahminleri ve sürücüsüz araçların gelişimi gibi hayatı kolaylaştıran birçok alanda derin öğrenme kullanılabilir. Derin öğrenme mimarilerindeki katmanlarda bulunan yapay nöral ağlar, beyindeki nöronlardan ve bu nöronların birbirleriyle olan bağlantılarından ilham alınarak ortaya çıkarılmış ağlardır. İnsan beyindeki nöronlar belirli fiziksel mesafedeki diğer nöronların hepsiyle iletişim halindeyken, yapay sinir ağlarında ayrı katmanlar, bağlantılar ve ileri ve geri veri yayılım yönleri bulunmaktadır. Derin öğrenme mimarileri etiketlenmiş veya etiketlenmemiş büyük boyutlu verilerle eğitilir [15]. Eğitim aşamasının verimli veya verimsiz olmasıyla derin öğrenme mimarisinin performansı değişiklik gösterir. Örneğin; bir derin öğrenme mimarisi çeşitli çimen ve ayırık otu görselleri ile başarılı bir şekilde eğitildikten sonra, başka bir çimen veya ayırık otu görselindeki nesnelere tanıması istendiğinde çimen ve ayırık otunu ayırt edebilir. Ancak derin öğrenme modellerinin türlerine ve derin öğrenme modellerinde kullanılan parametrelerin optimizasyon durumlarına göre bu ayırt etme veya tahmin etme kabiliyetindeki başarımlar değişiklik gösterir. Derin öğrenme modellerinin performanslarının değerlendirilmesi için kök ortalama kare hatası gibi hata ölçüm metrikleri kullanılmaktadır.

Derin öğrenme yapay sinir ağları giriş katmanı, gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanına giriş vektörü belirlenip yapay sinir ağı bu vektör ile beslenir. Yapay sinir ağının öğrenmesi istenilen örnekler bu katmanda modele gönderilir. Girdinin öznitelik sayısı kadar giriş nöronu bulunmalıdır. Çıkış katmanında ise giriş vektör sayısından bağımsız olarak bir veya birden fazla çıkış nöronu bulunabilir. Örneğin; beş farklı giriş vektörünün bulunduğu bir modelde tek çıkış var olabilir. Çıkış katmanı öğrenilmesi beklenen vektörlerin hesaplanıp sonucunun alındığı katmandır. Gizli katmanlar ise giriş katmanı ve çıkış katmanı arasında bulunur. Bu katmanda geri yönlü ve ileri yönlü yayılımlar yapılabilir. Gizli katmandaki nöron sayısı, modelden öğrenilmesi istenen

problemin büyüklüğüne göre değişiklik gösterir. Ancak gizli katmandaki nöron sayısı arttıkça modelin eğitim ve hesaplama süresi artar ve model karmaşıklaşmaya başlar. Bundan dolayı probleme göre optimum gizli katman sayısı ve nöron sayısı belirlenmelidir. Bu katmandaki nöronların ağırlıkları vardır. Bu ağırlıklar, girdi örneklerinin çıkış katmanındaki sonuç üzerindeki etkisini belirlemek için kullanılır. Ağırlıklar pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan nöronlar çıkış katmanındaki sonucu değiştirmeyecektir. Yapay sinir ağındaki önemli noktalardan birisi de nöronların ağırlığını probleme göre doğru hesaplayabilmektir. Yapay sinir ağlarındaki arka arkaya görünen katmanların her biri, bir önceki katmanda çıkan sonucu kendisine girdi olarak alır [16].



Şekil 3.1: Derin öğrenme katmanları

Derin öğrenme modelinin oluşturulabilmesi için girdi vektörünün belirlenmesi (eğitimde kullanılacak veriler), çıkış katmanının belirlenmesi, gizli katman sayısı ve bu katmandaki nöron sayısının belirlenmesi, optimizasyon algoritması seçimi, katmanlardaki nöron ağırlıklarının belirlenmesi, aktivasyon fonksiyonlarının ve öğrenme yönteminin seçilmesi gibi parametrelerin belirlenmesi gerekmektedir. Bu parametrelerin optimum belirlenebilmesi için çeşitli yapay sinir ağları üzerinde eğitim ve test metodları uygulanıp başarımın karşılaştırılması gerekmektedir.

Aktivasyon fonksiyonlarının problem türüne uygun olarak seçilmesi derin öğrenme modeli için büyük bir öneme sahiptir. Yanlış seçilen aktivasyon fonksiyonları modelin başarımını düşereceği gibi performansını da azaltacaktır [17]. Aktivasyon fonksiyonu, nörona gelen verinin aktivasyonunu hesaplayarak çıktı üretir. En çok tercih edilen aktivasyon fonksiyonlarından biri doğrusal bir fonksiyon olmamasından dolayı Sigmoid fonksiyonudur ve değeri sıfır ile bir arasında olan sonuçlar üretir [18].



<i>İsim</i>	<i>Fonksiyon Denklemi</i>	<i>Fonksiyon Türevi</i>
<b>Sigmoid</b>	$f(x) = \frac{1}{1+e^{-x}}$	$f'_{(x)} = f(x)(1 - f(x))$
<b>tanH</b>	$f(x) = \tanh x = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'_{(x)} = (1 - f(x)^2)$
<b>reLu</b>	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'_{(x)} = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
<b>Leaky reLu</b>	$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'_{(x)} = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
<b>ELU</b>	$f_{(\alpha,x)} = \begin{cases} \alpha(e^x - 1) & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$	$f'_{(x)} = \begin{cases} f_{(\alpha,x)} + \alpha & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$
<b>SoftPlus</b>	$f(x) = \ln(1+e^x)$	$f'_{(x)} = \frac{1}{1+e^{-x}}$

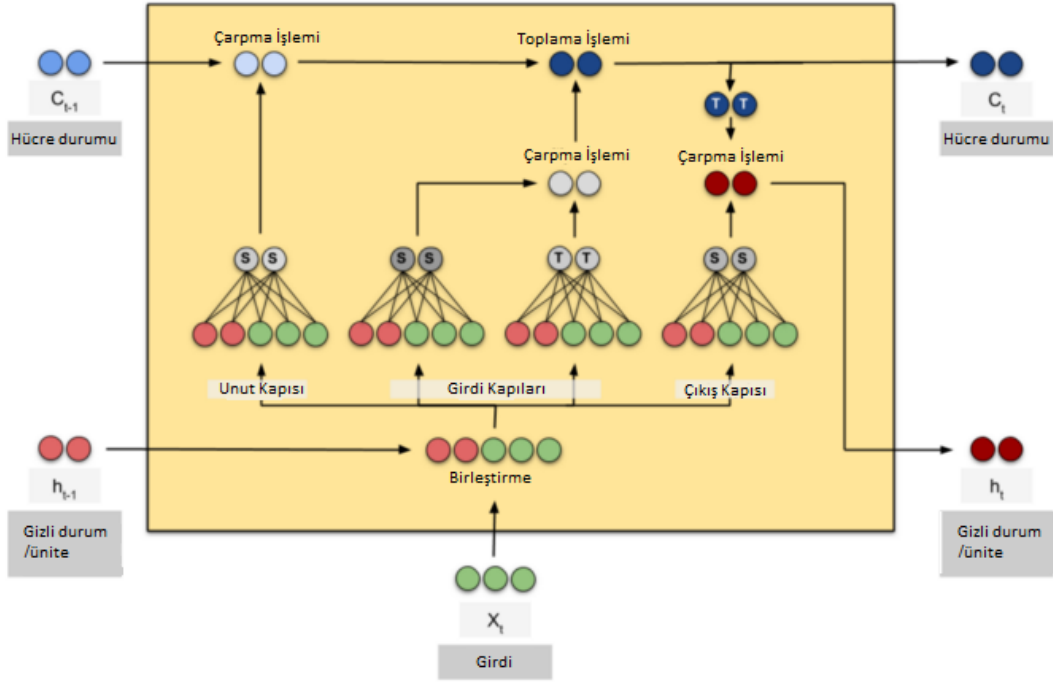
Tablo 3.1: Aktivasyon Fonksiyonları

Ancak çok katmanlı yapay sinir ağlarında hata geri yayılımı esnasında türev kaybolduğundan dolayı ilk katmanlara doğru ağırlıklar güncellenmez. Bu sebeple son yıllardaki zaman-serisi problemleri için yapılmış derin öğrenme çalışmalarında ReLU aktivasyon fonksiyonu daha çok kullanılmıştır. Ancak fonksiyonun yapısı sebebiyle ReLU fonksiyonunda negatif değerler sıfır ile sonuçlanır. Bu da negatif değer işleyen nöronların etkisiz kalmasına sebep olur. Bu sorunu da çözen aktivasyon fonksiyonları bulunmaktadır. Dolayısıyla problemin ve oluşturulan yapay sinir ağının modeline göre doğru aktivasyon fonksiyonu seçimi başarı oranının artıp hata payının düşmesi için önem arz eder.

### 3.1 LSTM Ağ Yapısı

LSTM, uzun-kısa süreli hafıza ağı olup tekrarlayan sinir ağının özel bir yapısıdır. Tekrarlayan sinir ağlarındaki uzun vadeli bağımlılık sorunu uzun-kısa süreli hafıza ağı kullanılarak giderilmiştir. Tekrarlayan sinir ağlarında, son verinin tahmini için kullanılan eğitim verisinde fazla miktarda gereksiz veri bulunursa, ilk verilerin analiz edilmesiyle son verinin doğru tahmin edilme ihtimali düşmektedir. Örneğin; “Ben Türkiye’de doğdum... Ana dilim Türkçe’dir.” cümlesinin son kelimesi olan ‘Türkçe’ kelimesi, ‘Türkiye’de doğdum.’ cümlesiyle tahmin edilecekse, üç nokta bulunan kısmın mümkün olduğunca kısa olması gerekmektedir. Eğer bu iki cümle arasında uzun boşluk bulunursa bazı tekrarlayan sinir ağları bu iki cümle arasında bağlantı kuramamaktadır. LSTM’de ise bu sorun bulunmamaktadır. Bunun sebebi ise LSTM’in uzun veri dizilerini hatırlama kabiliyetidir. Dolayısıyla geçmiş finansal veriler ile hissenin sonraki günlerinin fiyatlamasını tahmin etmek için uzun-kısa süreli hafıza sinir ağı kullanımı uygundur.

Yukarıda bahsedilen özelliklerden dolayı sinir ağı yapısı olarak iki katmanda uzun-kısa süreli hafıza modeli kullanılmıştır. Her bir girdi verisinin giriş çevriminde gizli katmanda elde edilen çıktılarının dönüşü için dönüş sırası aktif hale getirilmiştir. Bu katmanlarda uzun süreli eğitim işlemlerinde sinir ağının eğitim verilerini ezberlememesi için her bir devirde eğitim verilerinin yüzde yirmisi rastgele atılmaktadır. Ezberlemeyi engellemek adına bir kısım verilerin rastgele atılması haricinde, her çevrim sayısında eğitim verileri rastgele bir sırayla sinir ağına girmektedir. Bu şekilde sinir ağının verileri ezberlemesi engellenmiştir. Ayrıca bu yapıdan sinir ağında iki katman bulunmakta olup, her bir katmanda 150 birim vardır.



Şekil 3.2: LSTM yapısı [19]

LSTM yapısının üst kısmında düz bir çizgi halinde bulunan kısım hücre halidir. Hücre hali çizgisi üzerinden verinin değişmeden çıktı olması kolaydır. Fakat LSTM yapısında kapılar bulunmaktadır. Bu sayede verilerin üzerine bilgi ekleme veya bilgi çıkarma yeteneğine sahiptir. Şekil 3.2’de yer alan unutmaya, girdi ve çıktı kapılarında bulunan ‘S’ sembolleri sigmoid’i ifade etmektedir. Sigmoid bir değerini aldığı zaman verilerin geçmesine izin verirken, sıfır değerini aldığı durumda verilerin geçişine izin vermez. Bu şekilde hücre halindeki durumu kontrol altında tutma kabiliyetine sahiptir [20]. LSTM’in nasıl çalıştığı incelenirken öncelikle unutmaya kapısından başlanmalıdır. Bu kapı hücre durumundan nelerin atılacağına karar

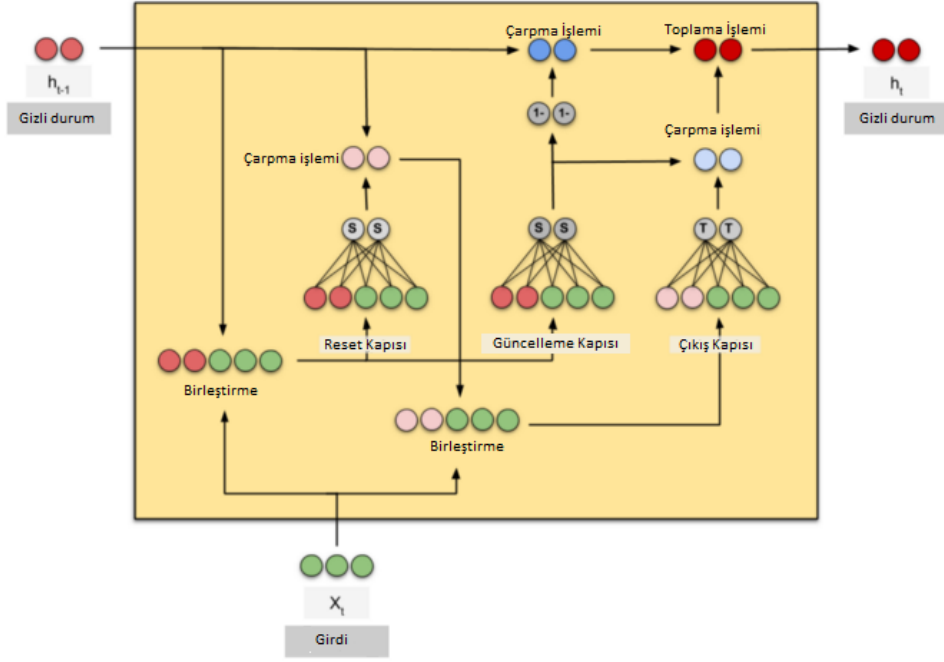
verirken Şekil 3.2'deki  $X_t$  (girdi) ve  $h_{t-1}$ 'e (önceki gizli durum çıktısı) bakar ve sigmoid fonksiyonu sıfır veya bir değerini üretir. Üretilen değere göre de tutma veya çıkarma işlemi yapılır. İkinci kapıda ise (girdi kapısı) önceki gizli katman çıktısı ile girdi sigmoid ve tanh fonksiyonundan geçer, sigmoid fonksiyonunda sıfır veya bir üretilir, tanh fonksiyonunda ise eksi bir veya bir üretilir. Ardından üretilen bu değerler çarpılır ve sigmoid fonksiyonunun çıktısı tanh fonksiyonundan gelen hangi verilerin önemli olduğuna karar verir. İkinci kapıdaki işlemler de tamamlandıktan sonra ilk kapıdan gelen çıktılar ile ikinci kapıdan üretilen çıktılar ile yeni hücre durumu ortaya çıkarılır. Bu Şekil 3.2'de  $C_t$  sembolü ile ifade edilmiştir. Üçüncü kapı ise çıkış kapısıdır. Çıkış kapısının görevi bir sonraki gizli durumun (gizli durum önceki girdiler hakkında bilgi saklar ve tahminde kullanılabilir) ne olacağına karar vermektir. Önceki gizli durum ve şu anki girdi verisi bu kapıda sigmoid fonksiyonu ile işlenir. Ardından yeni elde edilmiş hücre durumu tanh fonksiyonunda işleme alınır ve sigmoid ve tanh fonksiyonlarının çıktıları, yeni oluşacak gizli durumun hangi verileri taşıyacağı kararının verilmesi için çarpılır ve yeni gizli durum oluşturulmuş olur [21].

## 3.2 GRU Ağ Yapısı

GRU, geçitli tekrarlayan ünite, uzun-kısa süreli hafıza ağından türetilmiş olup tekrarlayan sinir ağının özel bir yapısıdır. Tıpkı uzun-kısa süreli hafıza yapısında olduğu gibi tekrarlayan sinir ağlarındaki uzun vadeli bağımlılık sorunu geçitli tekrarlayan ünite yapısında da bulunmamaktadır. Böylelikle hisselerin geçmiş finansal verilerine dayalı gelecek fiyatlamaların nasıl olacağını tahmin etmek için bu yapının kullanılması uygundur. Geçitli tekrarlayan ünite yapısının bu özelliğinden dolayı iki katmanda bu yapı kullanılmış olup, her bir katmandaki birim sayısı yetmişbeştir.

Tıpkı LSTM katmanında olduğu gibi GRU katmanında da eğitim sırasında eğitim verilerinin ezberlenmemesi için her devirde eğitim verisinin yüzde yirmisi verisetinden rastgele seçilerek çıkarılmıştır. Sinir ağının ezber yapmasını engellemek amacıyla bir kısım verilerin atılması haricinde, eğitim verilerini her bir devirde sinir ağının karışık sırayla giriş olarak alabilmesi için bu katmanda da karıştırma aktif hale getirilmiştir. Bu sayede eğitim daha verimli geçmektedir ve tahmin edilen veri grafiği çizelgesi ile gerçek veri grafiğinin çizilgesi arasındaki fark azalır, kayıplar azalmaktadır.

GRU yapısının LSTM'den en büyük farkı kapı sayısının 3'den 2'ye inmesidir. Bu kapılar sıfırlama ve güncelleme kapılarıdır. LSTM modelinde bulunan unutma ve girdi kapıları, GRU yapısında güncelleme kapısı olarak bulunmaktadır ve veri taşımak için gizli durumu kullanmaktadır. Güncelleme kapısı, LSTM'deki unutma ve girdi kapıları gibi hangi yeni verilerin eklenip, hangilerinin çıkarılacağına ka-



Şekil 3.3: GRU yapay sinir ağ yapısı [19]

rarını vermektedir. Sıfırlama kapısında ise hangi geçmiş verilerin unutulacağını kararı verilmektedir. Bu sayede, GRU modeli LSTM modeline göre bir miktar daha hızlı çalışmaktadır ve LSTM modelinin daha karmaşık yapısı GRU modelinde basitleştirilmiştir. Aralarındaki temel fark bunlardır [21].

### 3.3 Dense Ağ Yapısı

Dense yapısını diğerlerinden ayıran en önemli özelliği tamamen birbirine bağlı bir yapıda olmasıdır. Dense modelinin katmanları tam-bağlı katmanlar olarak da adlandırılmaktadır. Dense yapısının tüm giriş katmanları tüm çıkış katmanlarına bağlı olup, hepsinin bir ağırlığı bulunmaktadır. Her bir nöron, kendisinden önceki katmandaki bütün nöronlardan girdi alır.

### 3.4 Derin Öğrenmenin Tarihçesi

Derin öğrenme günümüzde popülerleşmeye başlamış olsa bile derin öğrenmenin ilk adımları 1940'lı yılların başlarına uzanmaktadır. İlk defa “beyni taklit etme” deyimini bu yıllarda ortaya çıkmıştır. Walter Pitts ve Warren S. McCulloch isimli iki bilim insanı, “Sinir sistemi içerisindeki düşüncelerin mantıksal hesabı”

isimli derin öğrenmenin günümüz temellerini kuran etkili bir makaleye imza atmışlardır [22]. Makalede, yapay sinir nöronlarının sade mantıksal işlemleri nasıl gerçekleştirebildiğine dair bilgiler anlatılmaktadır.

Edmund Berkeley 1949 yılında “Dev Beyinler veya Düşünen Makineler” isimli eserinde, makinelerin yüksek hız ile bilgi toplayabildiklerinden dolayı insan beyinine benzer bir karmaşık işleyişe sahip olmalarından, bilgi işleme kapasitelerinden, yaptıkları hesaplamalardan ve sonuçlandırabilme ve seçebilme kabiliyetlerinden ötürü bir makinenin düşünebileceği teorisini savunmuştur [23]

1950 yılında iki önemli araştırma ortaya çıkmıştır. Claude E. Shannon “Satranç oynamak için bilgisayar programlama” isimli makalesinde satranç oynayabilmek için bir algoritma geliştirme amacıyla etkili bir makale yayınlamıştır [24]. Aynı yıl Alan Turing makinelerin de insan gibi düşünebileceğini öne süren makalesinde imitasyon oyununu ortaya çıkarmıştır ve makinelerin düşünebileceğine dair teorilerini anlatmıştır [25]. 1952 yılında ise Arthur Samuel ilk kendi kendine öğrenebilen dama oyununu geliştirdi. 1959 senesinde John McCarthy cümleleri seçme, ekleme ve çıkarma yoluyla değiştirebilen bir ‘tavsiye verici’ algoritma tanımladı.

1969 yılında Yu-Chi Ho ve Arthur Bryson yapay sinir ağlarındaki geri yayılım işlevinin erken bir versiyonunu ortaya çıkarmıştır. Günümüz derin öğrenme modellerine önemli bir katkıda bulunmuşlardır çünkü günümüz yapay sinir ağ katmanlarında hala bu metot kullanılmaktadır. 1970 senesinde Stanford Üniversitesi tarafından ‘MYCIN’ isimli bir yapay zeka geliştirmiştir. Bu yapay zekanın amacı ise enfeksiyonlara sebep olan bakterilerin saptanması ve antibiyotik önerilmesidir [26]. 1980 yılında Japonya Waseda Üniversitesinde geliştirilmiş olan ‘Wabot-2’ isimli robot iletişim kurabilme, müzik notası okuyabilme gibi kabiliyetlere sahip olacak şekilde tasarlanmıştır. Ernst Dickmanns’ın önderliğinde 1986 senesinde geliştirilen ilk sürücüsüz otomobil, boş yollarda saatte 90 kilometre hızla gidebilmekteydi.

1997 senesinde Sepp Hochreiter ve Jürgen Schmidhuber bu tez çalışmasında da kullanılmış olan uzun-kısa hafıza (LSTM) yapay sinir ağı modelini ortaya çıkarmışlardır [20]. LSTM modeli günümüzde derin öğrenme problemlerinde aktif olarak kullanılmaktadır. Aynı yılda geliştirilen ‘DeepBlue’ isimli algoritma ilk defa dünya satranç şampiyonunu yenmiştir. 2000 yılında Honda tarafından geliştirilen ‘ASIMOV’ isimli insan robotu yürüyebilme yeteneğine sahip bir robottu. 2009 yılında Google’ın çalışmalarına başladığı sürücüsüz otomobili 2014 senesinde Nevada eyaletinde kendi kendine trafikte yol kateden ilk sürücüsüz otomobil oldu. 2011 yılı içerisinde, bir evrişimli sinir ağı (CNN) trafik işaretleri tanımada %99,46 doğruluk oranı ile doğruluk oranı aynı testte %99,22 olan insanları geçmeyi başarmıştır. 2014 yılında ise yüz tanıma algoritmaları geliştirilmiş olup, sanal asistan algoritmaları gün yüzüne çıkmaya başlamıştır [27]. İçinde bu-

lunduğumuz yıllarda derin öğrenme uygulamaları birçok alanda kendini göstermektedir ve her geçen zaman daha da gelişmektedir. Finans dünyasından, yüz algılamaya; ses tanımadan, duvarların arkasını görmeye kadar birçok farklı alanda derin öğrenme gelişmektedir.

### 3.5 Zaman-Serisi Tahmini için Derin Öğrenme

Zamanımızda derin yapay sinir ağları ürün fiyatları tahmini, döviz kuru tahminleri ve hisse senedi fiyat tahmini gibi zaman-serisi problemi olan çeşitli finans verilerin tahmininde kullanılmaktadır. Borsa endeksinin fiyat değişimlerinin tahmin edilemezlik durumu ise yatırımcıların önündeki büyük engellerden birisidir. Görelî güç endeksi (RSI), ortalama yakınsama ve uzaklaşma hareketi (MACD) ve emtia kanal endeksi indikatörü (CCI) gibi birçok farklı teknik kullanılmaktadır. Fakat piyasa hareketlerinin tahminini tam olarak sağlamaktan ziyade insanların tahmin yürütmesini destekleyici nitelik taşımaktadırlar. Günümüzde yatırımcılar tarafından en çok kullanılan RSI indikatörü, değeri 0 ile 100 arasında değişen değerler alır. Bu değerın aşırı alım veya satım bölgelerine yaklaşması ile yatırımcılar alım-satım işlemlerine karar verirler. Genellikle satış işlemi için değerin 75 olması, alım işlemi için 25 seviyesinde olması beklenmektedir. Ancak RSI değeri karar vermek için tek başına yeterli olmayıp, yatırımcıların RSI indikatörünü yorumlamaları gerekmektedir. Örneğın; bir hisse senedinin fiyatı çok yükselmişken, RSI değeriindeki yükseliş sınırlı ise ve işlem hacmi de azalıyorsa, fiyatların yükselişten düşüşe geçeceğini işaret edebilir. Yatırımcıların yorumuna ihtiyaç duymadan tahmin yapılması ise ancak derin öğrenme ile mümkündür. Teknolojinin gelişmesi ve derin öğrenme metotlarının yaygınlaşması ile borsa endeks tahmini yapma çalışmaları popülerleşmiştir. Bu alanda yapılan araştırmaların artmasıyla beraber yakın gelecekte derin öğrenmenin daha etkin kullanılacağı görülmektedir.

Borsa endeks tahmini amacıyla yapılacak olan çalışmalarda birçok farklı derin öğrenme modeli kullanılabilir. Yapay sinir ağları çok katmanlı yapıları sayesinde daha karmaşık problemi daha yüksek doğruluk ve daha düşük hata payı ile eğitilebilme kabiliyetinden ve daha başarılı sonuçlar elde edilebildiğinden dolayı daha uygun bir metottur. Problemin büyüklüğüne göre katmanların ayarlanabilmesi, problemin türüne göre aktivasyon fonksiyonlarının ve öğrenme yönteminin değiştirilebilmesi gibi kazançları olduğundan dolayı ideal yöntem derin öğrenmedir. Fakat finansal verilerin tahmini problemi için farklı yöntemler de kullanılabilir. Geçmiş zamanlarda yapılmış olan çalışmalarda genellikle ARIMA modelinin kullanıldığı gözlemlenmiştir. Yakın dönemde yapılan çalışmalara göre tekrarlayan sinir ağ yapısının (RNN) bir modeli olan uzun-kısa dönem hafıza (LSTM) modeli ARIMA modelinden zaman-serisi problemlerinde daha iyi başarımlar göstermiştir [5].

## 4 Metodoloji

### 4.1 Veri Temini

Finansal verilerin derin öğrenme ile tahmininde gerekli olan finansal veriler Yahoo Finance ile temin edilmiştir. Yahoo Finance kullanılmasının en önemli avantajı verileri günlük olarak güncel bir şekilde alabilmektir. Hisse senetlerine ait olan geçmiş veriler, günlük bazda gün içi en yüksek değer, gün içi en düşük değer, işlem hacmi, günün açılış değeri ve günün kapanış değeri olarak alınabilmektedir.

Tarih	En Yüksek	En Düşük	Açılış	Kapanış	Hacim
2010-02-17	4.37996	4.24309	4.27731	4.34574	19743856.0
2010-02-18	4.37996	4.24309	4.31152	4.27731	16670740.0
2010-02-19	4.41418	4.17465	4.20887	4.37996	26362060.0
2010-02-22	4.55105	4.34574	4.37996	4.37966	26582139.0
2010-02-23	4.48262	4.24309	4.31152	4.31152	38067440.0
2019-06-19	5.58000	5.37000	5.45000	5.53000	45629185.0
2019-06-20	5.70000	5.53000	5.63000	5.55000	53911723.0

Tablo 4.1: İş Bankasına Ait Örnek Veri

Yahoo Finance sayesinde hem hisse verilerinde diğer uygulama programlama arayüzlerinde (API) bulunan gecikme yaşanmamış olup hem de Borsa İstanbul hisseleri de dahil olmak üzere çeşitli ülkelerdeki hisseleri almaya imkan sağlamaktadır.

Yahoo Finance üzerinden hisselerin finansal verilerini alabilmek için pandas'ın datareader ve Yahoo Finance kütüphaneleri yüklenmiştir. Seçilen hissenin 2010 yılından itibaren olan günlük olarak açılış, kapanış, en yüksek, en düşük değerleri ile günlük işlem hacmi verileri alınmıştır. Uygulamanın çalıştırıldığı zamana ait veriler de aynı şekilde eklenmiştir.

## 4.2 Veri Setinin Hazırlanması

Derin öğrenme ile hisse senetlerinin finansal analizinin yapılabilmesi için hisse senetlerinin geçmiş verilerinin toplanması gerekir. Bu verilerin günlük bazda alınıp, günlük kapanış fiyatları ile sinir ağı eğitim ve testi yapılmaktadır. Alınan verilerin 2018 yılına kadar olan kısmı eğitim amacıyla bir veri setine aktarılmıştır.

### 4.2.1 Yahoo Finance'den Verilerin Çekilmesi

Veriler Google Colaboratory ve PyCharm üzerinden işlenmiştir. Colaboratory ile bulut sunucu üzerinden daha performanslı donanımlar ile çalışma imkanı sunduğu için Pycharm yerine genellikle Colaboratory kullanılmıştır. Colaboratory veya PyCharm kullanılarak Yahoo Finance üzerinden veri alabilmek için pandas datareader ve Yahoo Finance kütüphanelerinin yüklenmesi gerekmektedir. Bu iki kütüphane kullanılarak istenilen hisselerin geçmiş fiyatlamaları bir veri seti içerisine alınabilmektedir. Gelen veride herhangi bir boşluk olmaması için kayıp veriler dataset içerisinden silinmektedir. Alınan veriler altı farklı veri kümesi içermektedir. Tablo 4.1'de de örnek olarak bulunan bu veri kümesi sırasıyla veriye ait tarih, o tarihteki en yüksek ve en düşük değerler, açılış değeri, kapanış değeri ve işlem hacmidir. Kullanılacak derin öğrenme modeli için önemli olan veri kümeleri ise tarih ve günlük kapanış değerinin bulunduğu verilerdir. Derin öğrenme modeli ile gerçekleştirilecek olan eğitim ve test işlemleri günlük kapanış değerleri üzerinden yapılmaktadır.

### 4.2.2 Verilerin Ön-İşlemesi

Yahoo Finance ile alınan geçmiş yıllara ait finansal veriler altı farklı veri kümesi halinde gelmektedir. Derin öğrenme modelinde kullanılacak olan veriler ise son dokuz yıla ait günlük kapanış fiyatlarıdır. Alınan veriler arasından günlük kapanış değerlerini içeren veri kümesi seçilmiştir ve bir veri setine aktarılmıştır. Alınan veri kümesinde bulunan 2010 ile günümüz arasındaki veriler eğitim ve test için ayrılmıştır. Eğitim için ayrılan veriler 2010 ile 2018 yılları arasına aittir. 2018 yılından günümüze kadar olan veriler ise test için kullanılmaktadır. Uygulama eğer piyasalar açıkken gün içi bir zamanda çalıştırılıyorsa o güne ait son veri alınmıştır. Eğitim ve test için ayrılan veriler öncelikle 0-1 arasına ölçeklendirilmiş olup daha sonra tek boyutlu NumPy dizisi olarak tanımlanmıştır. Herhangi bir hata olması ve bazı verilerin boş gelmesi durumunda boş gelen verilerin kullanılmasını engellemek amacıyla veriler kontrol edilip boş olan veriler seriden çıkarılmaktadır. Derin öğrenme modelinin hatasız çalışabilmesi için NumPy dizisinde bulunan geçmişe ait finansal veriler yeniden biçimlendirilmiştir.



## 4.3 Derin Sinir Ağının Eğitilmesi

Sinir ağının eğitimi esnasında seçilmiş olan hisseye ait olan geçmiş verilerin 2010 ile 2018 yılları arasındaki piyasa verileri eğitim için 2018'den günümüze kadar olan veriler ise test verisi olarak kullanılmıştır. Bu veriler günlük bazda toplanmış olup, günlük kapanış değerleri ile bir dizi oluşturulmuştur. Oluşturulan bu dizi, yeniden şekillendirilerek yapay sinir ağının eğitilmesine hazır hale getirilmiştir. Aynı zamanda, eğitim esnasında yapay sinir ağının verilen eğitim verilerini ezberlemesi için karıştırma yöntemi uygulanmıştır. Bu sayede eğitim esnasındaki her bir çevrimde, eğitim verilerinin yapay sinir ağına giriş sırası değişmektedir ve yapay sinir ağının eğitim verisini ezberlemesi engellenmektedir. Bu sayede yapay sinir ağı daha kararlı hale getirilmiştir. GRU, LSTM, dense, simpleRNN ve melez modellerinin eğitimlerinde performans kıyaslamaları için çevrim sayısı 20, 40 ve 60 olarak seçilmiştir. Ayrıca, eğitim ve test esnasında kullanılan pencere boyutu 5, 10 ve 30 seçilerek eğitim ve testler gerçekleştirilmiş olup, performans karşılaştırmaları yapılmıştır. Pencere boyutu, modelin bir sonraki günün fiyatını tahmin etmesi için kullanacağı günlük verilerin kaç günlük bir boyutta olacağını belirtmektedir. Her bir pencere boyutu için 20, 40 ve 60 olarak belirlenmiş olan çevrim sayılarında eğitimler ve testler tamamlanmış olup, Tablo 4.3'de derin öğrenme modellerinin aldıkları performans sonuçları listelenmiştir.

Derin öğrenmede yapay sinir ağlarında kullanılacak birçok farklı mimari bulunmaktadır. Bu tez çalışmasında simpleRNN, dense, GRU, LSTM ve GRU-LSTM melez modelleri kıyaslanmıştır. Zaman-serisi tahmin problemlerinden biri olan borsadaki hisse senetlerinin gelecek fiyatlamalarının tahmininde en başarılı sonuçlar GRU ve LSTM'in melez olarak kullanılması ile elde edilmiştir. Deneimler esnasında bu iki yapıyı aynı yapay sinir ağı içerisinde farklı katmanlara yerleştirilmesiyle diğer derin öğrenme modellerine göre daha başarılı sonuçların alınabildiği gözlemlenmiştir. Melez yapıdaki bu derin öğrenme modelinin katmanlarında hem GRU mimarisi hem de LSTM mimarisi bulunmaktadır.

### 4.3.1 Melez Mimarinin Yapısı

Çalışmada kullanılan melez derin öğrenme modelinin ilk ve dördüncü katmanı GRU olup, iki ve üçüncü katmanı LSTM yapısındadır. Çıkış katmanı ise dense'dir. GRU, simpleRNN, dense ve LSTM modellerinin tek başına kullanılmasıyla oluşturulan derin öğrenme modellerinden daha başarılı sonuçlar elde edilmiştir. Melez derin öğrenme modelinde ilk ve dördüncü katman 75 birimlik GRU modeli, ikinci ve üçüncü katmanlar 150 birimlik LSTM modeli olarak belirlenmiştir. Dense yapısı ise sadece çıkış katmanında bulunduğu ve bu tez çalışmasında istenilen çıkışın tek boyutlu olmasından dolayı tek birimdir.

Bu tez çalışmasındaki derin öğrenme modellerinde optimizasyon algoritması olarak Adam algoritması seçilmiştir. Hisse senetlerinin tahmini bir zaman serileri problemi olduğundan dolayı Adam algoritması en verimli sonuçları vermektedir. Açılımı adaptif momentum olan bu algoritmanın en önemli özelliği öğrenme oranlarıyla birlikte momentum değişikliklerini de önbellekte tutabilmesidir. Bu algoritma farklı kapsamlardaki farklı derin öğrenme mimarilerinde de kullanılabilir. Ancak, optimizasyon algoritmasının problem türüne ve derin öğrenme yöntemine uygun seçilmesi, derin öğrenme modelinden alınacak performansı optimum düzeyde tutmak için önemlidir. Örneğin; stokastik gradyan inişi (SGD) algoritması ile görüntü tanıma üzerinde çalışan bir derin öğrenme uygulaması yapılırsa başarılı sonuç alınmayabilir. Adam'ın diğer avantajlarından biri de diğer algoritmalara göre daha düşük belleğe ihtiyaç duymasındadır [28].

### 4.3.2 Parametre Optimizasyonu

Derin öğrenme mimarisi ile zaman serileri problemi olan hisse senedi fiyatlarında gelecek tahmini çalışmasında sonuç başarımını etkileyen önemli faktörlerden biri de parametrelerin ne kadar iyi optimize edildiğidir. Bir derin öğrenme yapısında modelin performansını doğrudan etkileyen unsurlardan biri de yığın(batch) boyutudur. Yığın boyutunun çok düşük olması durumunda modelin her bir çevrimde geçireceği süre artacaktır, ancak daha az çevrim sayısı ile düşük kayıp değerleri elde edilebilir. Ancak yığın boyutunun azaltılması ve çevrim sayısının da düşürülmesine rağmen derin öğrenme modelinin bütün çevrimleri bitirmesi için gereken süre, yüksek yığın boyutu ve daha yüksek çevrim sayısındaki süreden daha fazla olabilir. Aynı zamanda çok düşük yığın boyutu derin öğrenme modelinin ezber yapmasına sebep olabilir. Bundan dolayı kullanılan derin öğrenme modeline ve problemin karmaşıklığına göre yığın boyutunun deneyerek optimize edilmesi gerekmektedir. Bu tez çalışmasında yığın boyutu 32 olarak belirlenmiştir.

Derin öğrenme modelinden optimum başarımların alınabilmesi için çevrim sayısının da doğru belirlenmesi gerekmektedir. Derin öğrenme modelinin karmaşıklığına da bağlı olarak, bir çevrim sayısından sonra MSE sabit kalır veya yükselmeye başlayabilir ve bu noktadan sonra o derin öğrenme modelinin hala eğitilmesi modelin başarımını olumsuz etkilemektedir. Dolayısıyla, çevrim sayısının optimize edilmesi daha iyi sonuçlar alınabilmesini sağlarken, derin öğrenme modeli daha uzun süre eğitimde tutulmadığı için eğitim süresi kısaldı ve daha hızlı sonuçlar alınabilmesine olanak sağladı. Bu tez çalışmasında çevrim sayısı derin öğrenme modellerinin performanslarının karşılaştırılması için 20, 40 ve 60 olarak belirlenmiştir.

Bu tez çalışmasındaki farklı derin öğrenme modellerinin performans kıyaslamalarında 5, 10 ve 30 günlük pencere boyutları kullanılmıştır. Eğitim ve test esnasında 5, 10 ve 30 günlük veri pencereleri analiz edilerek ertesi gün için

fiyat tahminleri gerçekleştirilmiştir. Pencere boyutunun değişmesi derin öğrenme modelinin fiyat tahmin etme davranışında ve modelin başarımında farklılıklara sebep olmaktadır. Tez çalışmasında kullanılan derin öğrenme modellerinin farklı pencere boyutları ve çevrim sayılarındaki başarımlarını karşılaştırmaları Şekil 4.3 ve Tablo 4.3’de gösterilmiştir.

## 4.4 Derin Sinir Ağının Test Edilmesi

Yahoo Finance üzerinden alınan verilerin 2018 yılının başlangıcından itibaren günümüze kadar olan kısmı sinir ağının testi için kullanılan verilerdir. Bu veri, seçilmiş olan hissenin günlük bazda kapanış değerlerini içermekte olup, eğer uygulama gün içi bir zamanda çalıştırılıyorsa, çalıştırma esnasındaki anlık veri o gün için alınır. Derin sinir ağı eğitim verileriyle eğitilmesinden sonra, 2018 yılının başından itibaren olan veriler ile test yaptırılmıştır. Eğitim ve testler sırasında, 5, 10 ve 30 gün boyutlu pencereler kullanılmıştır. Pencere boyutunda belirlenen gün sayısına göre derin öğrenme modeli bir sonraki günün tahmini yapmaktadır. Çevrim sayıları ise 20, 40 ve 60 olarak belirlenmiş olup, farklı pencere boyutlarına ve farklı çevrim sayılarına göre GRU, LSTM, simpleRNN, dense ve melez modelden elde edilen sonuçlar Tablo 4.3’de gösterilmiştir. Yapılan bu tahmin sonrasında, uygulama gerçek hisse fiyatlarının bulunduğu siyah renkli bir grafik ile, bu fiyatların tahmin edildiği kırmızı renkli bir grafik olarak verir.

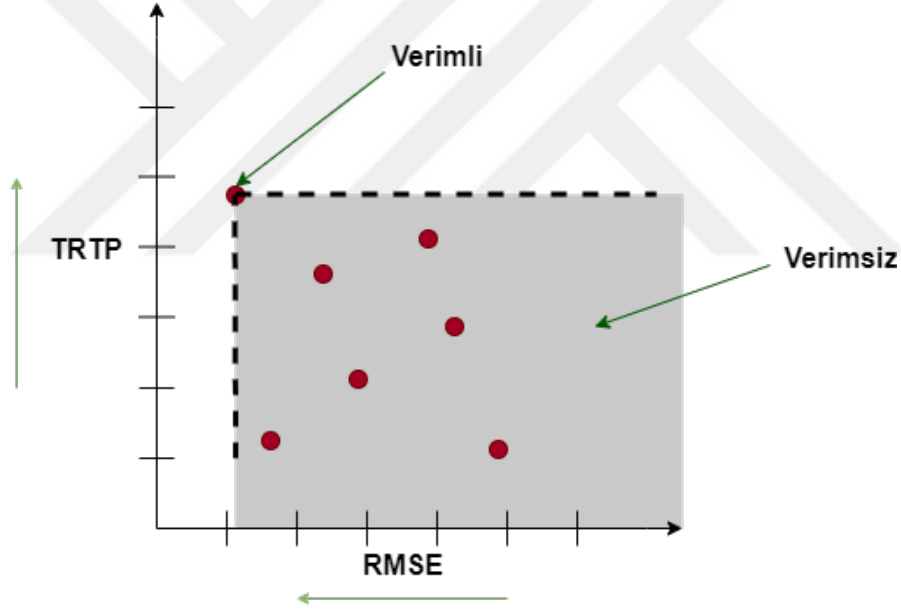
Derin öğrenme modellerinin performanslarının ölçümü için geliştirdiğimiz TRTP ve geleneksel RMSE ölçümleri kullanılmıştır. Bu iki metriğin beraber kullanılması ile en verimli ve başarılı modelin ortaya çıkarılması için Şekil 4.1’deki gibi pareto optimum bölgesi bulunmuştur. Pareto optimum durumunda, toplumdaki bireylerin hepsinin hayat koşulları eşit olarak artırıldıktan sonra, o toplumdaki bir bireyin hayat standartlarının daha fazla artırılabilmesi için aynı toplumdaki bir başka bireyin zarar görmesi gerekmektedir. Oluşan bu optimum duruma pareto optimum denir [29].

## 4.5 Geliştirilen Yeni Metrik: TRTP-RMSE

Derin öğrenme modellerinin performanslarını kıyaslamak için birçok metrik bulunmaktadır. Yaygın olarak kullanılan ortalama hata kareleri kökü (RMSE) ise yanıltıcı sonuçlar verebilmektedir. Örneğin; bu tez çalışmasında da performans kıyaslamasında kullanılan bazı modeller ortalama hata kareleri köküne göre çok iyi başarı göstermiş olmalarına rağmen, bir sonraki günün tahmin sonuçları ile bir önceki günün gerçek piyasa verisi birbirine çok yakın olabildiği gözlemlenmiştir. Bu durumda ortalama hata kareleri kökü metriği, modelin çok iyi çalıştığını göstermesine karşılık, modelin aslında bir sonraki günü tahmin etmek için sadece bir

önceki günün gerçek piyasa verisine baktığı ve bu gerçek veriye çok yakın bir değeri tahmin olarak sunmasından şüphelenilmiştir. Bundan dolayı ortalama hata kareleri kökü metriğinin yanında performans karşılaştırması için TRTP-RMSE geliştirilmiştir. TRTP, 'Bugünün Gerçekleri Yarının Tahminleri' anlamına gelmektedir.

TRTP-RMSE sayesinde derin öğrenme modeli sonraki günün tahmin değerini hesaplarken, tahmin verisinin bir önceki günün piyasa verisine ne kadar yakın olduğu ölçülmüştür. TRTP, sonraki günün tahmin verisi ile önceki günün gerçek piyasaya verisi arasındaki farkın karelerinin toplamını hesaplamaktadır. Eğer derin öğrenme modeli önceki günün gerçek değerini olduğu gibi sonraki güne tahmin olarak aktarırsa, TRTP'den alınan sonuç sıfır olacaktır. Fakat modelin tahminleri, önceki günlerin gerçek değerine endeksli olacağı için model başarılı bir grafik sunup, ortalama hata kareleri kökü (RMSE) metriği modelin başarılı olduğunu gösterecektir. Aksine, model oldukça başarısız olup sadece ' $t$ ' zamandaki piyasa verisini ' $t+1$ ' için tahmin olarak yazmaktadır.



Şekil 4.1: Pareto Optimum Grafiği

TRTP-RMSE yönteminin kullanılmasıyla derin öğrenme modellerinin performans sonuçları için pareto optimum bulunarak hangi modelin en yüksek başarıyı ve hangi modellerin düşük başarı gösterdiği bulunabilmektedir. Toplumdaki bireylerden en az biri daha kötü hayat yaşamadan, toplumdaki diğer bir bireye daha iyi bir hayat yaşatılamaması durumu pareto optimumdur [29]. Pareto optimum grafiğinde

bulunan yeşil okların yönü, TRTP-RMSE yöntemine göre daha iyi performans sergileyen modeli belirtmektedir. TRTP metriğine göre daha yüksek, RMSE'ye göre daha düşük sonuç elde edilmesiyle en performanslı çalışan model bulunmaktadır. Buna göre Şekil 4.1'deki 'verimli' olarak işaretlenen model pareto optimum noktasıdır ve grafikteki diğer modellerden daha performanslıdır. Bu tez çalışmasında aynı yaklaşım ile TRTP-RMSE sonuçları değerlendirilerek Şekil 4.1'deki gibi pareto optimum noktası bulunmuştur. Derin öğrenme modelinin, başarılı bir model olabilmesi için ortalama hata kareleri kökünden gelen sonucun daha düşük olması, TRTP'nin verdiği sonucun ise daha yüksek olması gerekmektedir. Eğer ortalama hata kareleri kökü düşükse ve TRTP ölçümünden alınan sonuç da sıfıra daha yakın veya sıfır ise, bu modelin önceki günün gerçek fiyatını sonraki günün tahmini olarak değerlendirdiğini göstermektedir ve ortalama hata kareleri kökü düşük olmasına rağmen bu model başarısız bir modeldir.

RMSE ve geliştirdiğimiz TRTP metriğinin birlikte kullanılmasıyla derin öğrenme modellerinin performans karşılaştırması için pareto optimum bölgesi oluşturulmuştur. Bu modelin uygulanması sonucu Şekil 4.1'de verimsiz olan bölge ve verimli örnek model gösterilmiştir. Grafikte örnek olarak eklenen modellerin en verimlisi, RMSE'nin en düşük olduğu ve geliştirdiğimiz TRTP ölçümünün en yüksek olduğu lokasyondur. Bu örnek model diğer örnek modellerden daha verimli ve başarılı çalışmaktadır. Derin öğrenme modellerinin başarısı sadece geleneksel metrik olan RMSE veya diğerleri ile kıyaslınsaydı, daha düşük RMSE değerine sahip ancak gerçek başarıyı düşük olan model başarılı sayılacaktı. Çıkan bu sonuç ile birlikte sadece geleneksel metriklerin, zaman serisi tahmin problemlerinde kullanılan derin öğrenme modellerinin performanslarının ölçümünde tek başına yeterli değildir.

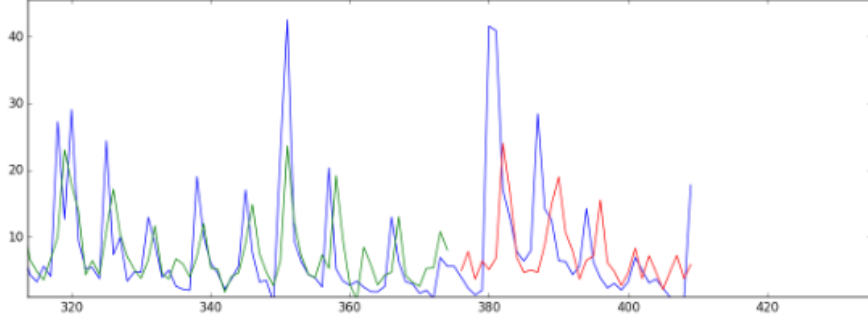
#### 4.5.1 Geleneksel Performans Metrikleri

Derin öğrenme modellerinin performans ölçümlerini yapabilmek için mimariye ölçüm metriklerinin eklenmesi gerekmektedir. Kök ortalama kare hatası (root mean-square error), ortalama hata karesi (mean-squared error), mutlak hata (absolute-error), mutlak doğruluk hatası (absolute accuracy error), ortalama mutlak yüzde hatası (mean absolute percentage error), kök ortalama kare logaritmik hata (root mean-squared logarithmic error) gibi metrikler derin öğrenme modelinin performans ölçümünde yaygın olarak kullanılmaktadır.

<i>İsim</i>	<i>Metrik Denklemi</i>
<b>MSE</b>	$MSE = \frac{1}{n} \sum_{i=1}^n (f_i - Y_i)^2$
<b>RMSE</b>	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - Y_i)^2}$
<b>Mutlak Hata</b>	$\Delta x =  x_i - x $
<b>MAPE</b>	$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left  \frac{A_t - F_t}{A_t} \right $
<b>RMSLE</b>	$RMSLE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(x_i + 1) - \log(y_i + 1))^2}$

Tablo 4.2: Geleneksel Performans Metrikleri Denklemleri

Zaman serisi tahmin problemlerinin performans ölçümlerinde bu metrikler yeterli olamamaktadır. Konuyla uğraşan araştırmacılar kullandıkları derin öğrenme modellerinin ‘ $t$ ’ zamanındaki gerçek veriyi ‘ $t+1$ ’ zamanına tahmin olarak değerlendirmesinden şikayetçi olmaktadır. Bu durum gerçek ve tahmin verilerinden oluşturulan grafiklerde, tahmin grafiğinin gerçek grafiğin sağına doğru kaymış olması ile görülebilmektedir [30]. 2018 yılına ait yapılan başka bir çalışmada yine zaman kayması sorunu görülmüş olup, zaman serisi problemlerine ait tahminlerde önemli bir problem haline gelmiştir [31]. LSTM ile yapılan diğer bir zaman serisi tahmininde benzer şekilde derin öğrenme modelinin önceki günün gerçek verisini ertesi gün için tahmin değeri olarak yazması görülmüştür [32].



Şekil 4.2: RNN Zaman Serisi Tahmininde Zaman Kayması [32]

Ancak önceki günün gerçek verisinin bir sonraki gün için tahmin olarak değerlendirilmesi durumunda RMSE sonucu yanıltıcı olarak oldukça iyi çıkmaktadır ve modelin iyi çalıştığı düşünülmektedir. Model ise yalnızca önceki günün piyasa verisini ertesi günün tahmin verisi olarak değerlendirmesi şeklinde basit bir işlem yapmaktadır ve RMSE metriğine göre modelin iyi çalıştığı gözlemlenmektedir. Derin öğrenme modelleri için performans kıyaslaması yapılırken bu durum hatalı sonuçların almasına sebep olmaktadır. Aslında düşük başarımların gösterilmesi

ren ve yaptığı iş sadece zamanda kaydırma olan modellerin RMSE sonucunun iyi çıkmasından dolayı iyi performans gösterdiği sanılmaktadır. Geleneksel hata hesaplama metriklerinin zaman serilerini tahmin eden derin öğrenme modellerini ölçmelerinde tek başlarına yetersiz olmalarının sebebi, derin öğrenme modellerinin geçmiş güne ait hisse senedi fiyatlamalarını, ertesi gün için fiyat tahmini olarak birebir aktarma yaptıklarında, RMSE gibi ölçüm metriklerinin derin öğrenme modelleri için oldukça başarılı oldukları sonucunu çıkarmasındadır ve RMSE gibi geleneksel ölçüm metrikleri tek başlarına yeterli değildir. Aslında çok iyi görünen ama gerçekte ilgili hisseye ait önceki günün piyasa verilerinin tahmin verisi olarak yazılması, RMSE gibi metriklerin derin öğrenme modelinin iyi çalışıyormuş gibi sonuç vermesine karşılık aslında modelin kötü başarımla göstermesiyle sonuçlanır.

RMSE ve MSE gibi geleneksel ölçüm metriklerinin tek başına yeterli olmadığı durumu gözlemlenmiştir. Gelenek performans metrikleri. Yapmış olduğumuz yeni ölçüm metriği, bir sonraki zamana ait tahmin verisinden, bir önceki zamana ait gerçek verinin farkının alınmasından sonra çıkan verinin karelerinin toplamı ile hesaplanmıştır ve dizideki bütün fiyat verileri üzerinde uygulanmıştır. Geliştirdiğimiz TRTP-RMSE ile beraber pareto optimum yönteminin kullanılmasıyla en iyi performansı sergileyen derin öğrenme modeli ortaya çıkarılmıştır.

## 4.6 TRTP-RMSE ve Melez Model Sonuçları

Uygulanan derin öğrenme modelinde uygulamanın başarı oranı kök ortalama kare hatası ve ortalama karesel hatanın düşük olması ve geliştirdiğimiz TRTP metriğinin sonucunun yüksek olması ile ölçülmüştür. Ortalama hata kareleri kökünün 0.5 değerinden daha büyük olması halinde modelin başarımla vasat, 0.3 değerinden daha düşük olması durumunda tahmin yeteneği iyi sayılabilmektedir [33]. Ancak, sadece ortalama hata kareleri kökü metriğine göre başarı grafiği oluşturmak hatalı sonuçlar alınmasına sebep olmaktadır.

Ortalama hata kareleri kökü sonucunun iyi olduğu durumlarda, derin öğrenme modeli önceki günün piyasa verisini ertesi günün tahmini olarak değerlendirebilmektedir ve bu durumda ortalama hata kareleri köküne göre iyi çıkan sonuçlar yanıltıcı olmaktadır. Bu durumun önüne geçebilmek için bu tez çalışmasında TRTP-RMSE geliştirilmiş olup, TRTP-RMSE ile beraber pareto optimum yönteminin kullanılmasıyla istikrarlı performans sonuçları elde edilmektedir. TRTP-RMSE ve pareto optimum ile geliştirdiğimiz yeni performans ölçüm yöntemi ile en başarılı çalışan model tam olarak tespit edilebilmektedir. Buna göre en başarılı model seçilirken TRTP sonucunun büyük, ortalama hata kareleri kökünün ise düşük olması gerekmektedir. TRTP'den elde edilen sonucun büyük istenmesinin sebebi, metriğin sonucunun sıfır gelmesi, derin öğrenme mo-

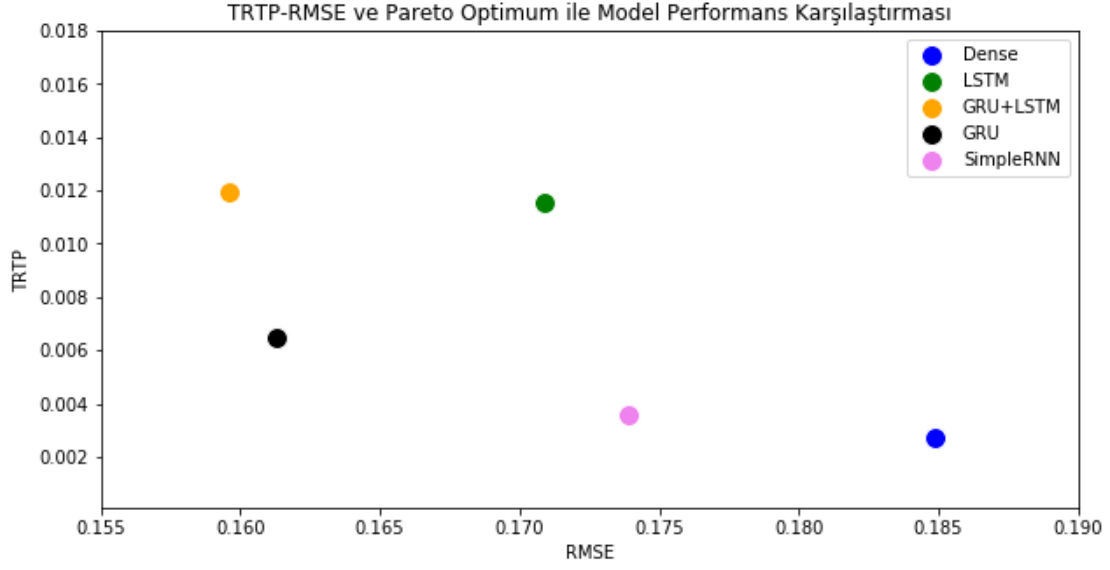
delinin önceki günün piyasa verisini sonraki günün tahmini olarak değerlendirmesi anlamına gelmektedir. Bu yüzden sadece ortalama hata kareleri kökü metriğine bakılarak başarı değerlendirilmesi yapılamamaktadır, hatalı sonuçlar alınmaktadır.

Model	Çevrim Sayısı	Pencere Boyutu	MSE	RMSE	TRTP-RMSE
Dense	20	5/10/30	0.0028/0.0018/0.0016	0.203/0.170/0.166	0.007945/0.000326/0.000985
	40	5/10/30	0.0015/0.0014/0.0014	0.179/0.180/0.159	0.000779/0.000217/0.004594
	60	5/10/30	0.0014/0.0013/0.0013	0.156/0.162/0.154	0.000182/0.000370/0.000101
GRU+LSTM	20	5/10/30	0.0019/0.0016/0.0015	0.199/0.181/0.169	0.011197/0.015627/0.011868
	40	5/10/30	0.0013/0.0012/0.0011	0.142/0.161/0.148	0.003647/0.016026/0.010253
	60	5/10/30	0.0011/0.0010/0.0009	0.142/0.143/0.139	0.005600/0.005287/0.003329
GRU	20	5/10/30	0.0018/0.0016/0.0017	0.178/0.157/0.167	0.006506/0.000212/0.000231
	40	5/10/30	0.0014/0.0012/0.0012	0.143/0.161/0.157	0.002575/0.007622/0.009514
	60	5/10/30	0.0012/0.0011/0.0011	0.167/0.146/0.139	0.003187/0.001204/0.000059
LSTM	20	5/10/30	0.0024/0.0022/0.0021	0.236/0.216/0.197	0.000620/0.002184/0.009744
	40	5/10/30	0.0014/0.0014/0.0014	0.173/0.174/0.162	0.002751/0.012605/0.000552
	60	5/10/30	0.0012/0.0012/0.0011	0.149/0.171/0.151	0.001259/0.000709/0.000479
SimpleRNN	20	5/10/30	0.0043/0.0094/0.0098	0.207/0.189/0.245	0.000330/0.000414/0.007646
	40	5/10/30	0.0029/0.0030/0.0032	0.161/0.164/0.196	0.001842/0.003356/0.003694
	60	5/10/30	0.0020/0.0023/0.0024	0.148/0.149/0.179	0.000792/0.002796/0.000265

Tablo 4.3: Derin öğrenme ile finansal verilerin tahmini için kullanılan farklı modellerin başarı tablosu

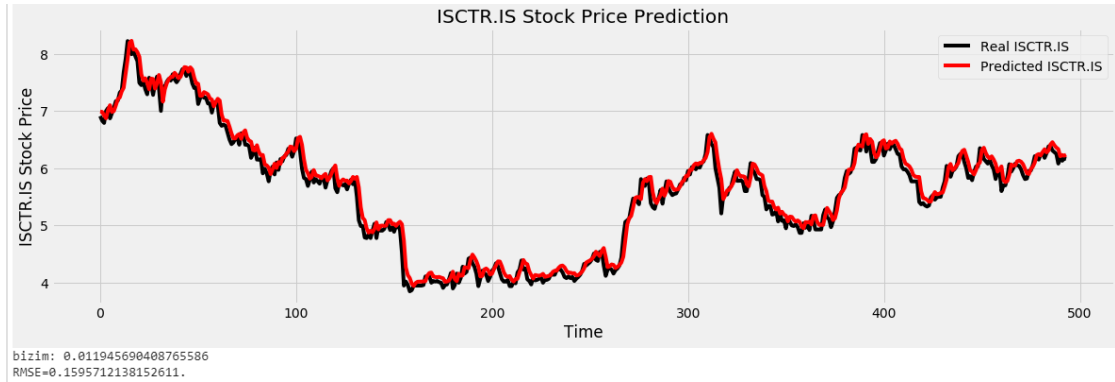
TRTP-RMSE ile pareto optimum yönteminin beraber kullanılmasıyla oluşturulan başarı grafiği Şekil 4.3’de verilmiştir. Performans verileri elde edilirken bütün çalışmalarda aynı parametreler ve aynı hisse senedine ait aynı verikümesi kullanılmıştır. TRTP-RMSE sayesinde en performanslı modelin, TRTP’de yüksek değer, ortalama hata kareleri kökünde düşük değer alması gerekmektedir. Bu koşulları sağlayan en başarılı model ise GRU ve LSTM modellerinin melez olarak kullanılmasıyla elde edilmiştir. Dense ve simpleRNN modelleri istenilenin aksine, daha yüksek ortalama hata kareleri kökü sonucu almış olup, TRTP-RMSE göre de en düşük başarıyı sergilemişlerdir. Tablo 4.3’de bulunan modellerin performans sonuçlarında pencere boyutunun 10, çevrim sayısının 40 olduğu durumda melez modelin RMSE sonucu 0.161 olup GRU’nun da RMSE sonucu 0.161’dir. Tek başına RMSE metriğine göre bu iki model aynı performansı göstermiştir denebilirdi, ancak TRTP-RMSE sonucuna bakıldığında melez modelin sonucu GRU’dan iki kat daha fazladır ve melez model ile GRU’dan daha yüksek başarı elde edilmiştir. RMSE sonuçları aynı olduğundan ve TRTP sonucunun iki katından daha fazla olmasından dolayı melez modelin GRU’dan daha başarılı olduğu görülmektedir. Melez ve GRU modellerindeki bu fark, pencere boyutu 30, çevrim sayısının 60 olduğu testte de görülmektedir. Çevrim sayısının 20 olduğu testler haricinde, melez model bu tez çalışmasında kullanılan diğer derin öğrenme modellerinden daha başarılı olmuştur ve pareto optimum noktasındadır. Pareto optimum noktasının





Şekil 4.3: Modellerin TRTP-RMSE ile kıyaslanması

bir örneği Şekil 4.3’de gösterilmiş olup, tabloya ait verilerin diğer pareto optimum grafikleri ekler bölümünde yer almaktadır. Çevrim sayısının 20 olduğu testlerde melez modelin RMSE sonucunun küçük bir miktar daha kötü olmasına rağmen, TRTP sonuçlarının daha yüksek olmasıyla, yani geçmiş güne ait fiyatların gelecek gün için tahmin olarak değerlendirme eğiliminin daha az olmasıyla, diğer modellerden daha iyi başarı göstermiştir. Böylelikle geliştirdiğimiz TRTP-RMSE ve pareto optimum yönteminin beraber kullanılması ile çok daha doğru performans kıyaslamaları yapılabilir hale gelmiştir.



Şekil 4.4: Model performans karşılaştırması grafiğindeki GRU-LSTM Melez modelinden alınan sonuçlar

## 5 Sonuç ve Değerlendirme

Finansal sektördeki verilerin derin öğrenme ile analiz edilip tahmin edilmesi son yılların popüler konularından biridir. Bunun sebebi finansal verilerin gelecekte ne olacağını tam olarak tahmin etmenin imkansız olmasındandır. Buna rağmen bireysel ve kurumsal yatırımcılar yatırımlarını doğru hisseler yapabilmek için borsanın gelecekteki fiyatlamalarını öngörmek istemektedirler. Aynı zamanda finansal sektördeki bilgi kirliliği ve spekülasyonlar da öngörülerini olumsuz etkilemektedir. Borsadaki hisselerin gelecekte nasıl fiyatlayacağı insanların davranışlarıyla bile değişebilen bir yapıdadır. Bundan dolayı finansal sektörde geliştirilecek derin öğrenme modelleri, gelecekte finansal verilerin tahmini için önemlidir.

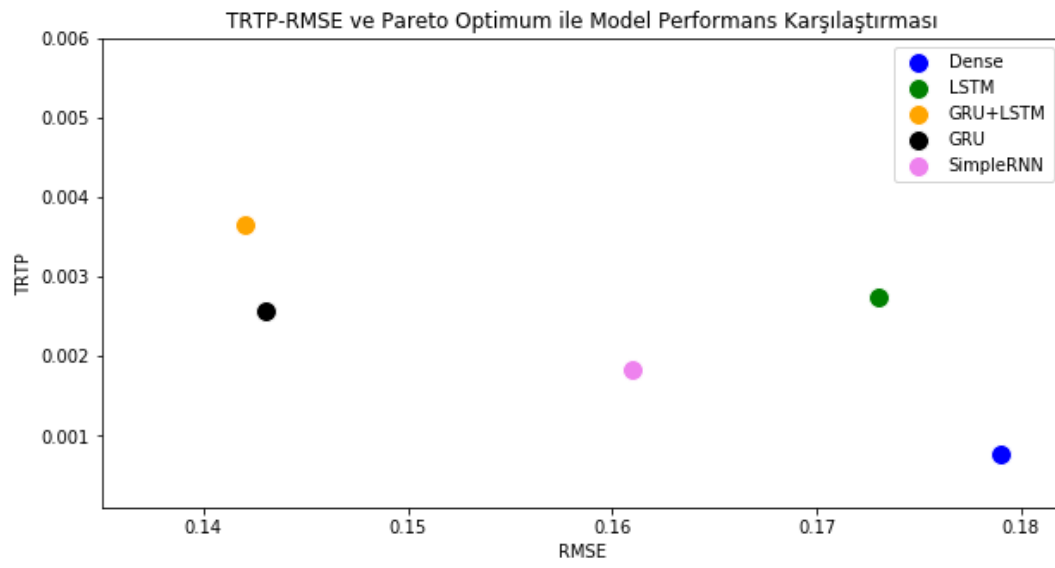
Bu tez çalışmasında, zaman serisi problemlerinde yaygın olarak kullanılan kök ortalama kare hatası gibi metriklerin tek başına derin öğrenme modelinin başarımını göstermek için yeterli olmadığı ispatlanmıştır. Kök ortalama kare hatasının çok düşük olduğu durumlar için model çok başarılı denebilmektedir, ancak bu tez çalışmasında geliştirilen TRTP-RMSE sayesinde sadece kök ortalama kare hatasına göre derin öğrenme modellerinin performanslarının kıyaslanmasının hatalı sonuçlar çıkaracağı ortaya çıkarılmıştır. Bu çalışma kapsamında, bazı derin öğrenme modellerinin bir önceki günün piyasa verisini bir sonraki gün için birebir tahmin olarak değerlendirme eğilimleri görülmüş olup tek başına kullanılan RMSE gibi geleneksel hata ölçüm metriklerinde, modellerin gerçekleştirdiği bu naif ve etkisiz yaklaşımın yakalanamayıp, bu modellerin tahmin yeteneklerinin yanıltıcı olarak çok başarılıymış gibi görüldüğü ortaya çıkmıştır. Bu durumun çözümü için ise TRTP-RMSE yöntemi geliştirilmiştir. Yapılacak olan kıyaslamalarda sadece kök ortalama kare hatasının düşüklüğüne dikkat edilmesi yerine, bu tez çalışmasında önerdiğimiz TRTP metriğinden alınan sonucun yüksek olması, kök ortalama kare hatasının ise mümkün olduğunca düşük olması ile pareto optimum bölgesindeki sonuçlara göre derin öğrenme modelinin başarılı veya başarısız olma durumu gözlemlenmelidir. Sadece kök ortalama kare hatasına göre derin öğrenme modellerinin başarımları kıyaslandığında, gerçekte başarısız olan modellerin başarılıymış gibi görünmesine sebep olmaktadır. Bu tez çalışmasında önerdiğimiz TRTP-

RMSE ise bu soruna çözüm sunmaktadır. Gelecekte yapılacak olan çalışmalar için de kullanılabilecek olan bu yöntem, gelecek çalışmalarda kullanılacak olan derin öğrenme modellerinin başarımı hususunda önemli bilgiler verebilecek durumdadır.

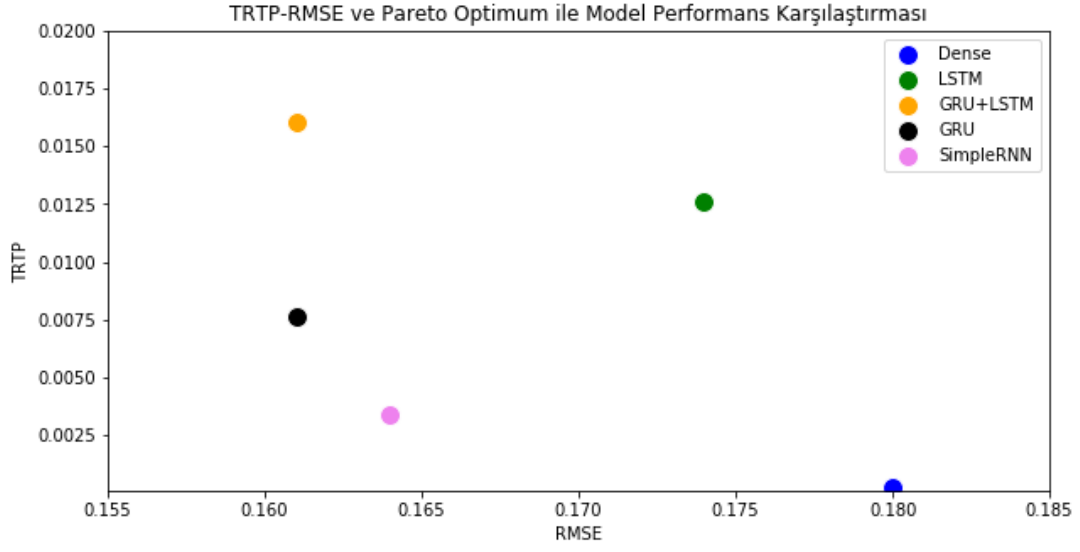
Çalışma kapsamında derin öğrenme modelleri kullanılarak geçmiş yıllara ait hisse senedi fiyat verilerinin kullanılmasıyla ileri zamanlardaki fiyatlamaların tahmin edilebileceği görülmüştür. Bu tez çalışmasında geliştirilen TRTP-RMSE yöntemi sonuçlarına göre melez yapı, diğer derin öğrenme modellerinden daha iyi başarımlar göstermektedir. Modelde borsa hisselerinin geçmiş verilerinin yanında girdi olarak döviz kurlarının artış ve azalışının borsaya olan etkisi de eklendiğinde daha başarılı sonuçlar almak mümkündür. Ayrıca bir sonraki kapanış fiyatlamaları tahmininde fiyatlamamın yön tahmini doğru çıkmakta ancak artış veya azalış miktarı hakkında daha fazla çalışmaya ihtiyaç duyulmaktadır. Örneğin, bir hisse senedinin %2 artacağı tahmin edilen senaryoda, fiyatın %2 yerine %8 veya daha fazla arttığı görülmüştür. Bunun başlıca sebepleri ise günlük haber akışları, insanların yatırım davranışları veya şirketlerden gelen önemli haberlerin etkisi dalgalanmanın boyutu ciddi artırmaktadır. Bu problemin önüne geçilebilmesi için, gelecek çalışmalarda yerli ve yabancı haber kaynaklarından canlı veriler çekilerek, ilgili hisselerle ait verilerin seçilip değerlendirmeye alınarak sadece geçmiş verilere endeksli bir model kullanılmayıp, haber akışını da anlık takip eden bir model geliştirilerek ani düşüş ve yükselişlerin de doğru tahmin edilme çalışmaları yapılabilir.

Finansal verilerin gelecekteki fiyatlamalarının tahmin edilmesi borsa var olduğundan beri yapılmaya çalışılıyor ancak teknolojinin gelişmesiyle bilgisayarın daha fazla işlem kapasitesine sahip olmasıyla derin öğrenme ile fiyat analizinin yapılıp tahmin yürütülmesi son yıllarda popülerlik kazanmıştır. Bu çalışma gelecekte yapılacak olan çalışmaların temelini oluşturabilir ve henüz zaman-serilerinde kullanılan derin öğrenme modellerinin performans ölçümleri için pek araştırma bulunmamaktadır. Özellikle derin öğrenme modellerinin performans kıyaslamaları için çoğunlukla geleneksel modeller tercih edilirken, bu geleneksel metriklerin zaman-serileri problemleri için tek başlarına yeterli olmadıkları ispatlanmıştır. Finansal sektörün ise finansal verilerin tahmininde kararlı, adeta yatırım danışmanı gibi çalışan ve her zaman doğru yatırımlara yönlendiren, bilgi kirliliğinden ve insan duygusallığından uzak yapay zeka modellerine ihtiyacı vardır. Bundan dolayı bu çalışma sayesinde ilerleyen zamanlarda daha kararlı tahminler yapabilen yapılar oluşturulabilir.

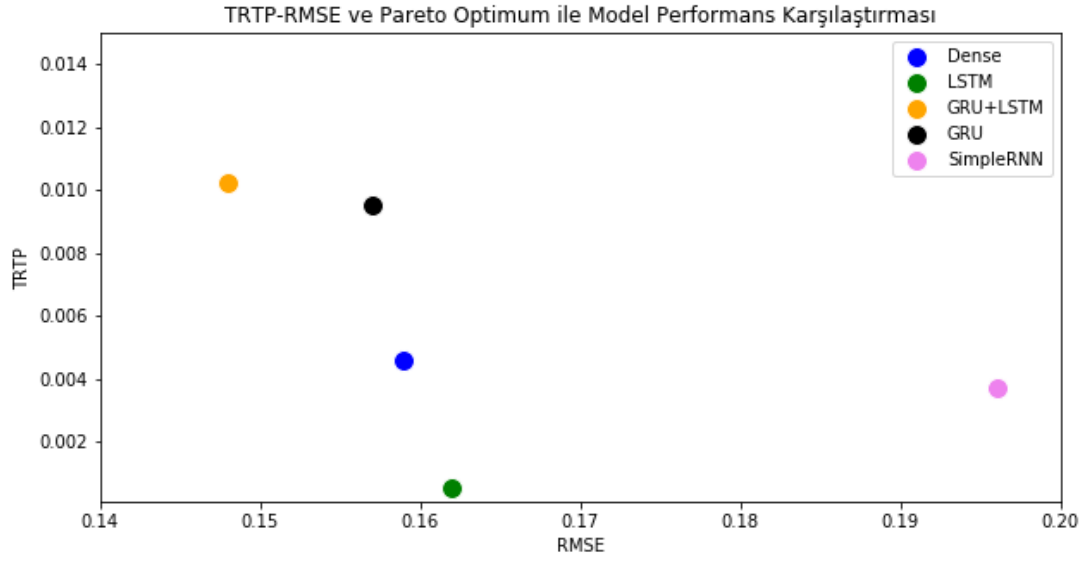
## 6 Ekler



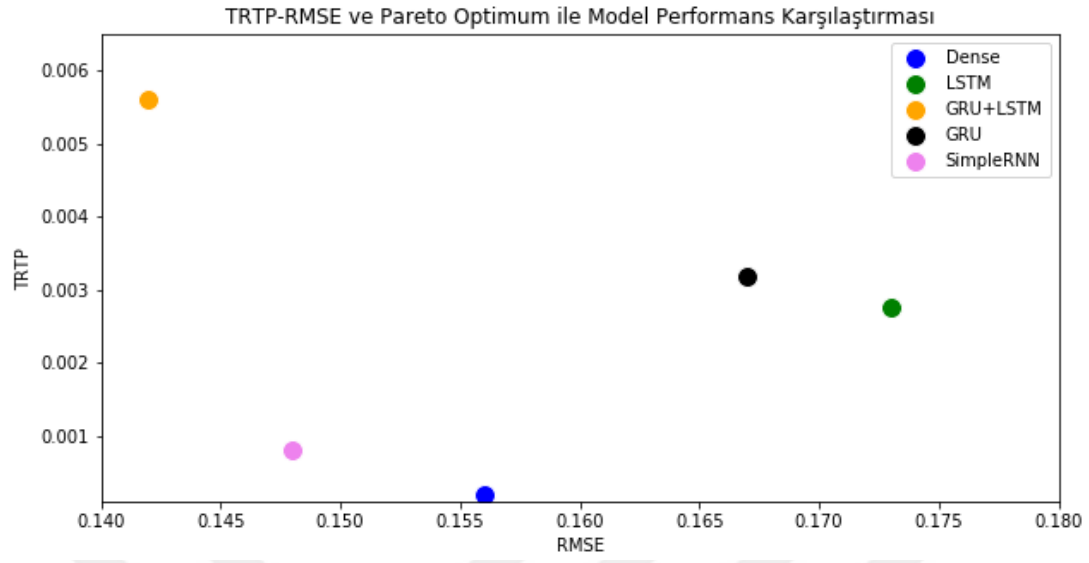
Şekil 6.1: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



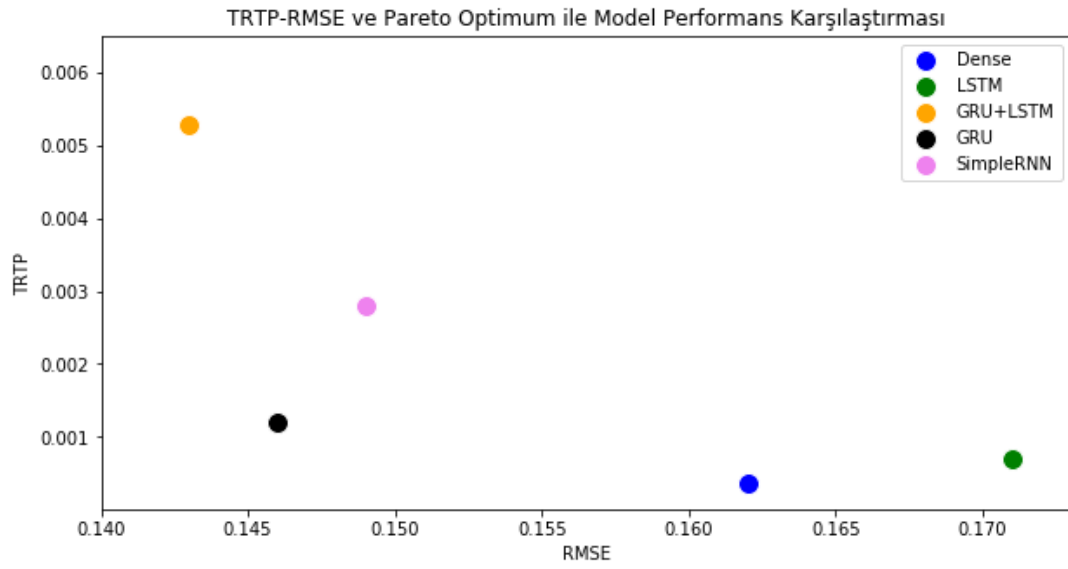
Şekil 6.2: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



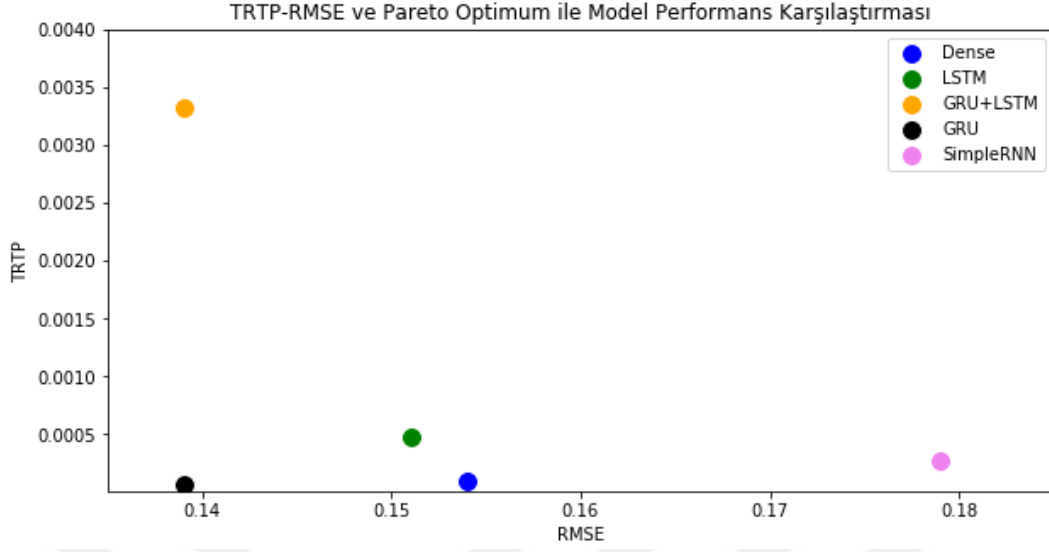
Şekil 6.3: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



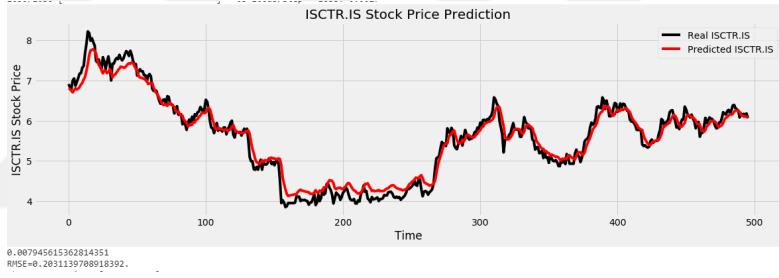
Şekil 6.4: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



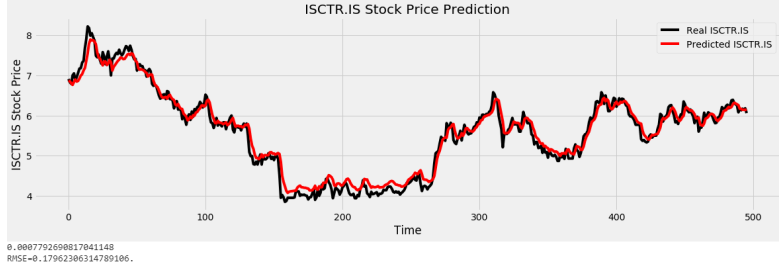
Şekil 6.5: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



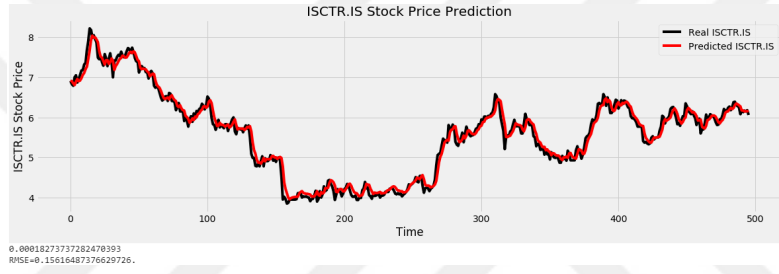
Şekil 6.6: Tablo 4.3'de Bulunan Sonuçların TRTP-RMSE Grafiği



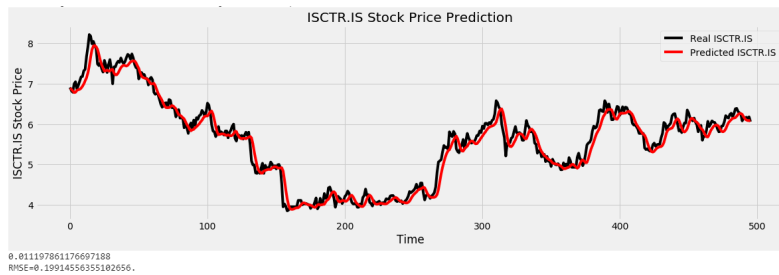
Şekil 6.7: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,2031 çıkarken, önerdiğimiz TRTP metriği 0,0079 olarak hesaplanmıştır.



Şekil 6.8: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1796 çıkarken, önerdiğimiz TRTP metriği 0,00077 olarak hesaplanmıştır.

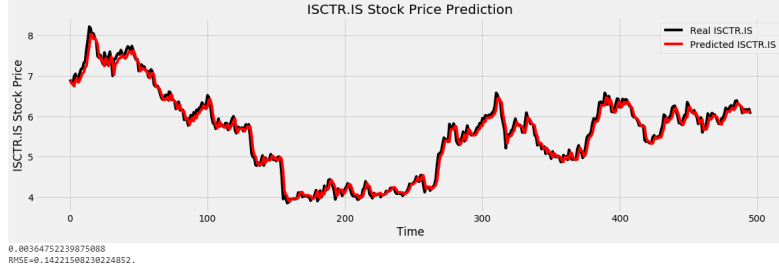


Şekil 6.9: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1561 çıkarken, önerdiğimiz TRTP metriği 0,00018 olarak hesaplanmıştır.

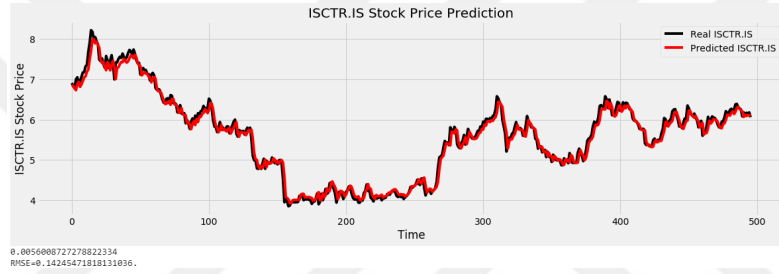


Şekil 6.10: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1991 çıkarken, önerdiğimiz TRTP metriği 0,01119 olarak hesaplanmıştır.

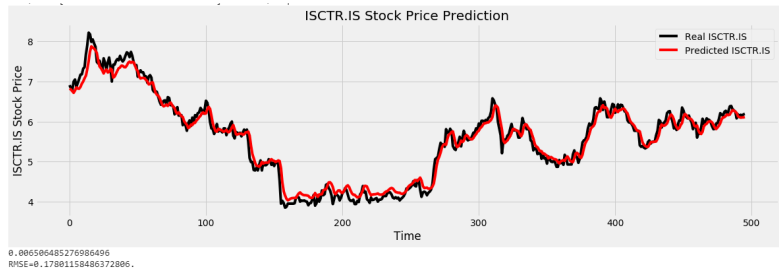




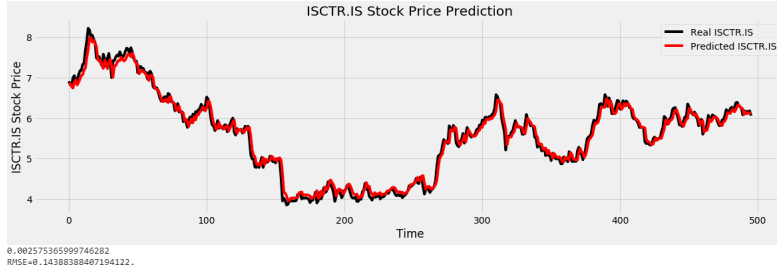
Şekil 6.11: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1422 çıkarken, önerdiğimiz TRTP metriği 0,00364 olarak hesaplanmıştır.



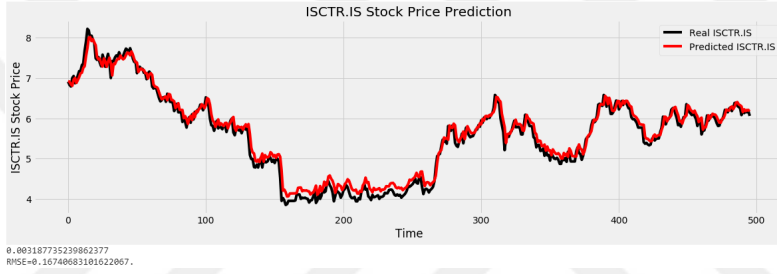
Şekil 6.12: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1424 çıkarken, önerdiğimiz TRTP metriği 0,0056 olarak hesaplanmıştır.



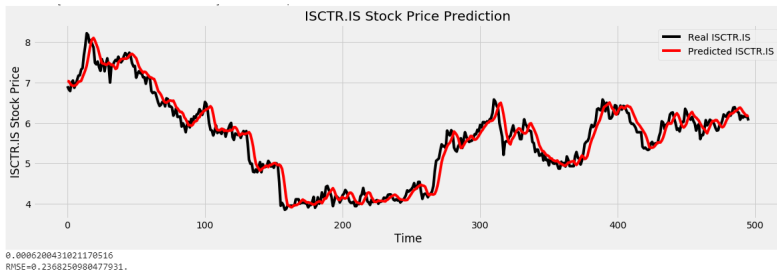
Şekil 6.13: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1780 çıkarken, önerdiğimiz TRTP metriği 0,0065 olarak hesaplanmıştır.



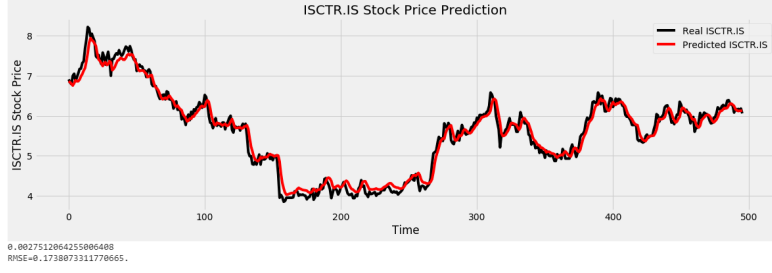
Şekil 6.14: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1438 çıkarken, önerdiğimiz TRTP metriği 0,0025 olarak hesaplanmıştır.



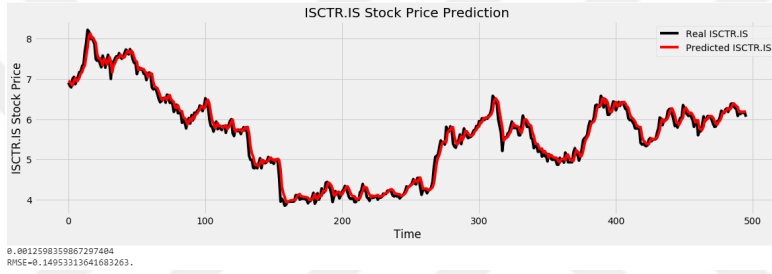
Şekil 6.15: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1674 çıkarken, önerdiğimiz TRTP metriği 0,0031 olarak hesaplanmıştır.



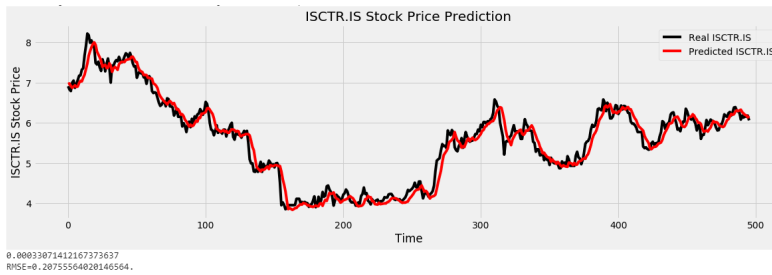
Şekil 6.16: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,2368 çıkarken, önerdiğimiz TRTP metriği 0,00062 olarak hesaplanmıştır.



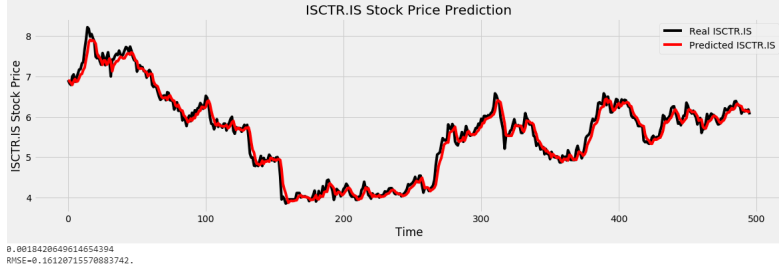
Şekil 6.17: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,173 çıkarken, önerdiğimiz TRTP metriği 0,00275 olarak hesaplanmıştır.



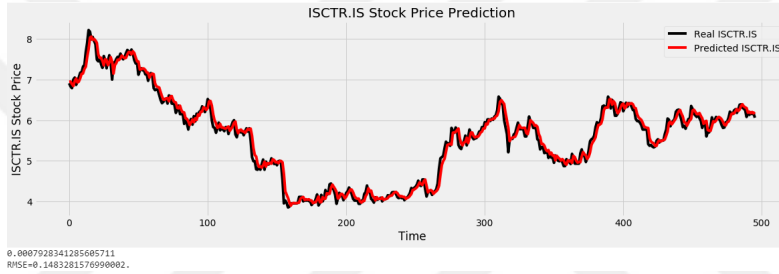
Şekil 6.18: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1495 çıkarken, önerdiğimiz TRTP metriği 0,00125 olarak hesaplanmıştır.



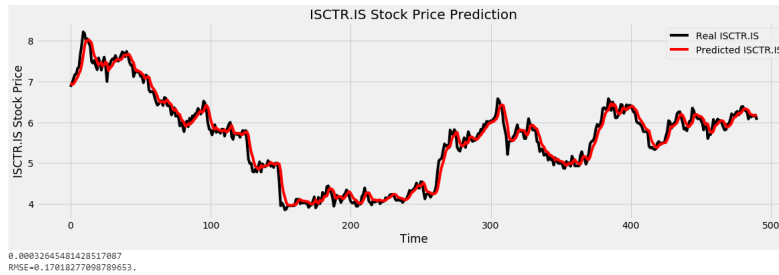
Şekil 6.19: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,2075 çıkarken, önerdiğimiz TRTP metriği 0,00033 olarak hesaplanmıştır.



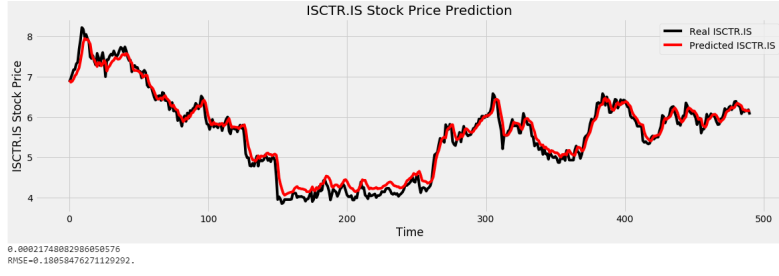
Şekil 6.20: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1612 çıkarken, önerdiğimiz TRTP metriği 0,00184 olarak hesaplanmıştır.



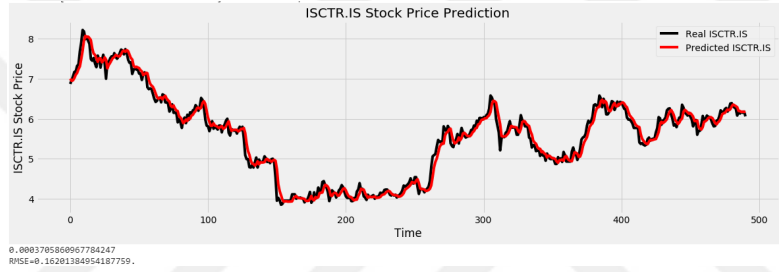
Şekil 6.21: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 5 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1483 çıkarken, önerdiğimiz TRTP metriği 0,00079 olarak hesaplanmıştır.



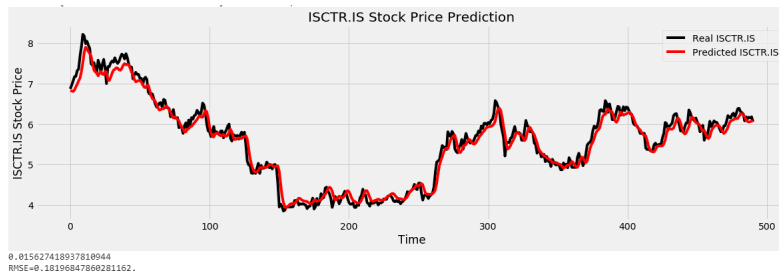
Şekil 6.22: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,170 çıkarken, önerdiğimiz TRTP metriği 0,00032 olarak hesaplanmıştır.



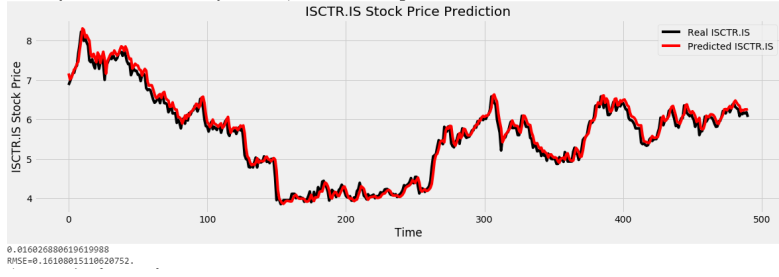
Şekil 6.23: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,1805 çıkarken, önerdiğimiz TRTP metriği 0,00021 olarak hesaplanmıştır.



Şekil 6.24: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,162 çıkarken, önerdiğimiz TRTP metriği 0,00037 olarak hesaplanmıştır.



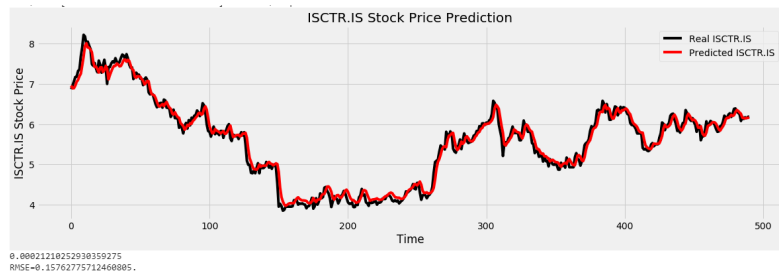
Şekil 6.25: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,181 çıkarken, önerdiğimiz TRTP metriği 0,01562 olarak hesaplanmıştır.



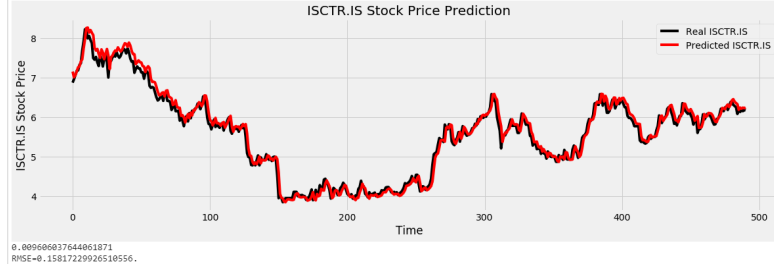
Şekil 6.26: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,161 çıkarken, önerdiğimiz TRTP metriği 0,01602 olarak hesaplanmıştır.



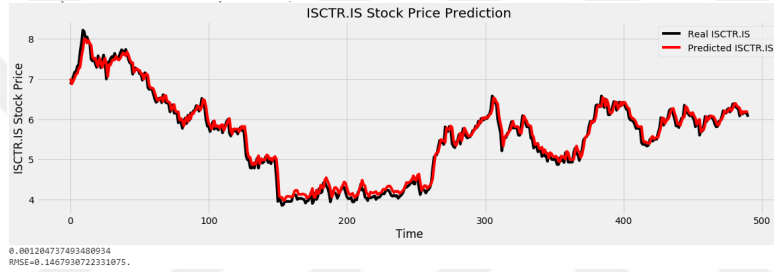
Şekil 6.27: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,143 çıkarken, önerdiğimiz TRTP metriği 0,005287 olarak hesaplanmıştır.



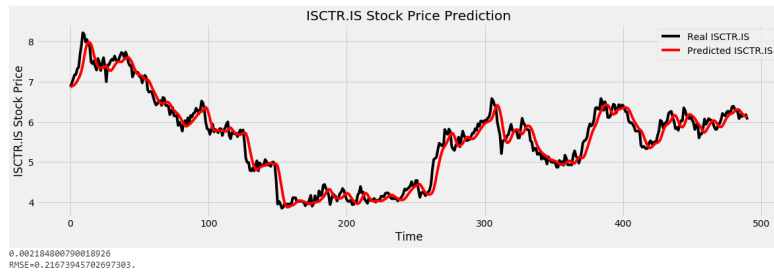
Şekil 6.28: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,157 çıkarken, önerdiğimiz TRTP metriği 0,00021 olarak hesaplanmıştır.



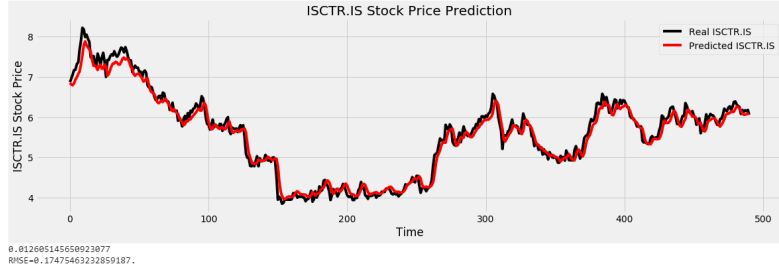
Şekil 6.29: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,158 çıkarken, önerdiğimiz TRTP metriği 0,0096 olarak hesaplanmıştır.



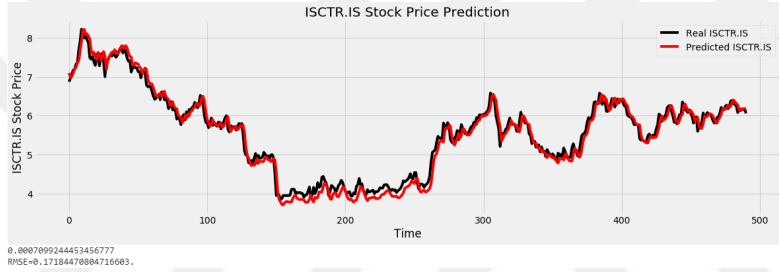
Şekil 6.30: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,146 çıkarken, önerdiğimiz TRTP metriği 0,001204 olarak hesaplanmıştır.



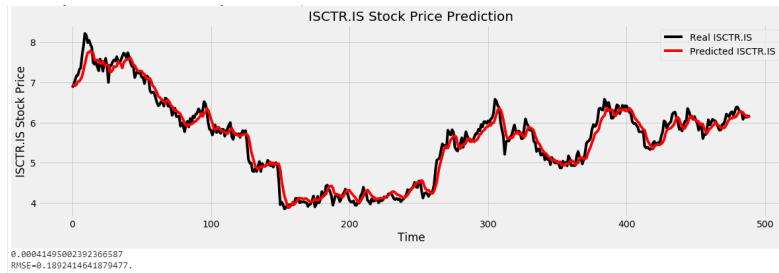
Şekil 6.31: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,216 çıkarken, önerdiğimiz TRTP metriği 0,00218 olarak hesaplanmıştır.



Şekil 6.32: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,174 çıkarken, önerdiğimiz TRTP metriği 0,01260 olarak hesaplanmıştır.

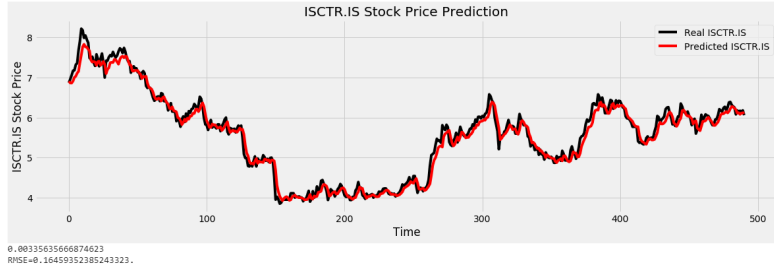


Şekil 6.33: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,171 çıkarken, önerdiğimiz TRTP metriği 0,000709 olarak hesaplanmıştır.

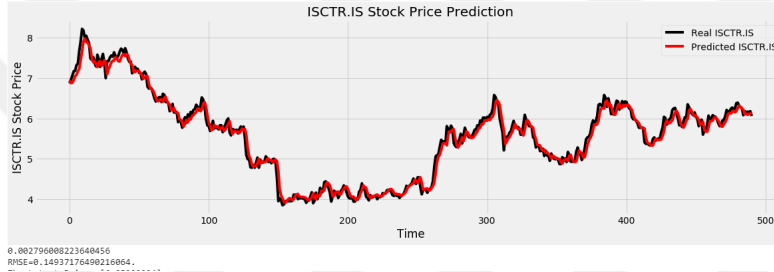


Şekil 6.34: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,189 çıkarken, önerdiğimiz TRTP metriği 0,00041 olarak hesaplanmıştır.

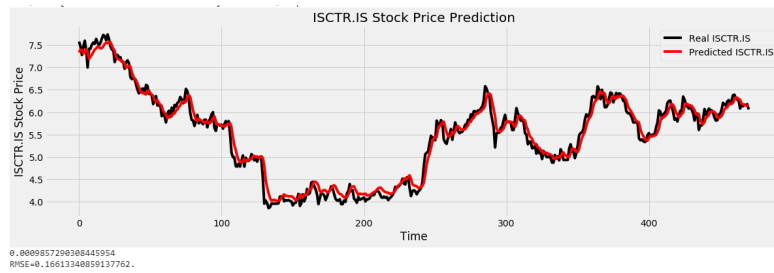




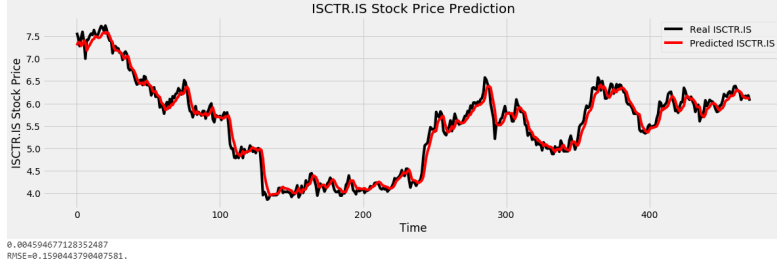
Şekil 6.35: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,164 çıkarken, önerdiğimiz TRTP metriği 0,003356 olarak hesaplanmıştır.



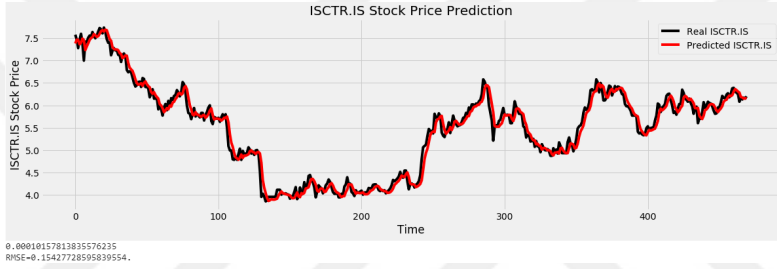
Şekil 6.36: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 10 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,149 çıkarken, önerdiğimiz TRTP metriği 0,00279 olarak hesaplanmıştır.



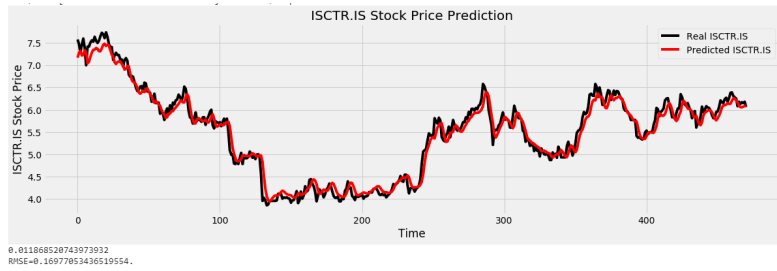
Şekil 6.37: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,166 çıkarken, önerdiğimiz TRTP metriği 0,00098 olarak hesaplanmıştır.



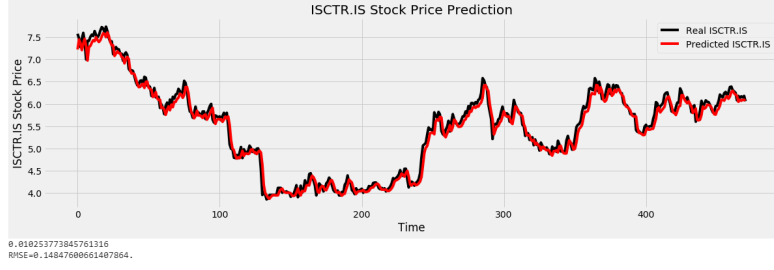
Şekil 6.38: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,159 çıkarken, önerdiğimiz TRTP metriği 0,004594 olarak hesaplanmıştır.



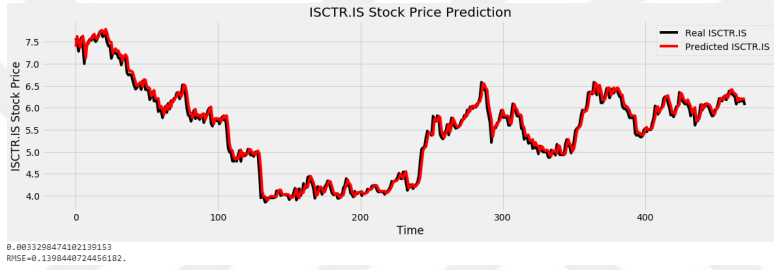
Şekil 6.39: Bütün katmanları dense tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,154 çıkarken, önerdiğimiz TRTP metriği 0,000101 olarak hesaplanmıştır.



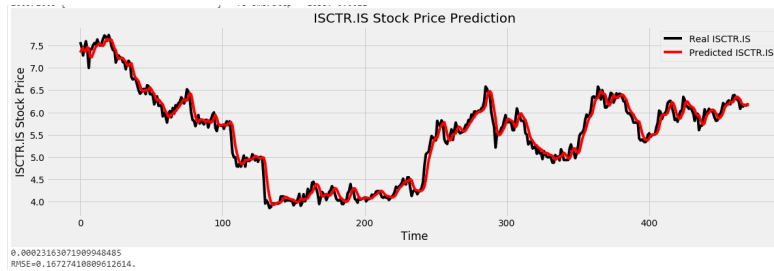
Şekil 6.40: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,169 çıkarken, önerdiğimiz TRTP metriği 0,01186 olarak hesaplanmıştır.



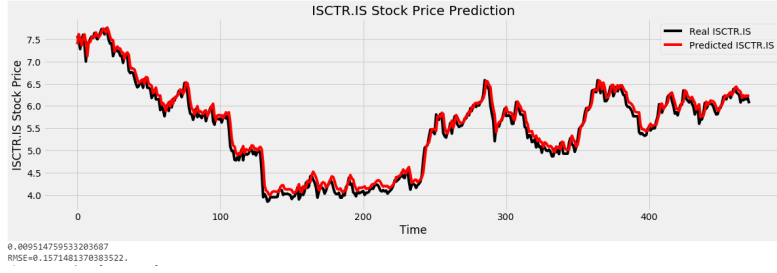
Şekil 6.41: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,148 çıkarken, önerdiğimiz TRTP metriği 0,010253 olarak hesaplanmıştır.



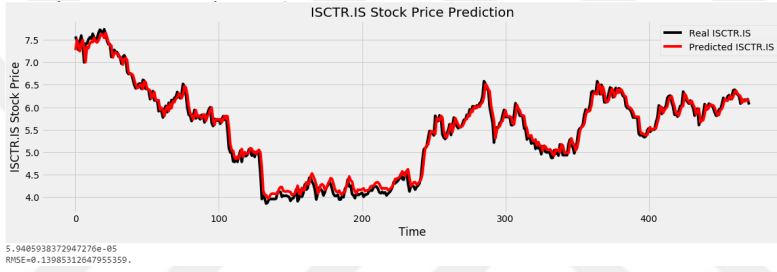
Şekil 6.42: Bütün katmanları GRU ve LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,139 çıkarken, önerdiğimiz TRTP metriği 0,003329 olarak hesaplanmıştır.



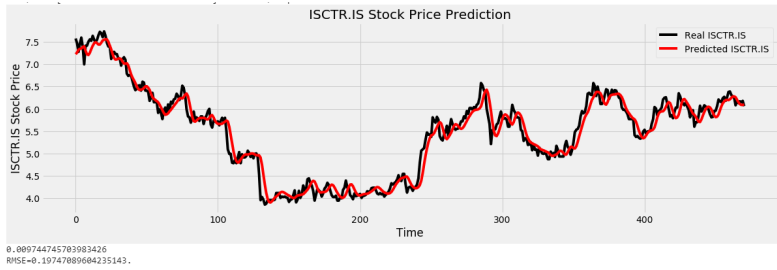
Şekil 6.43: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,167 çıkarken, önerdiğimiz TRTP metriği 0,000231 olarak hesaplanmıştır.



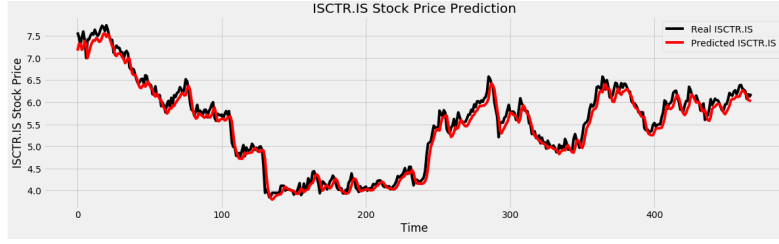
Şekil 6.44: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,157 çıkarken, önerdiğimiz TRTP metriği 0,009514 olarak hesaplanmıştır.



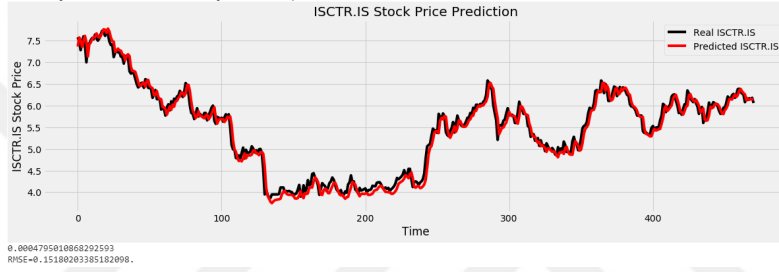
Şekil 6.45: Bütün katmanları GRU tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,139 çıkarken, önerdiğimiz TRTP metriği 0,00005 olarak hesaplanmıştır.



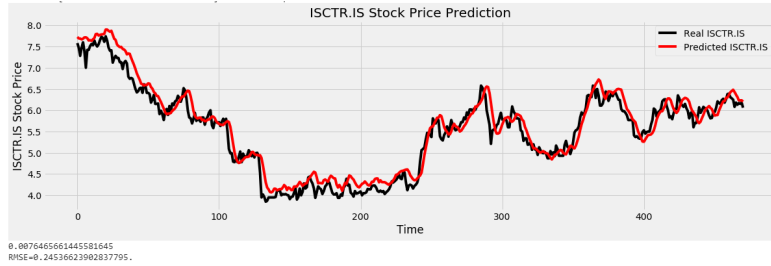
Şekil 6.46: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,197 çıkarken, önerdiğimiz TRTP metriği 0,009744 olarak hesaplanmıştır.



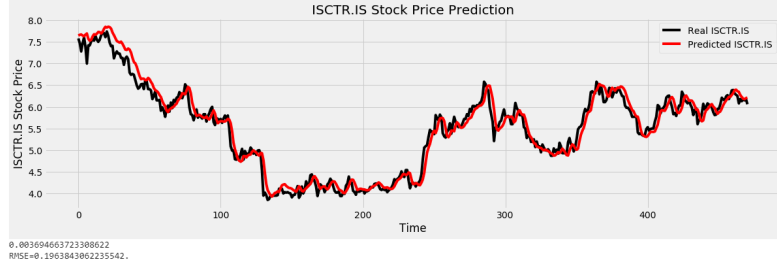
Şekil 6.47: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,162 çıkarken, önerdiğimiz TRTP metriği 0,000552 olarak hesaplanmıştır.



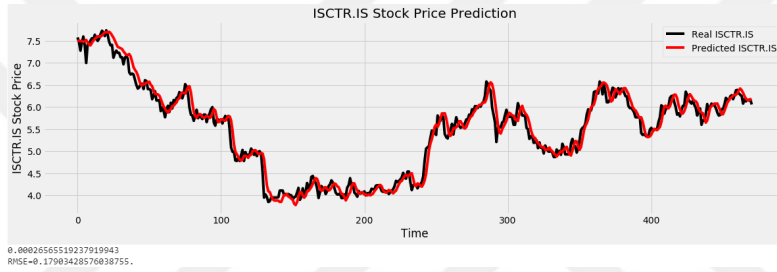
Şekil 6.48: Bütün katmanları LSTM tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,151 çıkarken, önerdiğimiz TRTP metriği 0,000479 olarak hesaplanmıştır.



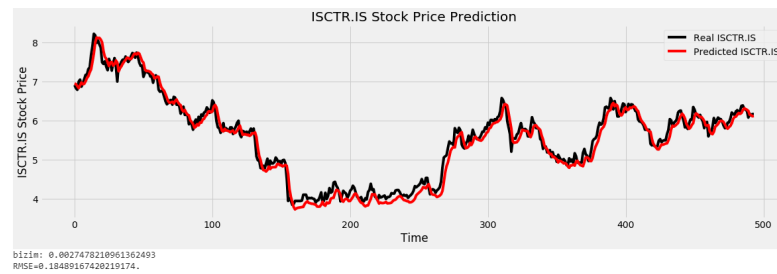
Şekil 6.49: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 20 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,245 çıkarken, önerdiğimiz TRTP metriği 0,007646 olarak hesaplanmıştır.



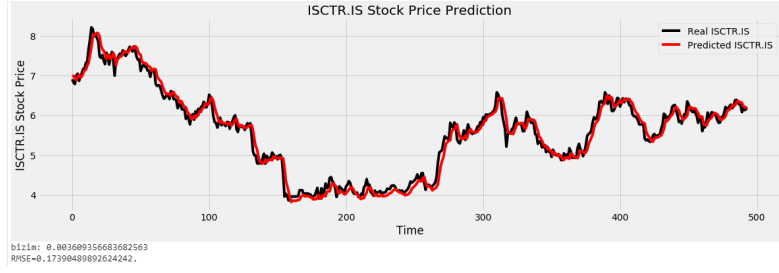
Şekil 6.50: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 40 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,196 çıkarken, önerdiğimiz TRTP metriği 0,003694 olarak hesaplanmıştır.



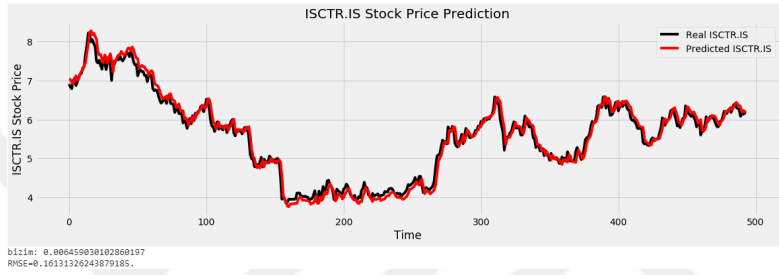
Şekil 6.51: Bütün katmanları simpleRNN tipi nöronlarla oluşturulmuş derin öğrenme mimarisinin 60 çevrim sayısı ile ve geriye dönük 30 günlük zaman penceresi üzerinden eğitilmiş modelin test performansı. Kırmızı grafik tahmin, siyah grafik ise gerçek verileri göstermektedir. RMSE değeri 0,179 çıkarken, önerdiğimiz TRTP metriği 0,000265 olarak hesaplanmıştır.



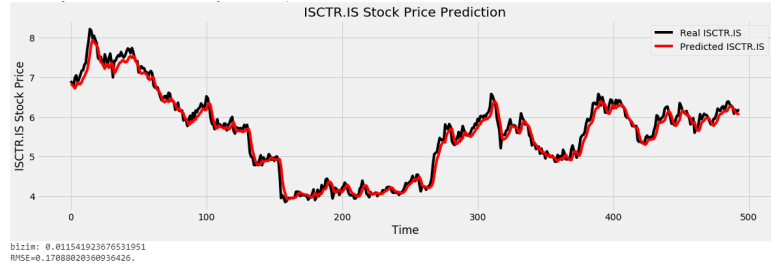
Şekil 6.52: Model performans karşılaştırması grafiğindeki dense modelinden alınan sonuçta RMSE değeri 0.184 çıkarken, önerdiğimiz TRTP metriği 0.00274 olarak hesaplanmıştır.



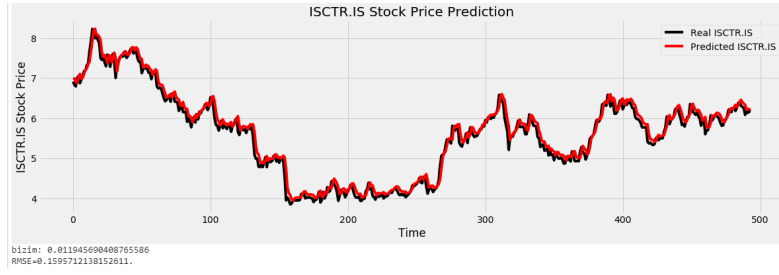
Şekil 6.53: Model performans karşılaştırması grafiğindeki simpleRNN modelinden alınan sonuçta RMSE değeri 0.173 çıkarken, önerdiğimiz TRTP metriği 0.0036 olarak hesaplanmıştır.



Şekil 6.54: Model performans karşılaştırması grafiğindeki GRU modelinden alınan sonuçta RMSE değeri 0.161 çıkarken, önerdiğimiz TRTP metriği 0.00645 olarak hesaplanmıştır.



Şekil 6.55: Model performans karşılaştırması grafiğindeki LSTM modelinden alınan sonuçta RMSE değeri 0.184 çıkarken, önerdiğimiz TRTP metriği 0.00274 olarak hesaplanmıştır.



Şekil 6.56: Model performans karşılaştırması grafiğindeki GRU ve LSTM melez modelinden alınan sonuçta RMSE değeri 0.159 çıkarken, önerdiğimiz TRTP metriği 0.0119456 olarak hesaplanmıştır.

```

1 !pip install pandas-datareader
2 !pip install fix-yahoo-finance
3 import tensorflow.compat.v1 as tf
4 tf.disable_v2_behavior()
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 import numpy as np
8 import math
9 from keras.models import Sequential
10 from keras.layers import Dense, LSTM, Dropout, GRU, CuDNNLSTM, SimpleRNN
11 from sklearn.preprocessing import MinMaxScaler
12 from sklearn.metrics import mean_squared_error
13 from keras import optimizers
14 from pandas_datareader import data as pdr
15 import fix_yahoo_finance as yf

```

Kod 6.1: Kütüphanelerin alınması

```

1 stock_share = "ISCTR.IS"
2 ds = pdr.get_data_yahoo(stock_share)
3 ds = ds.dropna()
4 ds.tail()

```

Kod 6.2: Yahoo Finance'den veri kümelerinin alınması



```

1 test_set = ds['2018:'].iloc[:,3:4].values
2 training_set = ds['2010':'2017'].iloc[:,3:4].values
3 print(test_set)
4 print(training_set)
5 ds[['Close']]['2010':'2017'].plot(figsize=(16,4),legend=True)
6 ds[['Close']]['2018:'].plot(figsize=(16,4),legend=True)
7 plt.legend(['Training set','Test set'])
8 plt.title(stock_share+'stock price')
9 plt.show()

```

Kod 6.3: Test ve eğitim verilerinin seçilmesi ve kontrol edilmesi

```

1 sc = MinMaxScaler(feature_range = (0, 1))
2 training_fit = sc.fit(training_set)
3 training_scaled = training_fit.transform(training_set)
4 X_train = []
5 y_train = []
6 windowsize = 30 #çalışma esnasında pencere boyutu 5–10–30 olarak kullanılmıştır.
7 for i in range(windowsize,len(training_scaled)):
8     X_train.append(training_scaled[i–windowsize:i,0])
9     y_train.append(training_scaled[i,0])
10 X_train = np.array(X_train)
11 y_train = np.array(y_train)
12 X_train = np.reshape(X_train, (X_train.shape[0],X_train.shape[1],1))
13 X_train.shape

```

Kod 6.4: Eğitim verisinin ölçeklendirilmesi ve yeniden şekillendirilmesi ile modelde kullanılan pencere boyutunun belirlenmesi [34]

```

1 model.add(GRU(units=75, return_sequences=True, input_shape=(X_train.shape[1],1)))
2 model.add(Dropout(0.2))
3 model.add(LSTM(units=150, return_sequences=True))
4 model.add(Dropout(0.2))
5 model.add(LSTM(units=150, return_sequences=True))
6 model.add(Dropout(0.2))
7 model.add(GRU(units=75))
8 model.add(Dropout(0.2))

```

```

9 model.add(Dense(units=1))
10 model.compile(optimizer='adam', loss='mean_squared_error')
11 model.fit(X_train,y_train,epochs=20,batch_size=32, validation_split = 0.2 , shuffle =
    ↪ True)

```

Kod 6.5: Melez modelin oluşturulması ve derlenmesi

```

1 model.add(Dense(units=75, input_shape=(X_train.shape[1],1)))
2 model.add(Dropout(0.2))
3 model.add(Dense(units=150))
4 model.add(Dropout(0.2))
5 model.add(Dense(units=150))
6 model.add(Dropout(0.2))
7 model.add(Dense(units=75))
8 model.add(Dropout(0.2))
9 model.add(Flatten())
10 model.add(Dense(units=1))
11 model.compile(optimizer='adam', loss='mean_squared_error')
12 model.fit(X_train,y_train,epochs=20,batch_size=32, validation_split = 0.2 , shuffle =
    ↪ True)

```

Kod 6.6: Dense modelin oluşturulması ve derlenmesi

```

1 model.add(GRU(units=75, return_sequences=True, input_shape=(X_train.shape[1],1)))
2 model.add(Dropout(0.2))
3 model.add(GRU(units=150, return_sequences=True))
4 model.add(Dropout(0.2))
5 model.add(GRU(units=150, return_sequences=True))
6 model.add(Dropout(0.2))
7 model.add(GRU(units=75))
8 model.add(Dropout(0.2))
9 model.add(Dense(units=1))
10 model.compile(optimizer='adam', loss='mean_squared_error')
11 model.fit(X_train,y_train,epochs=20,batch_size=32, validation_split = 0.2 , shuffle =
    ↪ True)

```

Kod 6.7: GRU modelin oluşturulması ve derlenmesi

```

1 model.add(SimpleRNN(units=75, return_sequences=True, input_shape=(X_train.shape
  ↪ [1],1)))
2 model.add(Dropout(0.2))
3 model.add(SimpleRNN(units=150, return_sequences=True))
4 model.add(Dropout(0.2))
5 model.add(SimpleRNN(units=150, return_sequences=True))
6 model.add(Dropout(0.2))
7 model.add(SimpleRNN(units=75))
8 model.add(Dropout(0.2))
9 model.add(Dense(units=1))
10 model.compile(optimizer='adam', loss='mean_squared_error')
11 model.fit(X_train,y_train,epochs=20,batch_size=32, validation_split = 0.2 , shuffle =
  ↪ True)

```

Kod 6.8: SimpleRNN modelin oluşturulması ve derlenmesi

```

1 model.add(LSTM(units=75, return_sequences=True, input_shape=(X_train.shape[1],1)
  ↪ ))
2 model.add(Dropout(0.2))
3 model.add(LSTM(units=150, return_sequences=True))
4 model.add(Dropout(0.2))
5 model.add(LSTM(units=150, return_sequences=True))
6 model.add(Dropout(0.2))
7 model.add(LSTM(units=75))
8 model.add(Dropout(0.2))
9 model.add(Dense(units=1))
10 model.compile(optimizer='adam', loss='mean_squared_error')
11 model.fit(X_train,y_train,epochs=20,batch_size=32, validation_split = 0.2 , shuffle =
  ↪ True)

```

Kod 6.9: LSTM modelin oluşturulması ve derlenmesi

```

1 sc = MinMaxScaler(feature_range = (0,1))
2 test_fit = sc.fit(test_set)
3 test_set_scaled = test_fit.transform(test_set)
4 X_test = []

```

```

5 for i in range(windowsize,len(test_scaled)):
6     X_test.append(test_scaled[i-windowsize:i,0])
7 X_test = np.array(X_test)
8 X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
9 prediction = model.predict(X_test)
10 prediction = sc.inverse_transform(predicted_stock_price)

```

Kod 7.0: X Test değişkeninin hazırlanması ve modelin test edilmesi

```

1 err = []
2 a = 0
3 b = 0
4 test_set2 = test_set>windowsize:]
5 for x in range(1,len(test_set2)):
6     a = predicted_stock_price[x]-test_set2[x-1]
7     a = math.pow(a,2)
8     err.append(a)
9 for y in range(1,len(err)):
10     b = err[y] + err[y-1]
11 print(b)

```

Kod 7.1: Geliştirdiğimiz TRTP metriğinin hesaplanması

```

1 plt.figure(figsize=(20,5))
2 plt.plot(prediction[:,], label='Predicted '+stock_share, color='red')
3 plt.plot(test_set[:,], label='Real '+stock_share, color='black')
4 plt.title(share+' Stock Price Prediction')
5 plt.xlabel('Time')
6 plt.ylabel(share+' Stock Price')
7 plt.legend()
8 plt.show()
9 rmse = math.sqrt(mean_squared_error(test_set, prediction))
10 print("RMSE={}".format(rmse))

```

Kod 7.2: Sonucun görselleştirilmesi ve RMSE hesaplanması [35]

# Kaynaklar

- [1] Kelly Morgan and Cormac O. Grada. Market Contagion: Evidence from the Panics of 1854 and 1857. *American Economic Review*, 90, 2000.
- [2] Larry Schweikart Charles W. Calomiris. The Panic of 1857: Origins, Transmission, and Containment. *Cambridge University Press*, 51, 2009.
- [3] Ömer Kaan Baykan Yakup Kara, Melek Acar Boyacioglu. Predicting Direction of Stock Price Index Movement using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange. *Science Direct*, 38:5311–5319, 2011.
- [4] Simon Caton Sean McNally, Jason Roche. Predicting the Price of Bitcoin Using Machine Learning. *Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 2018.
- [5] Rich Wolski Ian Foster Kyle Chard Matt Baughman, Christian Haas. Predicting Amazon Spot Prices with LSTM Networks. *ACM New York*, 2018.
- [6] Enda Barrett Deepak Janardhanan. CPU Workload Forecasting of Machines in Data Centers using LSTM Recurrent Neural Networks and ARIMA Models. 2017.
- [7] Sima Siami-Namini Akbar Siami Namin. Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. 2018.
- [8] G.PeterZhang. Time Series Forecasting using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 50:159–175, 2003.
- [9] Homa Baradaran Hashemi Mahdi Pakdaman Naeini, Hamidreza Taremian. Stock Market Value Prediction using Neural Networks. *The Institute of Electrical and Electronics Engineers*, 2010.
- [10] Ebru Şeyma Karakoyun. Derin Öğrenme ile Zaman Serilerinin Gerçek Zamanlı Tahmini. Master's thesis, Necmettin Erbakan Üniversitesi, 2018.

- [11] Chih-ShengLin Ping-FengPai. A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting. *Science Direct*, 33, 2005.
- [12] B. Nath M.R. Hassan. Stock Market Forecasting using Hidden Markov Model: A New Approach. *5th International Conference on Intelligent Systems Design and Applications (ISDA '05)*, 2005.
- [13] Kumkum Garg Rohit Choudhry. A Hybrid Machine Learning System for Stock Market Forecasting . *World Academy of Science, Engineering and Technology 39*, 2008.
- [14] Wang S.-P. Tsai C.-F. Stock Price Forecasting by Hybrid Machine Learning Techniques . *Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS 2009*, 2009.
- [15] Nagesh Singh Chauhan. Introduction to Artificial Neural Networks(ANN). <https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9>, 2016.
- [16] OSun-Chong Wang. *Artificial Neural Network. In: Interdisciplinary Computing in Java Programming*. Springer, Boston, MA, 2003.
- [17] Stacey Ronaghan. Deep Learning: Which Loss and Activation Functions should I use? <https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8>, 2018.
- [18] Andrew Y. Ng Andrew L. Maas, Awni Y. Hannun. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Stanford University*, 2013.
- [19] Raimi Karim. Animated RNN, LSTM and GRU. <https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45>, 2018.
- [20] Jürgen Schmidhuber Sepp Hochreiter. Long Short-Term Memory. *Massachusetts Institute of Technology Press Journals*, 9, 1997.
- [21] Michael Nguyen. Illustrated Guide to LSTM's and GRU's: A step by step explanation. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>, 2018.
- [22] Walter Pitts Warren S. McCulloch. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The University of Chicago Press*, 5, 1943.
- [23] Berkeley Edmund Callis. *Giant Brains Or Machines That Think* . 1949.
- [24] Claude E. Shannon. Programming a Computer for Playing Chess. *Philosophical Magazine*, 41, 1950.

- [25] Alan Mathison Turing. Computing Machinery and Intelligence. *Springer Science+Business Media B.V.* 2009, 1950.
- [26] Stanton G. Axline Bruce G. Buchanan C. Cordell Green Stanley N. Cohen Edward H.Shortliffe, Randall Davis. Computer-based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System. *Elsevier Inc*, 8, 1975.
- [27] Bhiksha Raj Haohan Wang. On the Origin of Deep Learning. <https://arxiv.org/abs/1702.07800>, 2017.
- [28] Necmettin Çarkacı. Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4>, 2018.
- [29] Jim Chappelow. Pareto Efficiency. <https://www.investopedia.com/terms/p/pareto-efficiency.asp>, 2019.
- [30] Keras LSTM Predicted Timeseries Squashed and Shifted. <https://stackoverflow.com/questions/48034625/keras-lstm-predicted-timeseries-squashed-and-shifted>, 2018.
- [31] LSTM NN Produces Shifted Forecast (Low Quality Result). <https://stackoverflow.com/questions/49697457/lstm-nn-produces-shifted-forecast-low-quality-result>, 2018.
- [32] Keras LSTM RNN Forecast - Shifting Fitted Forecast Backward. <https://stackoverflow.com/questions/39139446/keras-lstm-rnn-forecast-shifting-fitted-forecast-backward>, 2018.
- [33] Abhishek Jain Shalini Sivadasan Christapher P. Varghese Ram Kishore Agrawal Ravichandran Veerasamy, Harish Rajak. Validation of QSAR Models - Strategies and Importance. Vol: 2, 2011.
- [34] Thushan Ganegedara. Stock Market Predictions with LSTM in Python. <https://www.datacamp.com/community/tutorials/lstm-python-stock-market>, 2018.
- [35] Raoul Malm. NY Stock Price Prediction RNN LSTM GRU. <https://www.kaggle.com/raoulma/ny-stock-price-prediction-rnn-lstm-gru>, 2017.

# ÖZGEÇMİŞ

## Kişisel Bilgiler

Soyadı, adı : ORAL, Onur  
Uyruğu : TC  
Doğum tarihi ve yeri : 22/01/1992 - İzmir  
Medeni Hali : Bekar  
Tel : +90 507 855 87 37  
Fax : -  
E-mail : orallonur@gmail.com

## Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	: KTO Karatay Üniversitesi	-
Lisans	: KTO Karatay Üniversitesi	Haziran-2015
Lise	: Diltaş Anadolu Lisesi	Haziran-2010

## İş Deneyimi

Yıl	Yer	Görev
2016	Anadolu Birlik Holding	SAP-ABAP Geliştiricisi
2014	KOSKI IT	Stajyer Mühendis
2013	Türk Telekom NGN	Stajyer Mühendis

## Yabancı Dil

İngilizce (İyi), Çince (Başlangıç)

## Sertifikalar

- Katılım Belgesi: Third International Conference on Computational Science Engineering and Information Technology (CCSEIT 2013)
- KTO Karatay Üniversitesi Hazırlık Programı Başarı Sertifikası: KTO Karatay University English Preparatory Program Certificate of Appreciation for demonstrating excellent success in LET and quarterly assessment in English Prep Program (2011)



- European Portfolio Certificate for English Listening: B2 — Reading: B2 — Spoken Interaction: B2 Spoken Production: B2 — Writing: B2 — Linguistic Experience: Excellent (2006)

