



## Makine Öğrenmesi ve Derin Öğrenme Algoritmalarını Kullanarak Hisse Senedi Fiyat Tahmini

### Using Machine Learning and Deep Learning Algorithms for Stock Price Prediction

Seher Arslankaya<sup>1</sup>, Şevval Toprak<sup>1</sup>

<sup>1</sup>Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 54050, Sakarya, TÜRKİYE

Başyuru/Received: 20/07/2020

Kabul / Accepted: 10/12/2020

Çevrimiçi Basım / Published Online: 18/01/2021

Son Versiyon/Final Version: 18/01/2021

#### Öz

Günümüzde borsalar gözde bir dinamik yatırım aracıdır. Ancak bu dinamik yatırım aracı aynı zamanda ülkenin içinde bulunduğu durumdan, siyasetten, ülkeler arası izlenen politikalardan ve ülkenin sanayisinden büyük oranda etkilenmektedir. Bütün bu risklere karşın yatırımcı bu alanda yapmak isteyen yatırımcılar belli tahminlerde bulunmaktadır. Bu tahmin de geçmiş verilerden faydalanarak yapılmaktadır. Bunun nedeni olağanüstü durumlar dışında borsa hareketlerinin büyük oranda tahmin edilebilirliğindedir. Tahminlerin daha tutarlı olması için bazı tahmin sistemleri kurulabilir. Bu sistemlerde geleneksel tahmin yöntemlerinin yanı sıra yapay zeka yöntemleri de kullanılmaya başlanmıştır. Yapılan tahmin sistemlerinde yapay zeka yöntemlerinin geleneksel tahmin yöntemlerinden daha iyi sonuçlar verdiği gözlemlenmiştir. Bunun nedeni yapay zeka yöntemlerinin bir hafızası olmasıdır. Bu çalışmada makine öğrenmesi yöntemlerinden Polinom Regresyon ve Random Forest Regresyon ve derin öğrenme yöntemlerinden ise Tekrarlayan Sinir Ağları (RNN) ve Uzun-Kısa Süreli Bellek (LSTM) yöntemleri kullanılmıştır. Elde edilen sonuçlara göre en iyi sonucu Random Forest Regresyon modeli, en kötü sonucu ise Polinom Regresyon modeli vermiştir.

#### Anahtar Kelimeler

*"Hisse senedi fiyat tahmini, Polinom regresyon, Random forest regresyon, Tekrarlayan sinir ağı, Uzun-kısa süreli bellek"*

#### Abstract

Today, stock markets are a favorite dynamic investment method. However, this dynamic investment method is also greatly influenced by the situation in the country, politics, policies pursued across countries and the industry of the country. Despite all these risks, investors who want to invest in this area make certain predictions. This estimate is made by making use of historical data. The reason for this is that the stock market movements are highly predictable except for extraordinary situations. Prediction systems can be created to make the forecasts more consistent. These systems have started to use artificial intelligence methods as well as traditional forecasting methods. It has been observed that artificial intelligence methods give better results than traditional prediction methods. This is because it has a memory of artificial intelligence methods. In this study, Polynomial Regression and Random Forest Regression from machine learning methods and Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) from deep learning methods were used. According to the results obtained, the Random Forest Regression model was given the best result and the Polynomial Regression model was given the worst result.

#### Key Words

*"Stock price prediction, Polynomial regression, Random forest regression, Recurrent neural networks, Long-short term memory"*

## 1. Giriş

Günümüzde finansal zaman serileri ile ilgili doğru gelecek değer tahminleri yapabilmek yatırımcılar için son derece önemlidir. Özellikle borsalar bireysel ve kurumsal yatırımcılar için önemli bir kazanç kapısı halini almıştır. İnsanlar buradan kazanç elde etmek için belli riskleri alarak hisse senedi alım satımlarına karar verir ve kâr elde etmeye çalışırlar. Yatırım kararlarının en doğru şekilde verilmesi riski azaltacağından ve kâr elde etme olanağını artıracığından önemlidir. Kararları vermek için hisse senetlerinin geçmişteki değerlerinin durumuna bakılır ve bu değerlerden gelecek değerleri tahmin edilir. Burada geçmiş değerlerden faydalanılmasının nedeni borsa hareketlerinin olağanüstü durumlar dışında öngörülebilir olmasıdır. Hisse senedi fiyatlarının anlık olarak değişen yapısı nedeniyle fiyatlardaki değişimin tutarlı tahmin edilebilmesi için tahmin sistemleri geliştirilmesi yatırımcıların doğru kararlar verebilmelerini sağlayacaktır (Özer vd., 2017; Baek&Kim, 2018; Sakarya&Yılmaz, 2019).

Yatırımcılar hisse senedi yatırım kararlarını verirken iki tür analizden faydalanabilirler. Bunlardan biri hisse senetlerinin gerçek değerlerini, ülkenin politik durumunu, sanayinin ve ekonominin performans göstergelerini kapsayan temel analizdir. Diğeri ise geçmiş değerleri ve işlem hacimleri gibi piyasa hareketlerini baz alarak hisse senetlerinin değerlendirilmesi amacıyla kullanılan teknik analizdir. Teknik analizlerde amaç gerçek verilerin tahmini değildir. Bunun yerine gelecekte mümkün olabilecek davranış kalıpları ve eğilimleri hisse senedi grafikleri yardımıyla tahmin edebilmektir. Bunun nedeni hisse senetlerinin politik değişimler, yatırımcı hareketleri ve ekonominin genel durumları gibi birçok faktörden önemli derecede etkilenmesidir (Pabuçcu, 2019; Raşo&Demirci, 2019).

Bu tahmin sistemleri için daha önceden kullanılan geleneksel yöntemlerin dışında yapay zeka yöntemleri de kullanılmaya başlanmıştır ve yapay zeka yöntemlerinin kullanım oranı geleneksel yöntemlere göre çok daha fazladır. Yapay zekanın gelişmiş hesaplama yeteneklerine sahip olmasından dolayı doğrusal olmayan tahmin problemlerinde daha verimli sonuçlar verdiği kanıtlanmıştır. Bunun nedeni yapay zekanın aynı insan beyni gibi bir hafızasının olmasıdır. Başka bir deyişle geçmiş verilerden tahmin için kullanılacak bilgilerin seçimini yapabilme yeteneğinin var olmasıdır. Yapay zekanın içindeki yöntemler hafıza yapıları, kullanılacak katman sayıları ve kullanıldığı alanlar olarak farklılık gösterir (Kutlu&Badur, 2009; Vijh vd., 2020).

Bu çalışmada Ereğli Demir ve Çelik Fabrikaları T. A. Ş.'nin 01.01.2014-01.06.2020 tarihleri arasındaki günler için hisse senedi fiyatlarının kapanış değerlerinden yararlanarak tahmin işlemi yapılmıştır. Bunun için makine öğrenmesi yöntemlerinden Polinom Regresyon modeli ve Random Forest Regresyon modeli, derin öğrenme yönteminden ise RNN ve LSTM yöntemleri Python 3.7 programlama dili yardımıyla Spyder (Anaconda3) editöründe uygulanmıştır. Kullanılan bu 4 yöntemin sonuçlarına göre aralarından hisse senedi fiyat tahmininde kullanılacak en iyi yöntemin bulunması amaçlanmıştır.

Çalışmanın bu aşamadan sonraki bölümleri; benzer çalışmalardan örneklerin verildiği ikinci bölüm, çalışmada kullanılan veri seti ve çözüm yöntemlerinin anlatıldığı üçüncü bölüm, uygulamanın verildiği dördüncü bölüm, uygulamanın istatistik analizinin yapıldığı beşinci bölüm ve sonuç ve bulguların verildiği altıncı bölümden oluşmaktadır.

## 2. Literatür Taraması

Borsa hareketleri için yatırımcıların ihtiyaç duyduğu tahmin yöntemleri olarak kullanılan geleneksel yöntemlerden ve yapay zeka modellerini içeren çalışmalardan bu bölümde bahsedilmiştir. Çalışmalara bakıldığında yapay zeka yöntemlerinin geleneksel yöntemlerden daha doğru sonuçlara ulaştırdığı görülmüştür.

Bahadır (2008) borsa gelecek değer tahmin çalışmasında Bayes teoremini ve yapay sinir ağlarını (YSA) kullanmıştır. Çalışmada New York Borsası (NYSE) bünyesinde bulunan ETF'ler incelenmiştir. Çalışmayı gerçekleştirmek için iki yöntem kullanılmıştır. Birinci yöntem belirli bir uzmanlık alanındaki kişilerden derlenen bilgilerle Bayes karar modeli üzerine bir uzman sistem modeli oluşturmaktır. Teknik analiz yöntemleriyle veriler test edilmiş ve en iyi sonucu bulmak için kurallar belirlenmiştir. İkinci yöntem olan YSA modelinde ise NeuroDimension firmasının NeuroSolutions programı birden fazla ağla ve birçok parametre ile çözümü gerçekleştirip en iyi sonucu bulanı tespit etmek amaçlanmıştır. Sonucunda yapay sinir ağlarının uzman sisteme göre daha başarılı sonuçlar verdiği gözlemlenmiştir. Kutlu & Badur (2009) tarafından İstanbul Menkul Kıymetler Borsası (İMKB) 100 Endeksi için yapılan çalışmada YSA kullanılmıştır. Test sonuçlarından elde edilen bilgiler basit hareketli ortalamalar yöntemiyle kıyaslanmıştır. Çalışmanın sonucunda elde edilen kurallara göre YSA'nın bu konuda ümit verici olduğunu söylemişlerdir. Moralı (2011) İMKB 100 Endeksi ile altın fiyatları, faiz oranı, bankalar arası çift taraflı işlem miktarı ve USD günlük kapanış değerleri gibi piyasa değişkenlerini kullanarak modellemeler yapmıştır. Bu tahmin modellerini zaman serileri için Bütünleşik Otoregresif Hareketli Ortalama (Autoregressive Moving Average Model-ARMA), nümerik arama modelleri için Newton yöntemini ve YSA yöntemlerinden de Geri Yayılım algoritmasını kullanarak gerçekleştirmiştir. Bu üç yöntemle yapılan tahminlerin hata değerleri bulunmuş ve korelasyon analizleri yapılmıştır. Bunların sonucunda hata değeri en az ve korelasyon değeri en yüksek çıkan tahmin yöntemi YSA olmuştur. Çalışkan & Deniz (2015) tarafından Borsa İstanbul (BIST) 30'daki hisse senetleri için YSA ile günlük tahmin yapılmıştır. Yapılan tahminlerin Ortalama Mutlak Hatası (MAE) ve Ortalama Mutlak Yüzde Hatası (MAPE) bulunmuştur. Tahminlerin başarı oranı %58 olarak bulunup YSA'nın genel anlamda BIST 30 tahminlerinde bulunabileceği sonucu çıkartılmıştır. Özer vd. (2018) gelişmekte olan 4 ve gelişmiş olan 4 olmak üzere 8 ülkenin hisse senetlerinin haftalık kapanış değerlerini kullanarak bulanık mantık ve YSA yöntemleriyle tahminde bulunup aralarından

en iyi sonucu veren yöntemi bulmayı amaçlamışlardır. Bu çalışmada iki yöntemin de yakın sonuçlar verdiği gözlemlenmiştir. Dayı (2018) İMKB’de işlem gören imalat şirketlerinin 1991-2010 yılları arasındaki hisse senedi verilerini kullanarak statik ve dinamik YSA yöntemleriyle tahminler yapmış ve gerçek değerleriyle karşılaştırmıştır. İki YSA türünün de kabul edilebilir başarıda sonuçlar verdiği ancak statik YSA’nın dinamik YSA’dan daha iyi sonuçlar verdiği gözlemlenmiştir. Hiransha vd. (2018) çalışmalarında Çok Katmanlı Algılayıcılar (Multilayer Perceptron-MLP), RNN, LSTM ve Evrimsel Sinir Ağları (Convolutional Neural Network-CNN) yöntemlerini kullanmışlardır. Hindistan Ulusal Borsası (NSE) ve NYSE’nin günlük kapanış fiyatları tahmin edilmiştir. Tahmin sonuçları karşılaştırıldığında CNN’in diğer yöntemlere göre daha iyi bir sonuç verdiği görülmüştür. Baek & Kim (2018) geçmiş verilerle tahmin verileri arasındaki aşırı uyumu önlemek ve değer tahmini yapabilmek için iki modüllü bir LSTM yaklaşımı sunmuşlardır. Veri olarak S&P 500 (ABD Borsa Endeksi) ve Kore Birleşik Hisse Senedi Fiyat Endeksi 200 (KOSPI 200) kullanılmıştır. Çalışmanın sonucunda yeni modelin tahmin sonuçlarının çok iyi bir düzeyde olduğunu ortaya koymuşlardır. Kim & Won (2018) Otoresif Koşullu Değişken varyans (GARCH) ve LSTM yöntemlerini birleştirerek yeni bir yaklaşım sunmuşlardır. Oluşturdukları yeni yaklaşımı kullanarak KOSPI 200 endeks verilerinin üzerinde uygulama yapmışlardır. Bunun yanında GARCH, üstel GARCH, Üstel Ağırlıklı Hareketli Ortalama, Derin İleri Beslemeli Sinir Ağı (Deep Feedforward Network-DFN) ve LSTM yöntemlerini de uygulayıp analizlerini yapıp mevcut metodolojiyle yeni yaklaşımlarını karşılaştırmışlardır. Sonucunda ortaya koydukları yeni yaklaşımın iyi bir sonuç verdiği gözlemlenmiştir. Sakarya & Yılmaz (2019) BIST 30 endeksinin tahmini için belli değişkenler kullanarak bu çalışmayı gerçekleştirmiştir. Verilere, verilerdeki gürültüleri temizlemek için öncelikle Dalgacık Dönüşümü uygulanmıştır. Bu işlemin çıktılarına ve değişkenlere denetimsiz eğitilmiş yığın otokodlayıcılar uygulanmıştır. Bu aşamanın çıktılarına ise LSTM uygulanmıştır. Çalışma sonucunda tahminlerin %80 oranında doğru sonuç verdiği gözlemlenmiştir. Pabuççu (2019) çalışmasında BIST 100 endeksinin tahmini için YSA, Destek Vektör Makineleri (Support Vector Machine-SVM) ve Navie Bayes olmak üzere üç farklı makine öğrenmesi yöntemi kullanmıştır. Çalışma sonunda bu üç yöntemin performansları değerlendirilmiş ve üçünün de sonucu birbirine çok yakın olmakla birlikte en iyi sonucu YSA modelinin verdiği gözlemlenmiştir. Kantar (2019) BIST 100 endeksinin 2009-2019 tarihleri arasındaki aylık verilerini kullanarak ARMA ve YSA ile tahmin yapmıştır. Burada bağımlı değişken olarak BIST100 ve bağımsız değişkenler ise makroekonomik değişkenler olarak belirlenmiştir. Çalışma sonucunda YSA modelinin ARMA modelinden çok daha iyi bir sonuç verdiği ortaya konulmuştur. Akşehir & Kılıç (2019) Çoklu Lineer Regresyon (Multiple Linear Regression-MLR), Karar Ağaçları (Decision Tree) ve Random Forest yöntemlerini kullanarak banka hisse senetlerinin fiyatları ile ilgili tahmin çalışması yapmışlardır. Çalışma sonucunda üç yöntemin de sonuçları birbirine yakın ve kullanılabilir durumda çıkmıştır. Çınaroğlu & Avcı (2020) Türk Hava Yolları şirketinin 2015-2018 yılları arasındaki BIST 100 ve BIST Ulaştırma endekslerini, petrol ve dolar fiyatlarını girdi olarak kullanarak günlük hisse senedi değerlerini YSA ile tahmin etmişlerdir. Çalışmanın sonunda YSA tahminlerinin gerçek değerlere çok yakın olduğu görülmüştür. Alhnaity & Abbod (2020) çalışmalarında FTSE 100 (Londra Borsa endeksi), S&P 500 ve Nikkei 225 (Tokyo Mekul Kıymetler Borsası endeksi) endekslerinin gelecek gün borsa kapanış değerlerini bulmayı amaçlamışlardır. Destek Vektör Regresyon (Support Vector Regression-SVR), Ensemble Ampirik Mod Ayırıştırma (Ensemble Empirical Mode Decomposition-EEMD), Genetik Algoritma (Genetic Algorithm-GA), RNN, Geriye Yayılımlı Sinir Ağları (Backpropagation Neural Network-BPNN) ve Ağırlık Ortalaması (Weight Average-WA) yöntemlerine dayalı üç aşamalı yeni bir hibrit model geliştirmişlerdir. Yeni yaklaşımı oluşturan her yöntemle ayrı ayrı tahmin işlemi yapıp yeni yaklaşımın sonuçlarıyla karşılaştırılmıştır. Yeni yaklaşımın sonuçlarının diğer yöntemlerin sonuçlarına göre daha kesin sonuçlar verdiği gözlemlenmiştir. Vijh vd. (2020) birbirinden farklı sektörlerde faaliyet gösteren beş şirketin bir sonraki gün borsa kapanış fiyatını tahmin etmek için YSA ve Random Forest yöntemlerini kullanmışlardır. Girdi olarak borsa açılış fiyat değerleri, gün içindeki en yüksek değerler, gün içindeki en düşük değerler ve borsa kapanış değerleri kullanılmıştır. Tahminlerin doğruluğunu analiz etmek için Ortalama Karekök Hata (RMSE) ve MAPE metrikleri kullanılmıştır. Metrik hesaplama sonuçlarının düşük değerler olduğu, bu nedenle de YSA ve Random Forest yöntemlerinin hisse senedi kapanış fiyatlarını tahmin etmeye uygun yöntemler olduğu görülmüştür.

### 3. Materyal ve Metot

Bu bölümde çalışma içerisinde kullanılan veri seti, yöntemler ve bu yöntemleri uygulamak için Python 3.7 programlama dilinde yararlanılan kütüphanelerin/yardımcı paketlerin uygulamada kullanım aşamaları anlatılmıştır. Bu çalışmada Python 3.7 programlama dilinin kullanılmasının nedeni, bu programın içinde bulunan kütüphaneler/yardımcı paketler ile yapay zeka problemlerinin çözümü için gerekli adımların uygulanmasında kolaylık sağlamasıdır. Veriler üzerindeki işlemlerin ve kontrol edilebilirliğin kolay olması nedeniyle veritabanı olarak Excel 2013 kullanılmıştır.

#### 3.1. Veri Seti

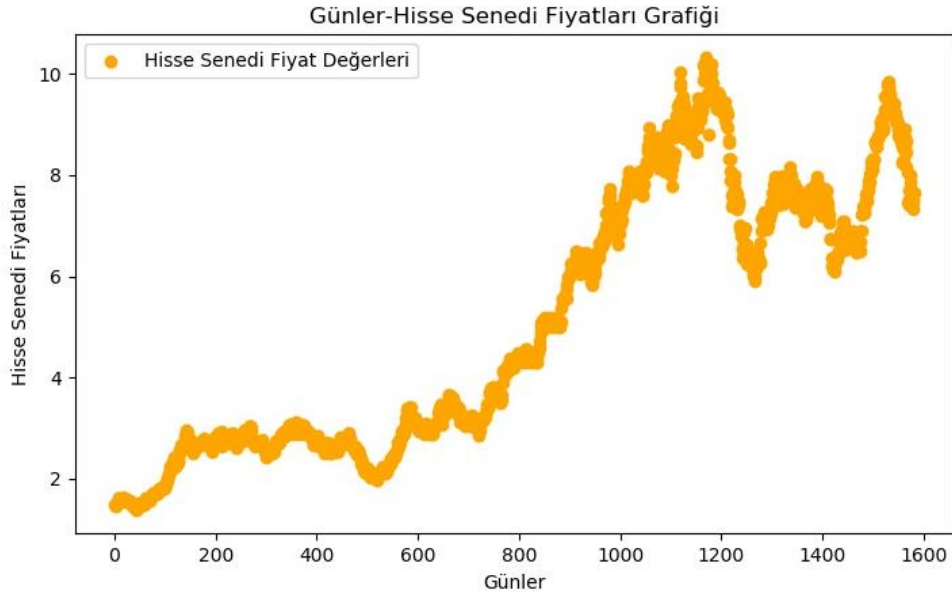
Bu çalışmada kullanılan veri seti bağımsız değişken olarak günlerden ve bağımlı değişken olarak Ereğli Demir ve Çelik Fabrikaları T.A.Ş.’nin 01.01.2014 - 01.06.2020 tarihleri arasındaki hisse senedi fiyatlarının borsa kapanış değerlerinden oluşmaktadır. Veriler <https://tr.investing.com/> adresinden alınmıştır. Toplam veri sayısı belirtilen tarihler arasındaki hafta sonlarında ve diğer tatil günlerinde borsanın kapalı olması nedeniyle 1619 adettir. Veri setinden ilk 5 ve son 5 veri olmak üzere toplam 10 veri Tablo 1’de örnek olarak gösterilmiştir. Tablo 2’de ise veri setindeki değişkenler ve bunların aralıkları verilmiştir. Veriler Şekil 1’de grafik olarak görselleştirilmiştir.

**Tablo 1.** Veri Setinden Örnek Değerler

Günler	Hisse Senedi Kapanış Fiyatları
1	1.48
2	1.45
3	1.47
4	1.5
5	1.51
...	...
1615	8.06
1616	8.15
1617	8.11
1618	8.05
1619	8.13

**Tablo 2.** Veri Setindeki Değişkenlerin Değer Aralıkları

Veri Seti Değişkenleri	Aralıklar	
	Alt Değer	Üst Değer
Günler	1	1619
Hisse Senedi Kapanış Fiyatları	1.37	10.32

**Şekil 1.** Günler-Hisse Senedi Fiyatları Grafiği

### 3.2. Makine Öğrenmesi

Makine öğrenmesi, matematik ve istatistik biliminden yararlanarak verilerin üzerinde yapılan işlemlerden çıkarımlarda bulunarak tahminler yapan sistemlerin bilgisayarlarla modellenmesidir. Model, mevcut veri seti ve kullanılan algoritmayla oluşturulur. Makine öğrenmesi, modellerden maksimum performansı elde etmek üzere kullanılır ve öğrenme türleri üç gruba ayrılır. Bunlar:

- Gözetimli Öğrenme (Supervised Learning) : Bu öğrenme şeklinde girdiler ve çıktılar arasında eşleme yapan fonksiyon regresyon veya sınıflandırma yöntemleriyle oluşturulur. Eğitim verileri hem girdileri hem de çıktıları içerir.
- Gözetimsiz Öğrenme (Unsupervised Learning) : Bu öğrenme yönteminde bilinmeyen bir yapı tahmin edilir. Giriş verisinin sınıfı belirsizdir.

- Takviyeli Öğreme (Reinforcement Learning) : Bu öğrenme şekli girdilerin çevreleriyle etkileşerek ödül denilen geri bildirimlerini maksimum seviyeye çıkarıp en uygun hareket tarzını bulmayı amaçlar bu yönüyle diğer öğrenim türlerinden farklıdır (Şafak, 2017).

Bu çalışmada gözetimli öğrenme yöntemlerinden Polinom Regresyon modeli ve Random Forest Regresyon modeli veriler üzerinde uygulanmıştır.

### 3.2.1. Polinom regresyon modeli

Gerçek hayat problemlerinde lineer ve çoklu lineer regresyonda olduğu gibi bağımlı ve bağımsız değişkenler arasında doğrusal bir ilişki olmayabilir. Böyle durumlarda lineer regresyon çeşitleri gibi doğrusal olmayan ve derecesinde değişiklikler yapılarak verilere en uygun hale getirilebilecek regresyon modellerinden birisi Polinom Regresyon modelidir. Şekil 1’de verilen grafikte verilerin eğrisel olarak değiştiği gözlemlenmiştir. Bu nedenle çalışmada kullanılan veriler için Polinom Regresyon modelinin uygun olduğu görülmüştür.

Polinom Regresyonla tahmin yöntemi Python 3.7 programlama dilinde Pandas, Numpy, SciKit-Learn, ve Matplotlib kütüphanelerini kullanarak gerçekleştirilmiştir. Adımları şu şekildedir:

- Adım 1-Verilerin okunması: Veritabanında bulunan veriler Pandas kütüphanesi yardımıyla okunup bir değişkene atanır.
- Adım 2- Verilerin düzenlenmesi: Veri setinin içinden kullanılacak özellikler sütun olarak değişkenlere atanır ve oluşturulmuş her değişken Numpy kütüphanesi ile iki boyutlu hale getirilir. Gözlem verileri Matplotlib kütüphanesindeki grafik özelliği kullanılarak bir grafik üzerinde gösterilir.
- Adım 3- Verilerin ön hazırlığının yapılması: SciKit-Learn kütüphanesinden kullanılan Preprocessing modülüyle tahmini yapılacak verilerden bağımsız değişkene polinom özellikleri kazandırılır. Burada yeni yapılandırılacak polinom değerlerinin kaçınıcı dereceden bir polinom olacağına bilgisi verilir.
- Adım 4- Tahmin işleminin yapılması: SciKit-Learn kütüphanesinden tahmin için Linear\_model modülünden LinearRegression sınıfı eklenir. Tahmin işlemi bağımsız değişkenin ön hazırlıktan geçmiş haliyle lineer regresyon modeliyle yapılır. Tahmin değerlerinin oluşturduğu fonksiyon gözlem değerlerinin bulunduğu grafiğe Matplotlib kütüphanesi yardımıyla yerleştirilir.

### 3.2.2. Random forest regresyon modeli

Random Forest modeli hem sınıflandırma hem de regresyon problemlerinde kullanılabilen bir makine öğrenmesi yöntemidir. Random Forest Regresyon modelinin çalışma mantığı karar ağaçları yapısına benzer. Veri seti rassal olarak küçük parçalara ayrılıp karar ağaçlarını oluşturur. Tahminleme aşamasında ise veri setinden oluşmuş karar ağaçlarının tahminlerinin ortalaması alınır. Bu çalışmada Random Forest Regresyon modelinin kullanılmasının nedeni, bu modelin eğitim sırasında kullanılan geçmiş verileri ezberleme (overfitting) sorununun büyük oranda önüne geçmiş olması, veri türünün zaman serisi şeklinde olması ve başarılı sonuçlar veren bir yöntem olmasıdır (Akşehir&Kılıç, 2019).

Random forest regresyonla tahmin yöntemi Python 3.7 programlama dilinde Pandas, Numpy, SciKit-Learn ve Matplotlib kütüphanelerini kullanarak gerçekleştirilmiştir. Adımları şu şekildedir:

- Adım 1-Verilerin okunması: Veritabanında bulunan veriler Pandas kütüphanesi yardımıyla okunup bir değişkene atanır.
- Adım 2- Verilerin düzenlenmesi: Veri setinin içinden kullanılacak özellikler sütun olarak değişkenlere atanır ve oluşturulmuş her değişken Numpy kütüphanesi ile iki boyutlu hale getirilir. Gözlem verileri Matplotlib kütüphanesindeki grafik özelliği kullanılarak bir grafik üzerinde gösterilir.
- Adım 3- Random Forest algoritmasının oluşturulması: SciKit-Learn kütüphanesinin farklı ya da aynı algoritmaları tekrar tekrar gerçekleştiren Ensemble modellerinden RandomForestRegressor sınıfı seçilir. Burada oluşturulması istenilen ağaç sayısı ve algoritmanın aynı değerlerle her çalıştığında farklı sonuçlar ortaya çıkarmasını engelleyen bir rastgelelik durum değişkeni verilir. Bu sınıf bir değişkene atanır. Bağımlı ve bağımsız değişkenler sınıfın atandığı değişkene fit edilir.
- Adım 4- Tahmin işleminin yapılması ve grafik üzerinde gösterilmesi: Tahmin işlemi RandomForestRegressor sınıfının atandığı değişkendeki yatay ekseninde yer alan günlere göre tahmin edilir. Gözlem değerleri ve tahmin değerleri Matplotlib kütüphanesi yardımıyla grafikte gösterilir.

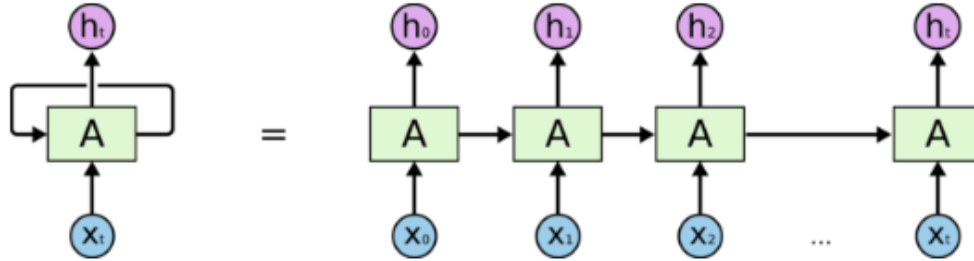
### 3.3. Derin Öğrenme

Derin öğrenme, sinir ağları algoritmalarından diğerlerine göre daha yeni olan bir türdür ve derin sinir ağlarını eğitir. Makine öğrenmesinin alt dallarından biridir. Bir ya da daha fazla girdisi olan, içerisinde çok sayıda katman bulunan ve en son bir ya da daha fazla çıktısı olan bir algoritmadır. Her katmanda kendisinden önceki bilgileri birleştirir ve bunlardan kompleks ve anlamlı sonuçları olan değerler üretir. Bu yönüyle diğer sinir ağları algoritmalarından daha tutarlı ve güçlüdür. Derin öğrenme de makine öğrenmesinin bir alt dalı olduğundan aynı onun gibi denetimli ve denetimsiz öğrenme modellerine sahiptir. Derin öğrenme modelleri, kaynaklardaki verilerin özelliklerini göz önüne alarak ve bunları gizli katmanlarında öğrenerek farklı veri dönüşüm aşamalarından meydana gelmektedir (Sakarya&Yılmaz, 2019).

Derin öğrenme, çok sayıda sahip olduğu gizli katmanla gelişmiş bir sinir ağıdır ve görüntü işleme, doğal dil işleme, el yazısı tanıma, çoklu sınıflandırma, regresyon problemleri, ses tanıma, zaman serisi tahminleri gibi birçok alanda etkin bir şekilde kullanılmaktadır.

### 3.3.1. Tekrarlayan sinir ağı

RNN dışarıdan gelen yeni bilginin yanında bir önceki birimin çıktısını bir sonraki birimde girdi olarak kabul edip bir döngü oluşturan bir yapay sinir ağı sınıfıdır. Bu sınıftaki ana düşünce bilgileri sıralı olarak kullanmaktır. RNN modellerinde alt katmanlardan ve bir önceki adımın çıktılarının girdi olarak kullanılmasıyla aktivasyon değerlerini hesaplayan gizli katmanlar kullanılır. Bu kendini yineleme özelliğini mevcut çıktılarla ilgili kararları yeni verilerle geçmişteki verileri birleştirerek sağlar. Çıktı üretildikten sonra kopyalanıp tekrar RNN'in giriş kısmına iletilir. Bu bilgi yeni girdilerde kullanılır. RNN yapıları yeni girdilerle işlem yapabilmek için hafızalarını kullanırlar ancak bu hafıza kısa süreli bir hafızadır (Hiransha vd., 2018; Mittal, 2019; Sakarya&Yılmaz, 2019).



Şekil 2. RNN'nin Genel Yapısı (Mittal, 2019)

Şekil 2'de RNN'in genel yapısı gösterilmiştir. Öncelikle ağı bir  $X_0$  girdisi girer ve işlemler sonucunda  $h_0$  olarak çıkar. Bir sonraki adımda girdi olarak yeni bir bilgi olan  $X_1$  ve bir önceki adımın çıktısı olan  $h_0$  girer. Yeni adımda ise yeni girdi olarak  $X_2$  ve bir önceki adımın çıktısı olan  $h_1$  girer. Bu işlemler bu şekilde kendini yineler (Mittal, 2019).

RNN ile tahmin yöntemi Python 3.7 programlama dilinde Pandas, Numpy, SciKit-Learn, Matplotlib, Pydot, Graphviz ve Keras kütüphanelerini kullanarak gerçekleştirilmiştir. Adımlar aşağıda verilmiştir.

- Adım 1-Verilerin okunması: Veritabanında bulunan veriler Pandas kütüphanesi yardımıyla okunup bir değişkene atanır.  $X$  bir veri seti olmak üzere bu veri setinin her bir elemanı  $x_i$  olsun ve bu  $X$  veri seti bir  $A$  kümesinde gösterilsin.  
 $A = \{x_i \in X\} \quad i=1,2,\dots, n$
- Adım 2-Verilerin Düzenlenmesi: Veri setinin içinden kullanılacak özellikler sütun olarak değişkenlere atanır ve oluşturulmuş her değişken Numpy kütüphanesi ile iki boyutlu hale getirilir. Gözlem verileri bir grafik üzerinde gösterilir.
- Adım 3-Verilerin ön hazırlığının yapılması: SciKit- Learn kütüphanesinden Preprocessing modülünden veri setini normalize etmek için MinMaxScaler sınıfı eklenir. Bu sınıf bir değişkene atanıp normalize işlemiyle verilerin hangi değerler arasına gireceği belirtilir.
- Adım 4- Eğitim kümesinin ve test kümesinin oluşturulması: Eğitim kümesi geçmiş verilerin öğrenildiği yani ağın eğitilmesi için seçilen verilerin bulunduğu kümedir. %  $t$  tüm veriler içinden eğitim kümesini oluşturan kısmın değeri olsun. Eğitim kümesinin veri sayısı Denklem 1'de gösterilmiştir.

$$EKVS = n * t/100 \quad (1)$$

Test kümesi, eğitim kümesinin dışındaki verileri kapsayan kümedir. Denklem 2'de verilmiştir.

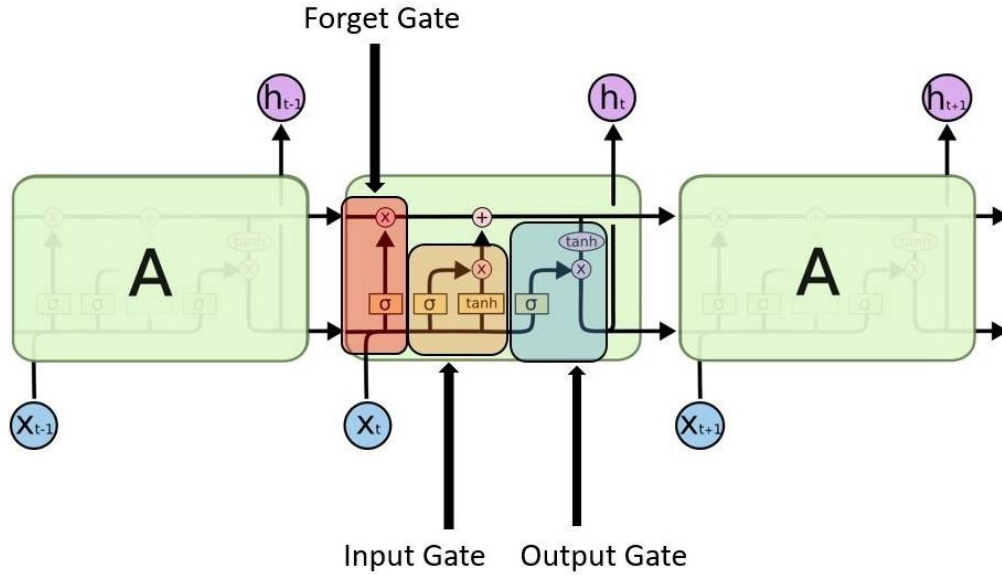
$$TKVS = n - EKVS \quad (2)$$

- Adım 5- Eğitim ve test setinin tahmin metodunun belirlenmesi: Eğitim seti iki farklı dizi oluşturacak şekilde ikiye ayrılır. Tahmin için sabit bir adım değeri belirlenir. Bu sabit değer kadar veri birinci diziye atanır. Bir sonraki veri ise tahmin edilip diğer diziye aktarılır. Bu şekilde bütün eğitim verileri tahmin edilir. Aynı işlemler test verisi için de gerçekleştirilir.
- Adım 6- RNN yapısının oluşturulması ve görselleştirilmesi: Derin öğrenme yapısının oluşturulabilmesi için Keras Kütüphanesinden model türlerinden verileri zaman serisi olduğu için Sequential sınıfı ve SimpleRNN, Dense ve Dropout katman türleri eklenir. Bu sınıf bir değişkene atanır ve sınıfın içinde yapıda kullanılacak katmanlar, bu katmanların sayısı, katmanlardaki blok sayısı ve özellikleri verilir. Oluşturulan ağ yapısı Pydot ve Graphviz yardımcı paketleri yardımıyla görselleştirilir.
- Adım 7- Tahmin işleminin gerçekleştirilmesi: Eğitim ve test verileri Sequential sınıfının atandığı değişkenle tahmin işlemini gerçekleştirir. Tahmin işlemlerinden sonra veriler normalize halde bulduklarından normal yapılarına geri döndürülüp gözlem değerleri, eğitim tahmin değerleri ve test tahmin değerleri Matplotlib kütüphanesi yardımıyla grafikte gösterilir.

### 3.3.2. Uzun-kısa süreli bellek

Tekrarlayan sinir ağlarının bir alt dalı olan LSTM, RNN'deki mevcut sorunları ortadan kaldırma amacıyla geliştirilmiştir. Bir LSTM yapısında giriş, çıkış ve unut olmak üzere üç kapı bulunmaktadır. Bu kapıların görevleri sırasıyla yazma, okuma ve sıfırlamadır. Hücre durumlarında gerçekleşen değişiklikler anlatılan üç kapı tarafından kontrol edilmektedir. Hücelere iletilecek bilgilerin seçimi, çıktı olarak bu bilgilerin ne kadarının nasıl kullanılacağı gibi seçimlerin gerçekleşmesini sağlayan kapı mekanizmalarını da sisteme ekleyerek RNN'deki mevcut sorunların çözülmesi sağlanmıştır. LSTM kapılarından giriş kapısının görevi belleğe eklenecek bilgiler kontrol etmek, unut kapısının görevi eski bilgilerin ne kadarının yeni verilere taşınacağını kontrol etmek, çıkış kapısı ise bellekte bulunan bilgilerin ne kadarının çıktı aşamasında kullanılacağını kontrol etmektir (Sakarya&Yılmaz, 2019).

Şekil 3'te LSTM hücresinin yapısı ve kapıları verilmiştir. Burada  $t$  anındaki girdi verisi  $x_t$  ve bir önceki hücreden gelen çıktı verisi ( $t$  anındaki hücre için girdi)  $h_t$ 'dir. Öncelikle  $t$  anındaki girdi olan  $x_t$  ve bir önceki adımın çıktısı olan  $h_{t-1}$  unut kapısına gelir. Bu değerlere göre unut kapısı  $x_t$  girdisine ve  $h_{t-1}$  çıktısına göre bir karar verir. Bir önceki hücreden gelen bilgi miktarı durum bilgisi mevcut  $t$  anındaki hücrede kontrol edilir. Girdi kapısında, bellek hücresine bir önceki hücreden gelen  $h_{t-1}$  ve  $x_t$  olarak yeni gelen bilginin ne kadarının kullanılacağına karar verilir. Bu bilgilerin kullanılıp kullanılmayacağı 0 ve 1 çıktılarında anlaşılır. Karar 0 ise bilgileri kullanılmaz, 1 ise bilgiler kullanılır. Çıktı kapısı ise bir çıktı olup olmayacağına karar verir. Çıktı kapısı da  $h_{t-1}$  ve  $x_t$  girdilerini alır ve girdi kapısında olduğu gibi 0 veya 1 şeklinde bir çıktının olup olmayacağına karar verir.  $t$  anındaki hücre durumu bir sonraki hücreye geçerken ( $t+1$  anı)  $t-1$  ve  $t$  anlarındaki bilgilerin toplamı olmaktadır (Kim&Won, 2018; Sakarya&Yılmaz, 2019).



Şekil 3. LSTM Hücre Yapısı ve Kapıları (Mittal, 2019)

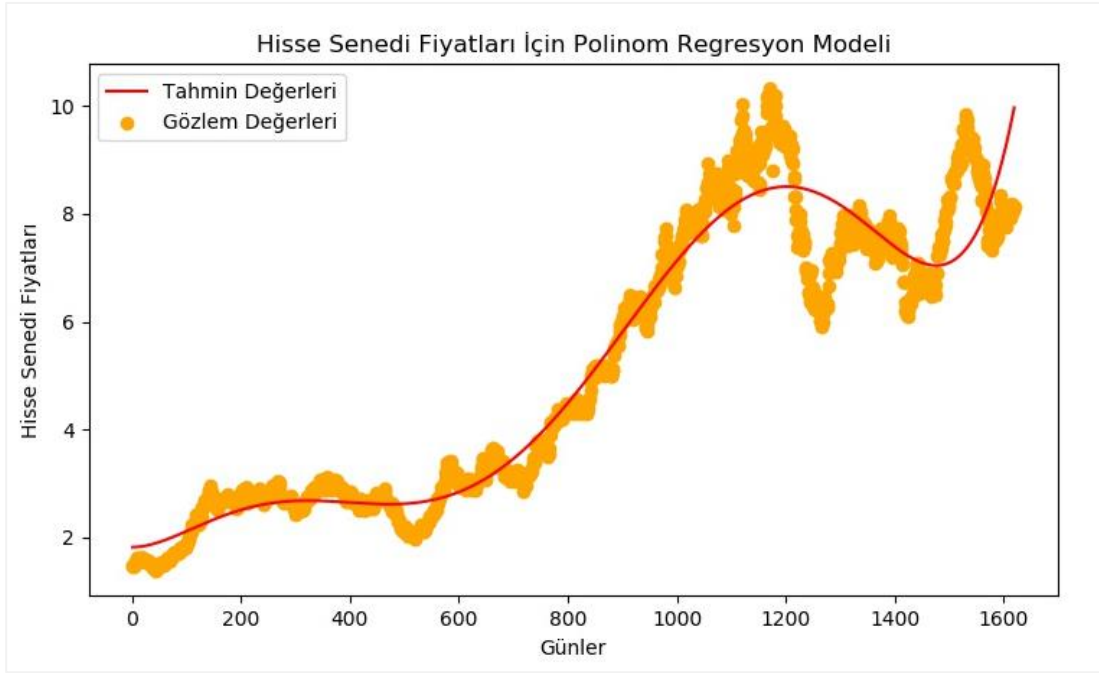
LSTM ile tahmin yöntemi Python 3.7 programlama dilinde Pandas, Numpy, SciKit-Learn, Matplotlib, Pydot, Graphviz ve Keras kütüphanelerini kullanarak gerçekleştirilmiştir. Bütün adımları RNN çözüm yönteminin adımlarıyla aynıdır. Tek farkı RNN ağı oluşturma adımında eklenen katmanlardan SimpleRNN'in yerine LSTM ağı oluşturulduğundan LSTM katmanının seçilmesidir.

## 4. Uygulama

Bu çalışmanın veri seti, günler ve Ereğli Demir ve Çelik Fabrikaları T. A. Ş.'nin hisse senedi fiyatlarının borsa kapanış değerlerinden oluşmaktadır. Çözüm yöntemi olarak makine öğrenme algoritmalarından Polinom Regresyon ve Random Forest Regresyon ve derin öğrenme algoritmalarından RNN ve LSTM yöntemleri kullanılmıştır. Amaç bu dört farklı tahmin yönteminden hangisinin hisse senedi fiyat tahmini konusunda daha iyi sonuçlar verdiğini bulmaktır.

### 4.1. Polinom Regresyon Modeli

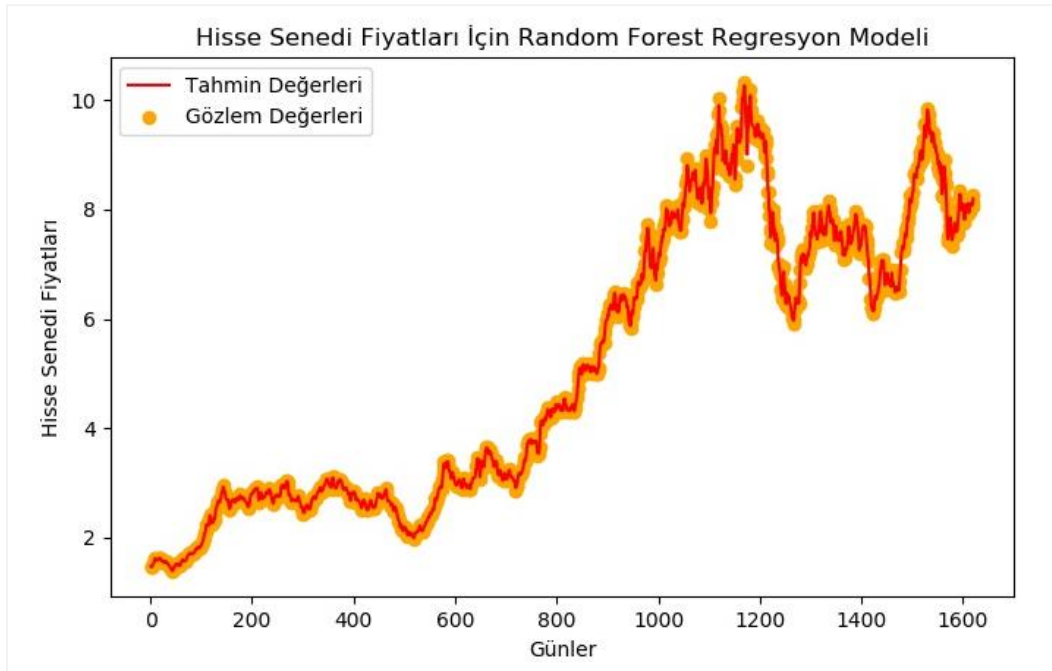
Verilere en uygun polinom fonksiyonunun derecesini bulmak için çok sayıda deneme yapılmıştır. Oluşturulan tahmin fonksiyonlarından gözlem değerlerine en uygun polinomun 6. dereceden bir polinom olduğu görülmüştür. Şekil 4'te Polinom Regresyon için gözlem ve tahmin değerlerinin grafik olarak görselleştirilmiş hali verilmiştir.



Şekil 4. Polinom Regresyon Modelinin Hisse Senedi Fiyat Tahmin Grafiği

#### 4.2. Random Forest Regresyon Modeli

SCiKit-Learn kütüphanesinin tekrarlanan algoritmalar modülünden çağırılan RandomForestRegressor sınıfına belirlenen karar ağacı sayısı 100 ve rassallık durumu 0 olarak belirlenmiştir. Karar ağacı sayısının seçiminde farklı sayılarda karar ağaçları oluşturulup denenmiştir. Karar ağacı sayısı azaldıkça tahmin değerlerinin gözlem değerlerinden uzaklaştığı, daha fazla sayıda karar ağacı oluşturulduğunda işlem süresinin uzadığı ve hata değerlerinde anlamlı bir fark olmadığı görülmüştür. Şekil 5'te Random Forest Regresyon için gözlem ve tahmin değerlerinin grafik olarak görselleştirilmiş hali verilmiştir.

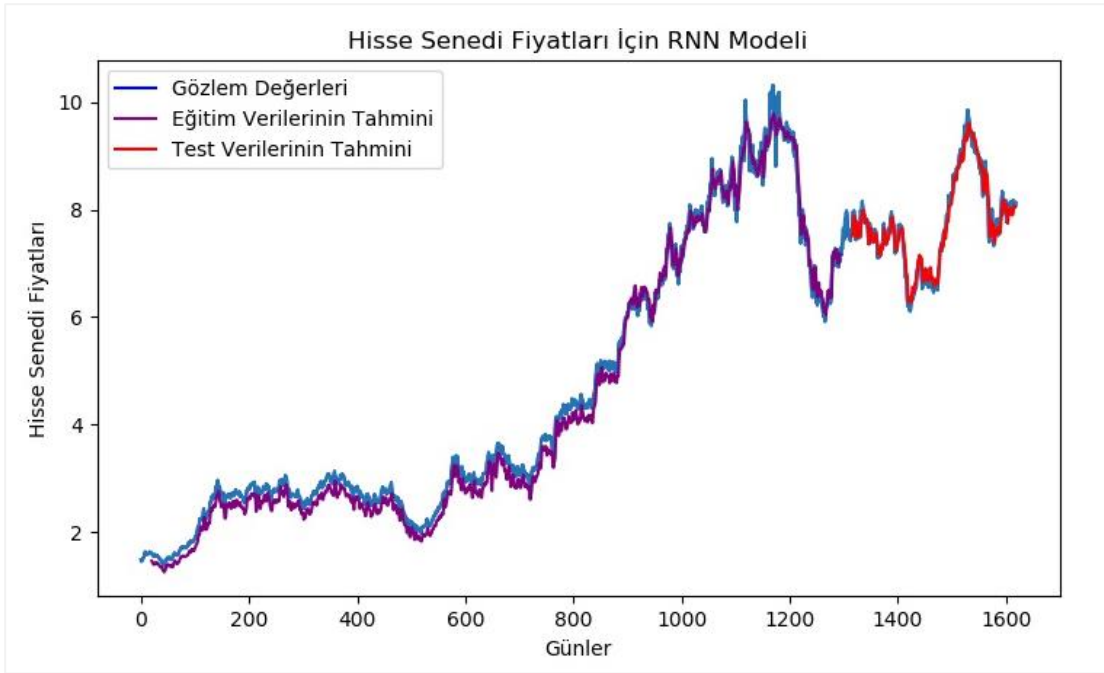


Şekil 5. Random Forest Regresyon Modelinin Hisse Senedi Fiyat Tahmin Grafiği



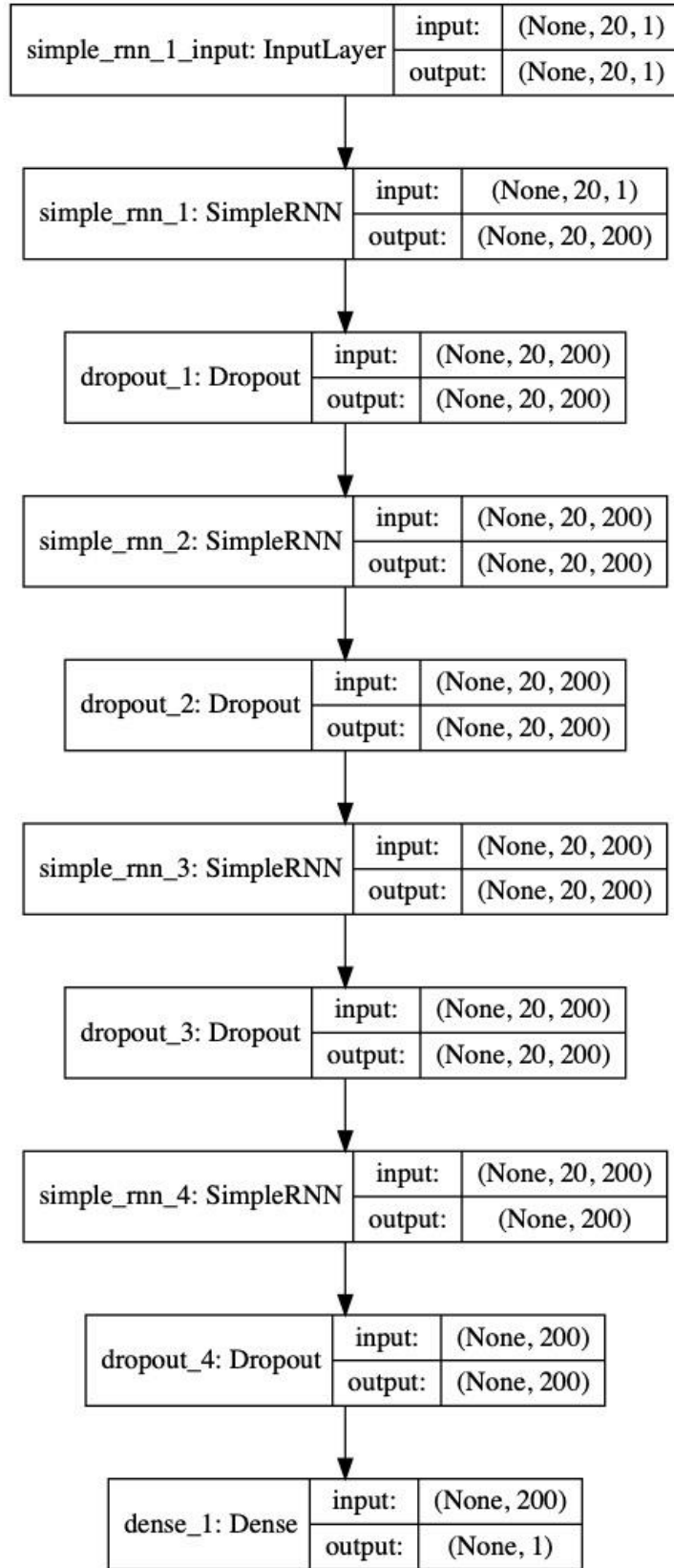
### 4.3. Tekrarlayan Sinir Ağları Modeli

Veriler normalize edilmiştir. Eğitim ve test kümesini belirlemek için Denklem 1 ve Denklem 2 kullanılmıştır. Veri setinin %80'ini (Denklem 1) test kümesinde kalan % 20'si (Denklem 2) test kümesinde yer almaktadır ve adım değeri 20 olarak belirlenmiştir. RNN ağı toplam 5 katmandan oluşmaktadır. İlk 4 katmanda blok sayısı 200 olarak verildiğinde tahmin değerlerinin mevcut verilere göre anlamlı tahminler ürettiği görülmüştür. Bu nedenle blok sayısı 200 olarak belirlenmiştir. Blok sayısının 200'den fazla belirlenmesi sonucunda tahminlerde anlamlı bir fark görülmemiş ve işlem süresi uzamıştır. Aktivasyon fonksiyonu olarak tanh fonksiyonu kullanılmıştır. Bunun nedeni bu fonksiyonun  $[-1,1]$  aralığında değerler alabilmesidir. Alabileceği değer sayısının fazla olmasının hızlı öğrenmeye katkısı da yüksek olmaktadır. Eşik değeri 0.2 olarak seçilmiştir. Bu değer aşırı uydurma sorununun önüne geçilmek için yapılmıştır ve RNN katmanlarına bağlıdır. Son katmanda bir tane çıktı verileceği için blok sayısı 1 olarak verilmiştir. Ağın iyileştirme fonksiyonu (optimizer) "adam (momentum)" olarak seçilmiştir. Bu iyileştirme metodunda öğrenme değeri (learning rate) ve bunun artış değerleri verilebilmektedir. Bunun dışın iyileştirme için seçilen kayıp fonksiyonu (loss function) değerini göz önüne alır ve öğrenme değerini revize eder. RNN ağı için kayıp fonksiyonu Ortalama Karesel Hata (MSE) olarak belirlenmiştir. En iyi sonuç tekrar sayısı (epoch) 250 olarak verildiğinde ve bir parti içindeki veri sayısı (batch size) 32 olarak verildiğinde elde edilmiştir. Bu parti değeri aynı anda kaç verinin ele alınıp ileri ve geri yayılım algoritmasının uygulanacağını göstermektedir. Şekil 6'da RNN modeli için gözlem ve tahmin değerleri grafik ile görselleştirilmiştir.



Şekil 6. RNN Modelinin Hisse Senedi Fiyat Tahmin Grafiği

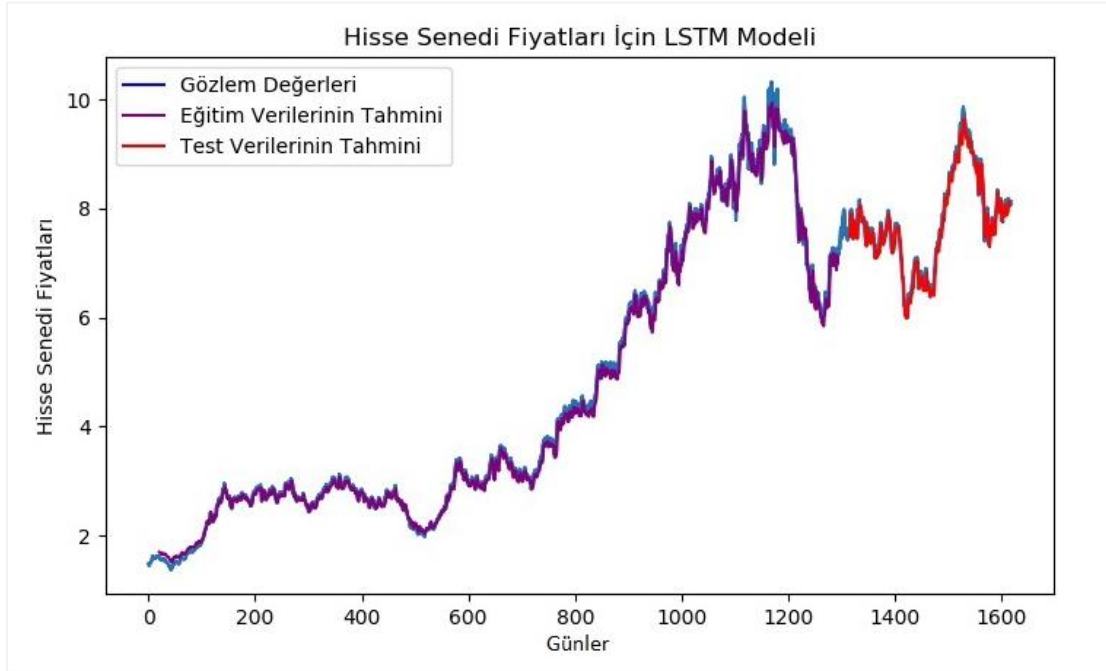
RNN modelinin ağ yapısı Şekil 7'de verilmiştir. Ağ yapılarında giriş katmanı ayrı bir katman olarak sayılmaz. "simple\_rnn\_1\_input:InputLayer", RNN ağının girdi katmanıdır. 20 adım sayısıdır ve blok sayısı 1'dir. Giriş katmanının girdisi (input) de çıktısı (output) da aynı değerleri alır. Bir sonraki katmanın (simple\_rnn\_1:SimpleRNN) girdi değeri giriş katmanının çıktı değeriyle aynıdır. Çıktı değeri ise daha önce belirlenen sabit adım değeri (20) ve birinci katmanın blok sayısı olan 200'dür. "dropout\_1: Dropout" katmanı eşik değerin veriliği katmanıdır ve her RNN katmanına bağımlı olan bir Dropout katmanı vardır. Bu nedenle algoritmayı Python 3.7 programlama dilinde yazarken Dropout katmanı için herhangi bir blok değeri yazılmamış olmasına rağmen bağlı olduğu RNN katmanının çıktısıyla Dropout katmanının girdisi ve çıktısı aynı değerlere sahiptir. "simple\_rnn\_4:Simple RNN" katmanı son katmandan bir önceki katman olduğu için girdisi diğer RNN katmanlarıyla aynıdır ancak çıktısında adım değeri görülmemektedir. Son katmanda (dense\_1:Dense) ise girdisi bir önceki katmanın çıktısıyla aynıdır ancak çıktısı Dense katmanında sadece bu katman için belirlenmiş olan blok sayısı 1 görülmektedir.



Şekil 7. RNN Modelinin Ağ Yapısı

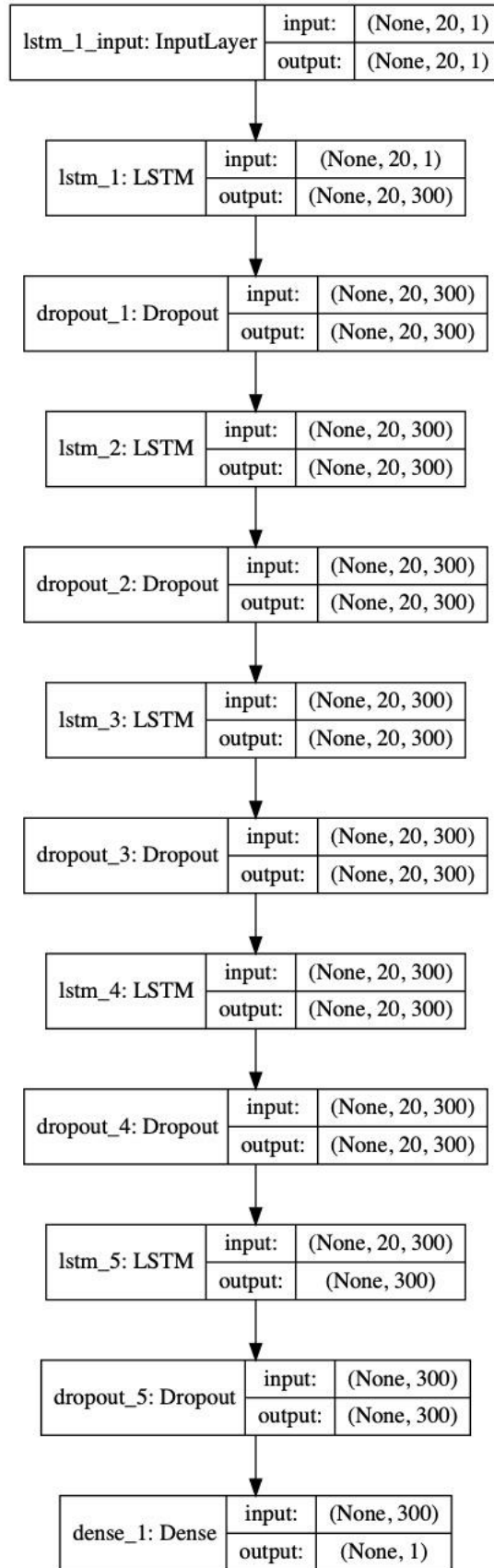
#### 4.4. Uzun-Kısa Süreli Bellek Modeli

Veriler normalize edilmiştir. Eğitim ve test kümesini belirlemek için Denklem 1 ve Denklem 2 kullanılmıştır. Veri setinin %80'ini (Denklem 1) test kümesinde kalan % 20'si (Denklem 2) test kümesinde yer almaktadır ve adım değeri 20 olarak belirlenmiştir. LSTM ağı toplam 6 katmandan oluşmaktadır. İlk 5 katmanda blok sayısı 300 olarak verildiğinde tahmin değerlerinin mevcut verilere göre anlamlı tahminler ürettiği görülmüştür. Bu nedenle blok sayısı 300 olarak belirlenmiştir. Blok sayısının 300'den fazla belirlenmesi sonucunda tahminlerde anlamlı bir fark görülmemiş ve işlem süresi uzamıştır. Eşik değeri 0.2 olarak seçilmiştir. Bu değer aşırı uydurma sorununun önüne geçilmek için yapılmıştır ve LSTM katmanlarına bağlıdır. Son katmanda bir tane çıktı verileceği için blok sayısı 1 olarak verilmiştir. Ağın iyileştirme fonksiyonu "adam" olarak seçilmiştir. LSTM ağı için kayıp fonksiyonu MSE olarak belirlenmiştir. En iyi sonuç tekrar sayısı 250 olarak verildiğinde ve bir parti içindeki veri sayısı 32 olarak verildiğinde elde edilmiştir. Bu parti değeri aynı anda kaç verinin ele alınıp ileri ve geri yayılım algoritmasının uygulanacağını göstermektedir. Şekil 8'de sinir ağı için gözlem ve tahmin değerleri grafik ile görselleştirilmiştir.



Şekil 8. LSTM Modelinin Hisse Senedi Fiyat Tahmin Grafiği

LSTM modelinin ağ yapısı Şekil 9'da verilmiştir. "lstm\_1\_input:InputLayer", LSTM ağının girdi katmanıdır. Adım sayısı 20 olarak belirlenmiştir ve blok sayısı 1'dir. Giriş katmanının girdisi (input) de çıktısı (output) da aynı değerleri alır. Bir sonraki katmanın (lstm\_1:LSTM) girdi değeri giriş katmanının çıktı değeriyle aynıdır. Çıktı değeri ise daha önce belirlenen sabit adım değeri (20) ve birinci katmanın blok sayısı olan 300'dür. "dropout\_1: Dropout" katmanı eşik değerini veriliği katmandır ve her LSTM katmanına bağımlı olan bir Dropout katmanı vardır. Bu nedenle algoritmayı Python 3.7 program dilinde yazarken Dropout katmanı için herhangi bir blok yazılmamış olmasına rağmen bağlı olduğu LSTM katmanının çıktısıyla Dropout katmanının girdisi ve çıktısı aynı değerlere sahiptir. "lstm\_5:LSTM" katmanı son katmandan bir önceki katman olduğu için girdisi bir önceki LSTM katmanının çıktısıyla aynıdır ancak çıktısında adım değeri görülmemektedir. Son katmanda (dense\_1:Dense) ise girdisi bir önceki katmanın çıktısıyla aynıdır ancak çıktısı Dense katmanının blok sayısı olarak belirlenmiş olan 1 görülmektedir.



Şekil 9. LSTM Modelinin Ağ Yapısı

## 5. İstatistiksel Analiz

Çözüm yöntemi olarak kullanılan dört tahmin yönteminden elde edilen tahmin değerlerini ve gerçek değerleri kullanarak analizler yapıp sonuçları karşılaştırılıp verilere en uygun çözümün bulunması gerekmektedir. Bu çalışmada sonuçların analizi için hata metriklerinden Ortalama Karesel Hata (Mean Square Error-MSE), Ortalama Karekök Hata (Root Mean Square Error-RMSE) ve Ortalama Mutlak Hata (Mean Absolute Error-MAE) yararlanılmak üzere seçilmiştir. Bu hata metrikleri zaman serileri verilerinde gerçek verileri ve tahmin değerlerini karşılaştırmada ve yorumlamada sıkça kullanılan analiz yöntemlerindedir. MSE değeri denklem 3'te, MAE değeri denklem 4'te ve RMSE değeri denklem 5 ve 6'da verilmiştir (Alhnaity&Abbod, 2020).

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2 \quad (3)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (5)$$

$$RMSE = \sqrt{MSE} \quad (6)$$

SciKit-Learn kütüphanesinden metrics metodu seçilerek MSE ve MAE hata değerleri bulunabilmektedir. Metodun içindeki değerler kullanıldığında RMSE değerinin bulunması için Math kütüphanesinden sqrt fonksiyonu seçilmiştir.

Yukarıda verilen Denklem 3, Denklem 4 ve Denklem 5 kullanılarak Tablo 3'teki makine öğrenmesi algoritmalarının hata değerleri gösterilmiştir. Makine öğrenmesi algoritmalarının metrik hata değerlerine bakıldığında Random Forest Regresyon modelinin hata değerlerinin çok düşük değerler olduğu görülmüştür. MSE hata değeri % 0,14 ile en düşük hata değerini gösterirken RMSE hata değeri % 3,7 ile en yüksek hata değerini göstermektedir. Polinom Regresyon için bakıldığında ise hata değerlerinin yüksek olduğu görülmüştür. MSE hata değeri % 44 ile en düşük hata değerini verirken RMSE hata değeri % 66 ile en büyük hata değerini vermiştir. İki yöntem karşılaştırıldığında Random Forest Regresyon modelinin tahmin sonuçlarının Polinom Regresyona göre çok daha iyi bir düzeyde olduğu görülmüştür.

**Tablo 3.** Makine Öğrenmesi Yöntemlerinin Hata Metrik Değerleri

Algoritmalar	Hata Metrikleri		
	MSE	MAE	RMSE
Polinom Regresyon	0.44	0.46	0.66
Random Forest Regresyon	0.0014	0.024	0.037

SciKit-Learn kütüphanesindeki metrics metoduyla MSE ve MAE değerleri bulunduktan sonra RMSE değerini bulabilmek için Math kütüphanesindeki sqrt fonksiyonu (Denklem 6) kullanılmıştır. Derin öğrenme algoritmalarının hata değerlerinin sonuçları Tablo 4'te gösterilmiştir. RNN modelinin eğitim kümesinin sonuçlarına bakıldığında en düşük hata değerinin % 7 ile MSE ve en yüksek hata değerinin % 20 ile RMSE olduğu görülmüştür. RNN modelinin test kümesinin hata değerlerine bakıldığında MSE'nin % 9 ile en düşük hata değeri ve RMSE'nin % 23 ile en yüksek hata değeri olduğu görülmüştür Eğitim kümesi ile test kümesi karşılaştırıldığında Eğitim kümesinin hata değerlerinin test kümesinin hata değerlerinden daha iyi olduğu görülmüştür. LSTM modelinin eğitim kümesinin sonuçlarına bakıldığında en düşük hata değerinin % 1,9 ile MSE ve en yüksek hata değerinin % 13 ile RMSE olduğu görülmüştür. LSTM modelinin test kümesinin hata değerlerine bakıldığında MSE'nin % 3 ile en düşük hata değeri ve RMSE'nin % 17 ile en yüksek hata değeri olduğu görülmüştür Eğitim kümesi ile test kümesi karşılaştırıldığında Eğitim kümesinin hata değerlerinin test kümesinin hata değerlerinden daha düşük olduğu görülmüştür. RNN'in ve LSTM'in metrik hata değerlerine bakıldığında algoritmalarının sonuçlarının çok yakın olduğu ancak LSTM modelinin sonuçlarının daha iyi olduğu görülmektedir.

**Tablo 4.** Derin Öğrenme Yöntemlerinin Hata Metrik Değerleri

Algoritmalar		Hata Metrikleri		
		MSE	MAE	RMSE
RNN Sinir Ağı	Eğitim Kümesi	0.07	0.19	0.20
	Test Kümesi	0.09	0.21	0.23
LSTM Sinir Ağı	Eğitim Kümesi	0.019	0.09	0.13
	Test Kümesi	0.03	0.14	0.17

## 6. Sonuç

Bu çalışmanın amacı Ereğli Demir ve Çelik Fabrikaları T. A. Ş.'nin hisse senedi fiyatlarının borsa kapanış değer tahminlerinin makine öğrenmesi yöntemlerinden Polinom Regresyon ve Random Forest Regresyon, derin öğrenme yöntemlerinden ise RNN ve LSTM algoritmalarıyla gerçekleştirip, hisse senedi fiyat tahmini için başarılarını değerlendirmektir. Algoritmalar Python 3.7 programlama dilinde Spyder (Anaconda3) editörü kullanılarak oluşturulmuştur. Çalışma veritabanı olarak kullanılan Excel 2013'te bulunan günlerin ve hisse senedi kapanış fiyatlarının oluşturduğu veri setiyle gerçekleştirilmiştir. Veri sayısı toplam 1619 adettir. Tahmin sonuçlarının başarısının istatistik analizi MSE, MAE ve RMSE hata metriklerinden faydalanarak yapılmıştır.

Hisse senedi fiyatlarının borsa kapanış değerleri tahmini için kullanılan makine öğrenmesi ve derin öğrenme yöntemleri kendi aralarında değerlendirilmiştir. Makine öğrenmesi modellerinden Polinom Regresyon için hata değerlerine bakıldığında, bu değerlerin % 44 ve % 66 arasında değiştiği görülmüştür. Random Forest Regresyon için bakıldığında ise hata değerleri % 0,14 ve % 3,7 arasında değişmektedir. Bu iki makine öğrenmesi modelinin hata değerleri karşılaştırıldığında Random Forest Regresyon modelinin Polinom Regresyon modeline göre çok daha iyi sonuçlar verdiği gözlemlenmiştir. Derin öğrenme modellerinden RNN modelinin eğitim kümesi için hata değerlerine bakıldığında, bu değerlerin % 7 ve % 20 arasında değiştiği görülmektedir. RNN modelinin test kümesi için hata metriklerinin sonuçlarına bakıldığında hata değerlerinin % 9 ve % 23 arasında değiştiği gözlemlenmektedir. LSTM modelinin eğitim kümesi için hata değerlerine bakıldığında bu değerlerin % 1,9 ve % 13 arasında değiştiği, test kümesine bakıldığında ise hata değerlerinin % 3 ve %17 arasında değiştiği gözlemlenmiştir. İki yöntemin de hata değerleri karşılaştırıldığında üç metrik için de LSTM modelinin hata değerlerinin RNN modelinin hata değerlerinden daha düşük olduğu görülmektedir. Her iki model için de eğitim kümelerinin hata değerlerinin test kümelerinin hata değerlerinden daha düşük olduğu görülmüştür. Bunun nedeninin eğitim kümelerinin tüm verinin % 80'ini, test kümelerinin ise tüm verinin % 20'sini oluşturması olduğu düşünülmektedir.

Modellerin hata değerlerine göre başarı durumları sıralandığında sırasıyla Random Forest Regresyon modeli, LSTM modeli, RNN modeli ve Polinom Regresyon modeli olduğu görülmektedir. Hisse senedi fiyatlarını etkileyecek diğer borsa hareketlerinin veriye dahil edilmesiyle ve veri sayısının artırılmasıyla modellerin daha doğru ve güvenilir sonuçlar çıkarabileceği düşünülmektedir.

## Referanslar

- Akşehir, Z. D., & Kılıç, E. (2019). Makine Öğrenmesi Teknikleri ile Banka Hisse Senetlerinin Fiyat Tahmini. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi.
- Alhnaity, B., & Abbod, M. (2020). A New Hybrid Financial Time Series Prediction Model. Engineering Applications of Artificial Intelligence.
- Baek, Y., & Kim, H. Y. (2018). ModAugNet: A New Forecasting Framework for Stock Market Index Value with An Overfitting Prevention LSTM Module and A Prediction LSTM Module. Expert Systems with Applications.
- Bahadır, İ. (2008). Bayes Teoremi ve Yapay Sinir Ağları Modelleriyle Borsa Gelecek Değer Tahmini Uygulaması.
- Çalışkan, M. M., & Deniz, D. (2015). Yapay Sinir Ağlarıyla Hisse Senedi Fiyatları ve Yönlerinin Tahmini. Eskişehir Osmangazi Üniversitesi İİBF Dergisi.
- Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science.
- Kantar, L., (2019). Bist100 Endeksinin Yapay Sinir Ağları ve Arma Modeli İle Tahmini. 23. Finas Sempozyumu— Marmara Üniversitesi İşletme Fakültesi. İstanbul: Gelişim Üniversitesi.
- Kim, H. Y., & Won, C. H. (2018). Forecasting the Volatility of Stock Price Index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models. Expert System with Applications.
- Kutlu, B., & Badur, B. (2009). Yapay Sinir Ağları İle Borsa Endeksi Tahmini.
- Mittal, A. (2019, Ekim 12). towards data science: <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e>
- Moralı, T. (2011). İMKB 100 Endeksinin Yapay Sinir Ağları ve Newton Nümerik Arama Modelleri ile Tahmini ve Sonuçların Karşılaştırılması.

Özer, A., Sarı, S. S., & Başakın, E. E. (2017). Bulanık Mantık ve Yapay Sinir Ağları ile Borsa Endeks Tahmini: Gelişmiş ve Gelişmekte Olan Ülkeler Örneği.

Pabuçcu, H. (2019). Borsa Endeksi Hareketlerinin Makine Öğrenme Algoritmaları İle Tahmini. Bayburt. azure.microsoft.com: <https://azure.microsoft.com/tr-tr/overview/what-are-private-public-hybrid-clouds/>

Raşo, H., & Demirci, M. (2019). Predicting the Turkish Stock Market BIST 30 Index using Deep Learning. Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi.

Sakarya, Ş., & Yılmaz, Ü. (2019). Derin Öğrenme Mimarisi Kullanarak BİST30 İndeksinin Tahmini. European Journal of Educational & Social Sciences .

Şafak, H. İ. (2017, Aralık 9). <https://medium.com/>: <https://medium.com/t%C3%BCrkiye/makine-%C3%B6%C4%9Frenmesi-nedir-20dee450b56e>

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction Using Machine Learning Techniques. Procedia Computer Science.