

Çok Değişkenli Finansal Zaman Serisi Analizinde Yapay Zekanın Kullanımı

Mustafa Öngün¹

1.OptiWisdom Inc.

Özet

Yapay zekâ günümüzde spordan sanata, siyasetten ekonomiye birçok alanda yaygın bir biçimde kullanılmaktadır ve artarak yaygınlaşmaya devam ettiği anlaşılmaktadır. Bu makalede, yapay zekanın ekonomi alanındaki kullanımlarından biri olan bir hisse senedinin tahmin algoritmalarına odaklanılmıştır. Çok değişkenli günlük zaman serisi olarak ele alınan problemde bağımlı değişken Vestel Beyaz Eşya Şirketi hisse senedinin belirli bir zaman aralığındaki kapanış değerleridir ve bağımsız değişkenler, bu hisse senedini etkileyen diğer değerlerdir. Bu değişkenlerden oluşturulan veri kümesi kullanılarak lineer regresyon, polinomsal regresyon, destek vektör makinesi, karar ağacı, rassal orman ve LSTM (long-short term memory) algoritmalarıyla tahminleme modelleri uygulanmıştır. Modeller içerisinde optimum sonucu veren algoritmanın hangisi olduğu, modellerin hata metrikleri karşılaştırılarak tartışılmış ve çok değişkenli finansal zaman serisi analizi için en uygun yöntemin LSTM olduğu gösterilmiştir.

Anahtar Kavramlar: Çok Değişkenli Zaman Serisi Analizi, Makine Öğrenmesi, Derin Öğrenme

Abstract

Today, artificial intelligence is widely used in many fields from sports to art, from politics to economy, and it is understood that it continues to become increasingly widespread. This article focuses on forecasting algorithms of a stock price, which is one of the uses of artificial intelligence in economics. In the problem, which is considered as a multivariate daily time series, the dependent variable is the closing values of the stock price of Vestel Beyaz Eşya in a certain time interval, and the independent variables are other values that affect this stock. By using the dataset created from these variables, prediction models were applied with linear regression, polynomial regression, support vector machine, decision tree, random forest and LSTM (long-short term memory) algorithms. The algorithm that gives the optimum result among the models is discussed by comparing the error metrics of the models and it is shown that the most appropriate method for multivariate financial time series analysis is LSTM.

Keywords: Multivariate Time Series Analysis, Machine Learning, Deep Learning

1.Giriş ve Tanımlar

Makine öğrenmesinin tarihi 1950’li yıllara dayanmaktadır. IBM çalışanı Arthur Samuel 1959 yılında ilk defa makine öğrenmesi ifadesini kullanmıştır [1]. Bu çalışmayı 1965 yılında Nilsson’un ‘Learning Machines’ isimli çalışması izlemiştir [2]. O günlerden günümüze kadar makine öğrenmesiyle ilgili birçok algoritma geliştirilmiştir [3]. Bu gelişmelerden birisi de yapay sinir ağları algoritmalarının kullanılmaya başlanmasıdır. İnsan beyninin nörolojik yapısından esinlenilerek makine öğrenmesinin çeşitli katmanlarla birbirlerine sinir ağlarına benzer yapılarla bağlandığı bu algoritmalara derin öğrenme denmektedir. Makine öğrenmesi ve derin öğrenme algoritmaları geliştikçe kullanım alanları da zenginleşmiştir. Bu kullanım alanlarından biri de hisse senetlerinin ve hisse senedinin değerini

etkileyen bağımsız değişkenlerin geçmiş değerlerini kullanarak, hisse senedinin gelecek değeri tahmininin yapılmasıdır.

Hisse senedi tahmini zaman serisi analizi yaklaşımıyla yapılmaktadır. Zaman serisi analizi ardışık olarak elde edilen zamansal verilerin kullanıldığı bir analiz yöntemidir [4]. Bu analiz yönteminde veri kayıtları kronolojik bir düzende dizilmektedir. Bu kronoloji bozulmadan makine öğrenmesi ve derin öğrenme algoritmaları kullanılmaktadır. Jason Brownlee zaman serisinin gözlemler arasına açık bir düzen bağımlılığı eklediğini ve bunun bir zaman boyutu olduğunu; bu ek boyutun hem bir kısıtlama hem de ek bilgi kaynağı sağlayan bir yapı olduğunu belirtmektedir [5]. Bu makaleye konu olan çalışmada da zaman serisi analizi metodları kullanılmıştır. Eğitim ve test verileri seçilirken kronolojik düzenin bozulmamasına dikkat edilmiştir. Bu makalede ikinci kısımda çalışmanın ele aldığı problem, kullanılan veri kümeleri, hata metrikleri ve tahmin algoritmaları açıklanacaktır. Üçüncü kısımda bu algoritmaların tahmin uygulamalarında kullanımı ve sonuçları açıklanacaktır. Dördüncü kısımda bütün uygulamaların hata oranları karşılaştırılacaktır. Son kısım olan sonuç bölümünde optimum modelin hangisi olduğu tartışılacaktır.

2. Problem, Materyal ve Yöntemler

Bu bölümde çalışmada kullanılan veri kümesi, hata metrikleri ve algoritmalar açıklanacaktır. Çalışmada ilk olarak veriler internet üzerinden elde edilmiştir. Elde edilen veriler düzenlenmiş ve veri kümesi oluşturulmuştur. Oluşturulan veri kümesine çeşitli makine öğrenmesi modelleri ve LSTM [6] modeli uygulanmıştır. Bu uygulamaların sonuçları belirlenmiş olan hata metrikleri bazında karşılaştırılmıştır. Çalışmanın proje akışı diyagramı şekil-1'de gösterilmiştir.



Şekil-1: Çalışmanın Proje Akış Diyagramı

2.1 Problem

Şirketler ve yatırımcılar para politikalarını belirlerken ekonomik parametrelerin öngörülebilir ve nispeten yüksek oranda doğru tahminlenebilir olmasını beklemektedir. Şirketlerin hisse senedi fiyatlarının diğer ekonomik değişkenlere bağlı olarak aldığı değerlerin modellenmesi, bu fiyatların gelecekteki değerlerini tahminleme için kullanılmaktadır. Bu tahminleme modelleri, şirketlerin ve yatırımcıların kâr oranını arttırmak için ve doğru para politikalarını belirleyebilmeleri için önemli bir araç olmaktadır. Bu çalışmada bir şirketi hisse senedi fiyatları tahminlemesi için yapay zekâ modelleri kurulmuş ve bu modellerle hisse senedinin gelecek değeri tahminlemesi yapılmıştır.

2.2 Veri Kümesi

Çalışmada kullanılan veri kümesi 10 Ekim 2018 tarihinden 1 Eylül 2022 tarihine kadar olan, Vestel Beyaz Eşya Sanayi ve Ticaret A.Ş.'nin hisse değeri, Brent Crude Oil hisse değeri, Türkiye Cumhuriyeti Merkez Bankası TÜFE değerleri, FED faiz oranı, USD/TRY ve XAU/USD değerleridir. Veri kümesi 1423 satır ve 7 sütundan oluşmaktadır. Türkiye

Cumhuriyeti Merkez Bankası TÜFE değerleri TCMB'nin kendi sitesinden [7], diğer veriler Investing'den [8] indirilmiştir. Tablo-1'de kullanılan veri kümesinin ilk 10 satırı görülmektedir. Tablodan da görüldüğü gibi veriler tarih sırasıyla kronolojik olarak sıralanmaktadır.

Tablo-1: Kullanılan veri kümesinin ilk 10 satırı

Tarih	Kapanış_broil	Kapanış_usdtry	Enflasyon	Faiz	fed_value	Kapanış_ons	Kapanış_vstl
2018-10-10	83,09	6,076	401,27	23,64920635	2,18	119340	0,839
2018-10-11	80,26	5,9258	401,27	23,63571429	2,18	122760	0,841
2018-10-12	80,43	5,8721	401,27	23,62222222	2,18	122200	0,846
2018-10-13	80,43	5,8721	401,27	23,60873016	2,18	122200	0,846
2018-10-14	80,43	5,8721	401,27	23,5952381	2,18	122200	0,846
2018-10-15	80,78	5,7867	401,27	23,58174603	2,18	123030	0,879
2018-10-16	81,41	5,6855	401,27	23,56825397	2,18	123100	0,882
2018-10-17	80,05	5,5835	401,27	23,5547619	2,19	122740	0,877
2018-10-18	79,29	5,6425	401,27	23,54126984	2,19	123010	0,87
2018-10-19	79,78	5,6425	401,27	23,52777778	2,19	122870	0,901

2.3 Veri Ön İşleme (Data Preprocessing)

Veri kümesi, pandas kütüphanesi kullanılarak Python dosyasına veri çerçevesi (data frame) olarak aktarılmıştır. Her bir verinin tarih sütunu 'to_datetime' fonksiyonuyla tarih formatına çevrilmiş, 'resample' fonksiyonu kullanılarak günlük zaman serisine döndürülmüştür. Her bir veri çerçevesindeki boş satırlar, 'interpolate' fonksiyonunun 'linear' metodu kullanılarak iki dolu satır arasındaki değerleri lineer bir artış değerine denk olacak şekilde doldurulmuştur. Tüm veri çerçeveleri pandas kütüphanesinin 'merge_asof' fonksiyonu kullanılarak birleştirilerek hepsinin değerlerini içeren bir ana tablo oluşturulmuştur. Hisse değerlerinde açılış, en yüksek, en düşük, kapanış ve hacim değerlerinden yalnızca kapanış değeri baz alınmıştır. Vestel Beyaz Eşya Sanayi ve Ticaret A.Ş. kapanış değeri bağımlı değişken, diğer değerlerse bağımsız değişken olarak kullanılmıştır. Makine öğrenmesi algoritmalarında veri kümesinin ilk %80'i eğitim, son %20'si test verisi olacak şekilde veriler ayrılmıştır.

2.4 Hata Metrikleri

Çalışmanın başarısını sayısal olarak ölçmek için bazı hata metrikleri kullanılmıştır. Bu metrikler mean squared error, mean absolute error ve r^2 'dir [9]. Mean squared error (MSE) hata karelerinin ortalamasıdır. Mean squared error değerini elde etmek için test verisindeki tüm gözlemcilerin gerçek değerleriyle tahmin edilen değerlerinin farkının karesi alınmaktadır ve daha sonra gözlemci sayısına bölünerek bu ortalama hata karesi elde edilmektedir. Mean absolute error (MAE) hata metriği, mean squared error metriğini hesaplama yöntemine çok benzer bir yöntemle hesaplanmaktadır. İki metriğin hesaplanması arasındaki tek fark, mean absolute error hata metriğinde test verisindeki gerçek değerle tahmin edilen değerin farkının mutlak değeri alınarak hesaplanmasıdır. R^2 değeri 0 ile 1 arasında değişmektedir ve 1'e yakın olduğu ölçüde tahmin modelimizin doğruluğunu teyit etmektedir. Çalışmada makine öğrenmesi algoritmaları ve derin öğrenme algoritması bu hata metrikleriyle ölçülerek birbirleriyle karşılaştırılmıştır.

2.5 Algoritmalar

Çalışmada zaman serisi analizi çeşitli makine öğrenmesi ve derin öğrenme algoritmalarıyla yapılmıştır. Kullanılan makine öğrenmesi algoritmaları; lineer regresyon analizi [10], polinomsal regresyon analizi [11], destek vektör makinesi (support vector machine) [12], karar ağacı (decision tree) [13] ve rassal orman (random forest) [14] yöntemidir. Polinomsal regresyon analizi yapılrken ikinci dereceden polinom analizi gerçekleştirilmiştir. Derin öğrenme algoritması olarak ise LSTM kullanılmıştır.

2.5.1 Lineer ve Polinomsal Regresyon

Regresyon modelleri, değişkenler arasındaki ilişkilerle ilgili bilimsel çalışmalarda sıklıkla kullanılmaktadır. Bu modelde bir yanıt değişkeninin bir dizi açıklayıcı değişkene bağımlılığı açıklanmaktadır. Regresyon modelinde 'Y' sembolüyle temsil edilen bağımlı değişken, 'X' sembolüyle temsil edilen bağımsız değişkene göre tahmin edilmektedir. Çok değişkenli lineer regresyonda ise birden fazla bağımsız değişken kullanılmaktadır. Polinomsal regresyon analizinde bağımlı ve bağımsız değişkenler arasındaki ilişki n dereceli bir polinomla formülize edilmektedir.

2.5.2 Destek Vektör Makinesi (Support Vector Machine)

SVM algoritmaları veri analizinde regresyon ve sınıflandırma için kullanılmaktadır. Bu algoritma hem lineer hem de non-linear regresyon ve sınıflandırmalarda verimli sonuçlar vermektedir. SVM non-linear regresyon ve sınıflandırmada çekirdek hilesi (kernel trick) yöntemini kullanarak verimliliğini arttırmaktadır. Temel olarak sınıflar arasındaki farkı çizmekte; bu farkın maksimum olduğu değerleri farklı sınıflara ayırarak hatayı minimize etmektedir. Stefan Rüping, 'SVM kernels for time series analysis' isimli çalışmasında SVM modelinin zaman serisi analizinde kullanım yollarını, özellikle çok değişkenli zaman serisinde çekirdek(kernel) seçimini tartışmaktadır. Çekirdek(kernel) yöntemleri içerisinde optimum seçimin 'RBF' olduğunu bu çalışmasında belirtmektedir [15]. Çalışmada, bu veriler ışığında 'RBF' çekirdeği kullanılmıştır.

2.5.3 Karar Ağacı (Decision Tree)

Karar ağacı (decision tree) sınıflandırma ve regresyon yöntemlerinde kullanılan bir akış şemasıdır. Karar ağacı yapısı basit sorular ve bu soruların cevaplarına göre şekillenen, dallanan bir yol şeklinde tasarlanmıştır. Karar ağacı yapısında veriler bu cevaplara göre hiyerarşik bir şekilde gruplara ayrılırlar. Bu hiyerarşiye ağaç, hiyerarşideki parçalara düğüm denmektedir. Orijinal verinin tamamı kök düğüm (root node) olarak adlandırılmaktadır. Bir düğüm, ardıllarıyla birlikte onu oluşturan düğümün bir dalını oluşturmaktadır; son düğümlerse yaprak olarak adlandırılmaktadır. Her yaprak için bir karar verilmekte ve yapraktaki tüm gözlemlere uygulanmaktadır.

2.5.4 Rassal Orman (Random Forest)

Karar ağacı algoritması bazı durumlarda optimal sonuçlar üretmemesi ve düşük değişimlere duyarlı hassas verilerde yüksek varyans üretmesi gibi sorunlar ortaya çıkarmaktadır. Rassal orman (random forest) algoritması bagging metodunu kullanarak bu sorunları minimize etmek amacıyla kullanılmaktadır. Bagging metodu, birden fazla karar ağacı oluşturup bunları birleştirerek ortalamalarını almaktadır. Bu metotla varyans karar ağacı algoritmasına göre azaltılmaktadır. Tüm yanıtların ağırlıklı ortalaması tespit edilerek her bir eğitim adımında ağırlıklar tekrar dağıtılmakta ve ağırlık artırma(boosting) işlemi sıralı olarak yapılmaktadır. Bu işlemler esnasında birbirinden bağımsız birçok karar ağacı yapısı çalıştırıldığından bilgisayar için yorucu olabilmekte aşırı ısınmaya neden olabilmektedir. Rassal orman algoritmasında eğitim verisi rastgele seçilmektedir.

2.5.5 LSTM (Long-Short Term Memory)

LSTM, RNN'deki veriyi uzun süreli olarak bellekte tutma problemlerine, vanishing gradient metodunun getirdiği modelin öğrenme yavaşlığı hatta bazen öğrenmenin durması sorununa bir çözüm olarak 1997 yılında Sepp Hochreiter ve Jürgen Schmidhuber tarafından önerilmiş bir derin öğrenme ağ modelidir. LSTM yapıları birbirlerinden daha uzun mesafelerle ayrılmış olan girdilerden öğrenebilirler, RNN'den daha uzun hafızaya sahiptirler. LSTM girdi kapısı, çıktı kapısı, unutma kapısı olmak üzere üç kapıdan ve kapılar yardımıyla bilgiyi tutmayı ve unutmayı sağlayan hücre durumundan oluşan bir sistemdir. Hücre durumu hücre içerisinde taşınması gereken verileri taşıyan ve ağ üzerindeki

veri akışını sağlayan yapıdır. Unutma kapısından gelen veriyle bir önceki katmandan gelen veri çarpılarak girdi kapısındaki değerle toplanmaktadır. Girdi kapısı yeni girdinin girmesine izin verilip verilmeyeceğine karar veren, bunun için veriyi sigmoid fonksiyonundan geçiren kapıdır. Unutma kapısında bir verinin silinip silinmeyeceğine karar verilmekte ve önemli olmayan bilgiler silinmektedir. Verinin önemli olup olmadığına karar vermek için önceki gizli katmandaki verilerle şimdiki veriler sigmoid fonksiyonuna sokulmakta, 0 ile 1 arasında bir değer olan sonuç 0'a yakınsa silinmekte 1'e yakınsa kullanılmaktadır. Çıktı kapısı hangi bilgilerin çıktı olarak alınacağına karar vermektedir [16-17].

3.Uygulama

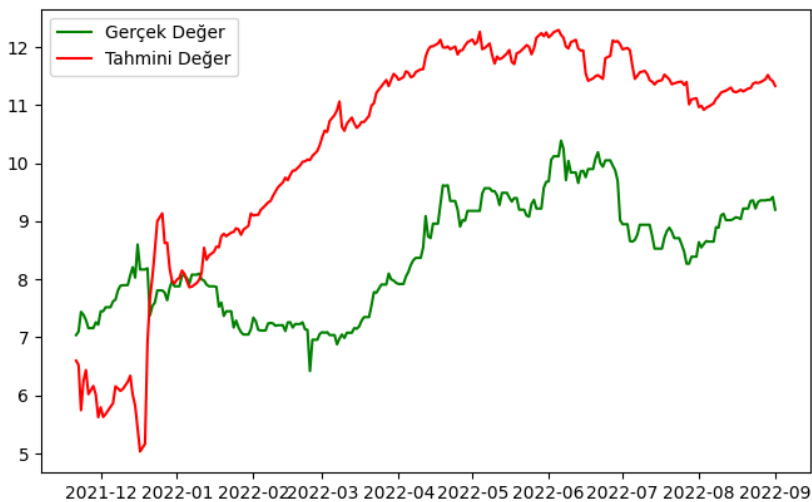
Bu kısımda birinci kısımda açıklanmış olan algoritmaların kullanılan veri kümesi üzerinde uygulaması ve çıktıları açıklanmıştır. Uygulama Python programlama diliyle yazılmıştır. Uygulamada Python yazılım dilinin numpy, pandas, sklearn, keras, matplotlib, seaborn, plotly ve math kütüphaneleri kullanılmıştır.

3.1 Lineer ve Polinomsal Regresyon ile Tahmin Uygulaması

Lineer ve polinomsal regresyon modelleri eğitim için ayrılan verilerle kurulmuş, test verisi için ayrılan verilere uygulanmıştır. Polinomsal regresyonda ikinci derece polinom modeli kullanılmıştır. Lineer regresyon uygulamasında test verisi ve bu verinin tahmininin grafiği şekil-2'de gösterilmiştir. Lineer regresyon uygulamasının hata skorlarıysa tablo-2'de gösterilmiştir.

Tablo-2: Lineer Regresyon Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
-5,5649768	2,293832994	6,074728401

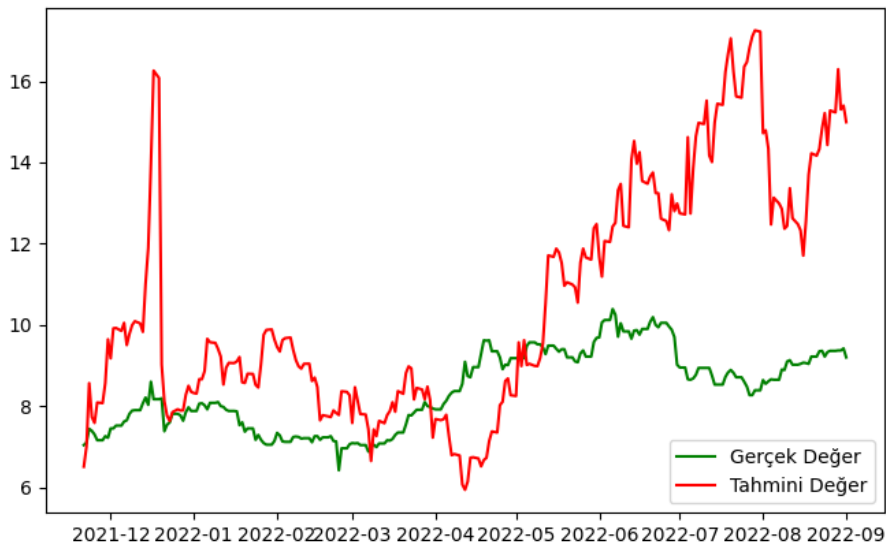


Şekil-2: Lineer Regresyon Modeli Grafiği

Polinomsal regresyon modeli tahmini ve gerçek verilerin zamana göre grafiği şekil-3'te, hata metriklerinin skorlarıysa tablo-3'te gösterilmiştir.

Tablo-3: Polinomsal Regresyon Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
-11,127129	2,558479922	11,22151903



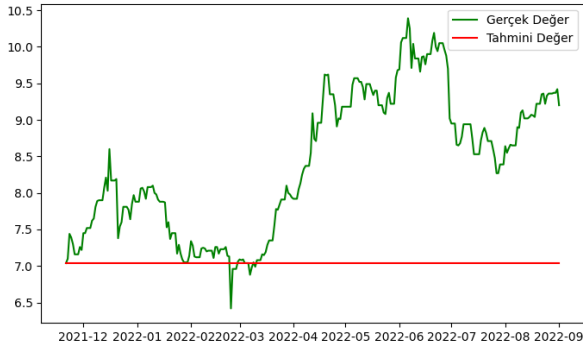
Şekil-3: Polinomsal Regresyon Modeli Grafiği

3.2 Destek Vektör Makinesi (Support Vector Machine) ile Tahmin Uygulaması

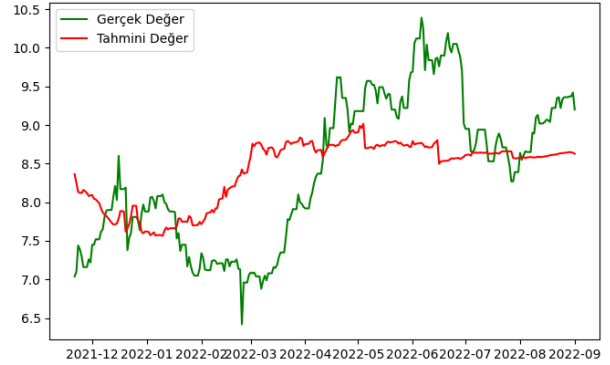
Çalışmada destek vektör makinesi (support vector machine) algoritması zaman serisi analizine uyarlanmıştır. Kernel olarak 'rbf' kullanılmıştır. Veri kümesindeki tüm veriler 'standard scaler' yöntemiyle standardize edilerek destek vektör makinesi algoritmasıyla destek vektör tahminleme (support vector regressor) modeli oluşturulmuştur. Modele verilerin gerçek değerlerini girdi olarak uygulayabilmek için veriler standardize edilmiş değerlerden tekrar gerçek değerlerine dönüştürülerek tahminleme işlemi gerçekleştirilmiştir. Şekil-10'da bu modelin tahmin değerleri gerçek değerlerle birlikte çizgi grafiği şeklinde gösterilmiştir. Tablo-4'te ise modelin hata skorları gösterilmiştir.

Tablo-4: Destek Vektör Makinesi Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
0,28916731	0,662371032	0,657750312



Şekil-4: Destek Vektör Makinesi Modelinin Tahmin Grafiği



Şekil-5: Karar Ağacı Modelinin Tahmin Grafiği

3.3 Karar Ağacı(Decision Tree) ve Rassal Orman(Random Forest) ile Tahmin Uygulaması

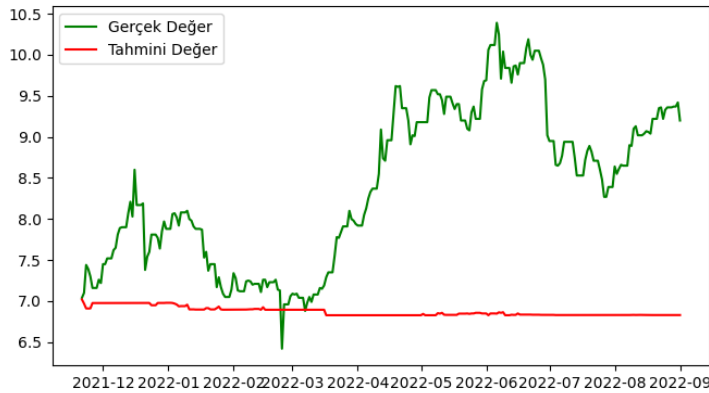
Çalışmada karar ağacı ve rassal orman modellerinin her ikisinde de 'random state' değişkeni 0 olarak belirlenmiştir. Rassal orman modelinde 'n_estimators' değişkeni hata oranlarına göre for döngüsüyle 10, 50, 100, 500 ve 1000 değerlerinde çalıştırılmış ve optimum değer olan 500 olarak alınmıştır. Şekil-11'de karar ağacı modelinin, Şekil-12'de rassal orman modelinin gerçek verilerle birlikte tahmin verileri çizgi grafiği şeklinde gösterilmiştir. İki modelin hata skorları tablo-5 ve tablo-6'da gösterilmiştir.

Tablo-5: Karar Ağacı Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
-1,989905	1,364877193	2,766629825

Tablo-6: Rassal Orman Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
-2,5840923	1,528413768	3,31644541



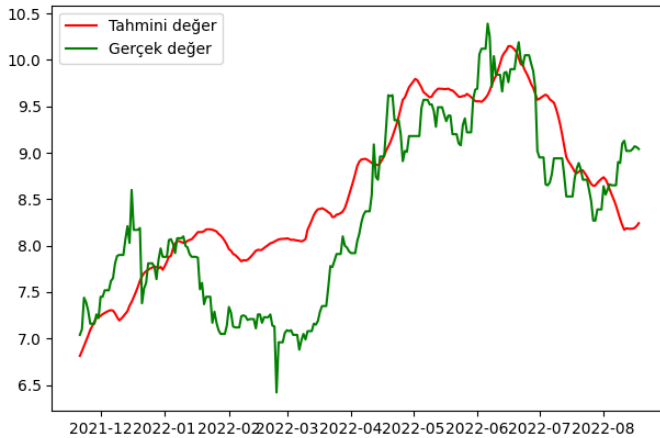
Şekil-6: Rassal Orman Modelinin Tahmin Grafiği

3.4 LSTM ile Tahmin Uygulaması

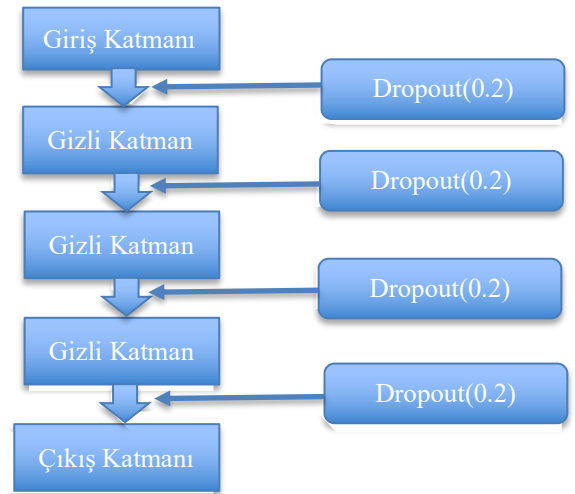
LSTM modeli 1 giriş katmanı, 1 çıkış katmanı, 4 gizli katman ve 4 dropout katmanı kullanılarak kurulmuştur. Giriş katmanında 512 birim(unit), gizli katmanların her birinde 255 birim kullanılmıştır. Dropout katmanı 0.2 değeriyle kullanılmıştır. Dropout katmanlarının kullanımıyla modelin aşırı öğrenmesinin minimuma indirilmesi amaçlanmıştır. Veri kümesinin yüzde 80'ine tekabül eden ilk 1138 nokta eğitim verisi, son 285 nokta test verisi olarak kullanılmıştır. LSTM modelinin uzun süreli hafızasını modellemek için 14 veri noktası içeren ve 'window size' olarak adlandırılan yapılar kaydırılmalı olarak kullanılmıştır. Bu kullanımda son 14 veri tahmin edilemeyeceği için veri kümesinin son 14 satırı çıkarılmıştır. Veriler daha iyi sonuç elde edebilmek amacıyla 'min-max scaler' yöntemiyle standardize edilmiştir. Optimize edici (optimizer) olarak 'adam' fonksiyonu 0.0001 öğrenme oranı (learning rate), hata metriği olarak 'mse', kayıp(loss) değişkeni olarak 'mean squared error' kullanılmıştır. Epoch parametresi 3, batch size parametresi 20 olarak ayarlanmıştır. LSTM modeli akış şeması şekil-8'de gösterilmiştir.

Tablo-7: LSTM Modeli Hata Skorları

R2_skoru	MAE_skoru	MSE_skoru
0,623305846	0,480033648	0,349350852



Şekil-7: LSTM Modeli Tahmin Grafiği



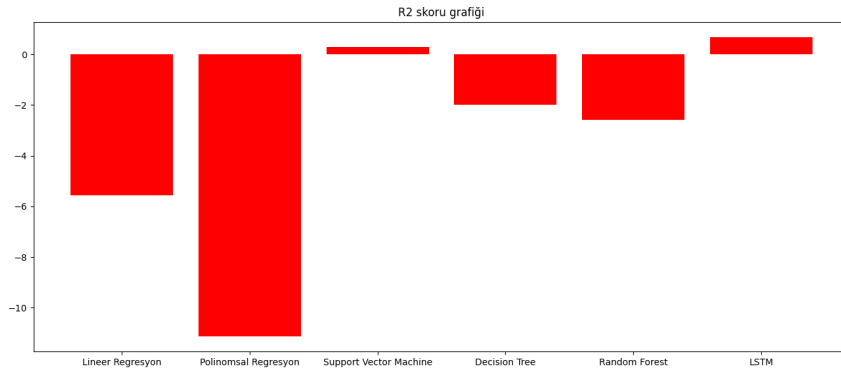
Şekil-8: LSTM Modeli Akış Şeması

4. Modellerin Karşılaştırılması

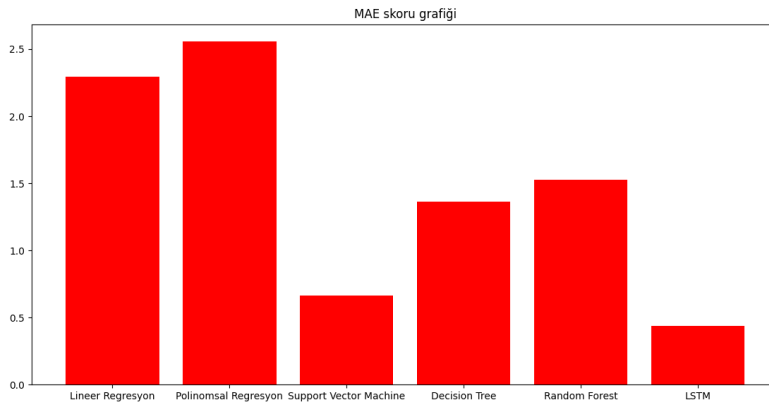
Çalışmada verilerin zaman serisi analizi yöntemiyle çeşitli modeller kullanılarak yapılmış olan tahminleme uygulamalarının hata oranları tablo-8'de ve şekil-8'de gösterilmiştir.

Tablo-8: Tüm Modellerin Hata Skorları

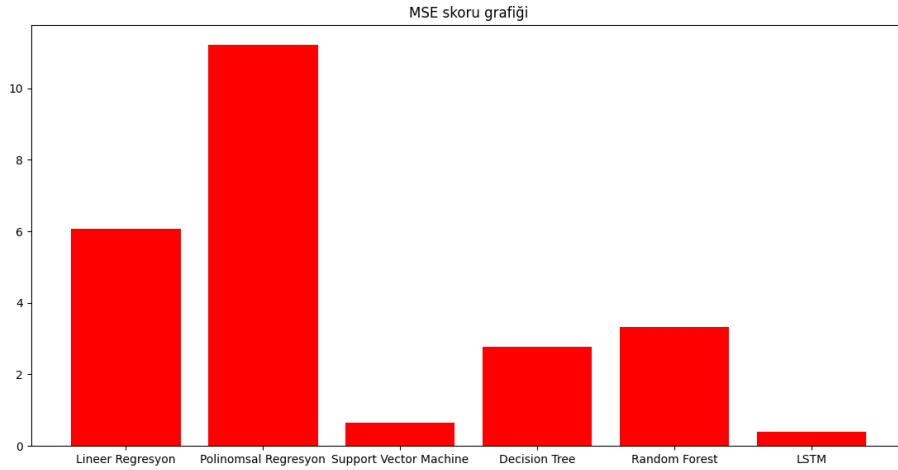
Model	R2_skoru	MAE_skoru	MSE_skoru
Lineer Regresyon	-5,564976816	2,293832994	6,074728401
Polinomsal Regresyon	-11,12712856	2,558479922	11,22151903
Support Vector Machine	0,289167307	0,662371032	0,657750312
Decision Tree	-1,989904973	1,364877193	2,766629825
Random Forest	-2,58409229	1,528413768	3,31644541
LSTM	0,623305846	0,480033648	0,349350852



Şekil-9: Tüm Modellerin R2 Skoru Grafiği



Şekil-10 Tüm Modellerin MAE Skoru Grafiği



Şekil-11: Tüm Modellerin MSE Skoru Grafiği

5. Sonuç

Çalışmada hisse senedi tahminlemesi için çok değişkenli finansal zaman serisi analizi yöntemi makine öğrenmesi algoritmalarından lineer regresyon, polinomsal regresyon, support vector machine, decision tree, random forest ve derin öğrenme algoritmalarından LSTM ile modellenmiştir. Bu modeller içerisinde r2 skoru en yüksek, MAE ve MSE skorları en düşük olan modelin LSTM olduğu görülmüştür. LSTM modelinin r2 skoru 0.62, MAE skoru 0.48 ve MSE skoru 0.45 bulunmuştur. İkinci sırada en iyi değerleri ise 0.29 r2 skoru, 0.66 MAE skoru ve 0.66 MSE skoruyla support vector machine modeli vermiştir. Diğer 4 makine öğrenmesi modelinin aşırı öğrenmeden kaynaklı olarak r2 skorları negatif; MAE ve MSE skorları ise 1'den büyük çıkmıştır. Çalışmada kullanılan modeller arasında çok değişkenli finansal zaman serisi analizinde kullanılacak optimum modelin LSTM olduğu görülmüştür.

5. Kaynaklar

- [1] Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development.
- [2] Nilsson N. Learning Machines, McGraw Hill, 1965.
- [3] Mahesh, Batta (2018). "Machine Learning Algorithms - A Review". International Journal of Science and Research (IJSR)
- [4] Box G.P.E, Jenkins G.M., Reinsel G.C., Ljung G.M. (2015). "Time Series Analysis: Forecasting and Control". Wiley.
- [5] Brownlee, J. (2020). "Introduction to Time Series Forecasting with Python". Machine Learning Mastery.
- [6] Tanışman, S., Karcıoğlu, A. A., Uğur, A. & Bulut, H. (2021). Bitcoin Fiyatının LSTM Ağı ve ARIMA Zaman Serisi Modeli Kullanarak Tahmini ve Karşılaştırılması. Avrupa Bilim ve Teknoloji Dergisi, (32), 514-520.
- [7] Web tarama, www.tcmb.gov.tr, Mart 2023
- [8] Web tarama, www.investing.com, Mart 2023

- [9] Web tarama: <https://www.medium.com>, tarih: 8 Mart 2023
- [10] Fitzmaurice, G.M. (2016). “Regression”. *Diagnostic Histopathology*. Volume 22, Issue 7, July 2016, p.p. 271-278
- [11] Maulud, D. H., Abdulazeez, A. M. (2020). “A Review on Linear Regression Comprehensive in Machine Learning”. Maulud et al. / *Journal of Applied Science and Technology Trends* Vol. 01, No. 04, pp. 140 –147
- [12] Mahesh, Batta (2018). “Machine Learning Algorithms - A Review”. *International Journal of Science and Research (IJSR)*
- [13] Moore, T., Jesse, C., Kittler, R. (2001). “An Overview and Evaluation of Decision Tree Methodology”. *ASA Quality and Productivity Conference*. May 23-25, 2001
- [14] L. Xue, Y. Liu, Y. Xiong, Y. Liu, X. Cui, G. Lei (2021). “A data-driven shale gas production forecasting method based on the multi-objective random forest regression”. *Journal of Petroleum Science and Engineering* 196 (2021) 107801
- [15] Rüping, Stefan (2001). *SVM kernels for time series analysis*, Technical Report, No. 2001,43, Universität Dortmund, Sonderforschungsbereich 475 - Komplexitätsreduktion in Multivariaten Datenstrukturen, Dortmund
- [16] Tuncer, Ahmet. “LSTM Metodu Kullanılarak Rüzgar Hızının Tahmin Edilmesi”. *Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Temmuz 2022*
- [17] A. Yadav, C. K. Jha, and A. Sharan, “Optimizing LSTM for time series prediction in Indian stock market,” *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.