



Hisse Senedi Fiyatlarının Linear Regresyon ve LSTM bazlı Tekrarlayan Yapay Sinir Ağları ile Tahmini

Musawer MURADI¹ 

¹Kocaeli University, Faculty of Computer Sciences, 41380 Kocaeli, Turkey

Makale Bilgisi

Received:

Accepted:

Anahtar Kelimeler

Borsa Tahmini
Linear Regresyon
Uzun Kısa Vadeli Hafıza
Tekrarlayan Yapay Sinir
Ağları
LSTM

Öz

Günümüzün finans dünyasında en önemli amaçlardan biri hisse senetlerinin fiyatlarını tahmin ederek uygun yatırımlar yapmak ve sonuç olarak da para kazanmaktır. Dolayısıyla hisse senetleri ile fonlar, uzun zamandır yatırımcılar, araştırmacılar ve bilim dünyası için çok önemli ve gizemini koruyan bir konu haline gelmiştir. Borsa sayesinde, yatırımlarını doğru değerlendirebilenler büyük kazançlar elde edebilirken yanlış değerlendirmeler sonucunda büyük kayıplar yaşayabilmektedirler. Bu çalışmanın amacı kendi Linear Regresyon modelimizi ve Tekrarlayan yapay sinir ağları ve keras kütüphanesi Uzun kısa vadeli hafıza modellerini kullanarak, bir zaman serisi problem olan hisse senedi borsa fiyatlarının tahminini yapmaktır. Veriler güncel olarak Wikipedia ve Yahoo Finance modülünden alınarak, her bir Linear Regresyon Modelinde ve LSTM hücreleri girdisi olarak kullanılacak şekillere getirilir ve yaratılan modelde öğrenme işlemine tabi tutulur. Farklı algoritma ve hiperparametreler kullanılarak birçok kez daha iyileştirilmeye çalışılan modelimiz birçok farklı zaman serisi için tahmin yapmaya çalışacaktır ve yapılan tahmin seti gerçek verilerle karşılaştırılıp doğruluk oranı saptanacaktır. Gerçek anlamda Borsada doğru tahmin yapabilecek ve 100% doğruluk oranına yakın çalışan bir sistemin hayata geçirilişi zaman serisi problemlerinden çok daha karmaşık bir problemidir. Çünkü hisse senedini etkileyen birçok dış etken vardır. Bu çalışmada kullanılan model detaylı ve teknik açıdan açıklanmıştır. Farklı hiperparametreler ve algoritmalarla farklı sonuçlara ulaşabilmek mümkündür. Proje yazılımı olabildiğince dinamik ve kolayca farklı parametrelerle kullanılabilecek şekilde tasarlanmıştır.

1. GİRİŞ (INTRODUCTION)

Günümüzün finans dünyasında en önemli amaçlardan biri hisse senetlerinin fiyatlarını tahmin ederek uygun yatırımlar yapmak ve sonuç olarak da para kazanmaktır. Dolayısıyla hisse senetleri ile fonlar, uzun zamandır yatırımcılar, araştırmacılar ve bilim dünyası için çok önemli ve gizemini koruyan bir konu haline gelmiştir. Borsa sayesinde, yatırımlarını doğru değerlendirebilenler büyük kazançlar elde edebilirken yanlış değerlendirmeler sonucunda büyük kayıplar yaşayabilmektedirler.

Yatırımcıların doğru tahmin yaparak, yükselecek olan hisse senedini almaları (veya elde tutmaları) ve düşecek olan hisse senetlerinden kaçınmaları (veya satmaları), hangi hisse senedini veya fonu ne zaman alacağı borsada doğru hamle yapma açısından hayati önem taşımaktadır. Ancak, bu tahmin işlemi borsanın kendilerini önceden yeteri kadar göstermeyen hem politik hem de ekonomik olaylardan etkilenmesinden dolayı bu oldukça zor hale gelmektedir. Dolayısıyla bu tür olaylar tahmin yapılırken yatırımcı açısından sık ve etkili bir şekilde kullanılamamaktadır. Finansal tahminler üzerinde birçok araştırmacı uzun yıllardır çalışmaktadır. Bu çalışmalarda hedeflenen amaç, bir finansal değerin gelecekteki değerinin önceden öngörülüp öngörülemeyeceğidir ancak hisse senedi piyasaları her gün çok sayıda işlemin yapıldığı büyük sistemler olduğundan böyle dinamik, doğrusal olmayan ve hayli karmaşık sistemlerde öngörü yapmak kolay olmamaktadır. Öngörü kavramı, belirli varsayımlar altında bir değişkenin gelecekte alabileceği

*Corresponding author, e-mail: musavir.muradi@gmail.com

değerlerin önceden yaklaşık olarak belirlenmesi şeklinde tanımlanmaktadır [1]. Zaman serisi çözümlemesi kullanarak öngörü ise, incelenen bir değişkenin şimdiki ve geçmiş dönemdeki gözlem değerlerini kullanarak öngörü değerlerinin hangi sınırlar arasında gerçekleşebileceğini birtakım varsayımlar altında ortaya koyan çalışmalardır. Doğru öngörünün (tahminin) başarılı kararları beraberinde getireceği ve bu şekilde fayda sağlamanın en üst düzeye çıkartılabileceği gerçeği, öngörü modellemesine olan ilgiyi artırarak bu alanda her geçen gün önemli gelişmelerin yaşanmasına ön ayak olmaktadır. Önceden matematiksel ve istatistiksel yöntemlerle yapılan tahmin denemeleri, özellikle bilgisayar teknolojisinin gelişimiyle birlikte daha başarılı sonuçlar vermiştir.

Hisse senetleri, dünya çapında birçok yatırımcının ilgisini çekmeyi başarmıştır. Borsa fiyatlarını etkileyen, politik, ekonomik, sosyal, market durumu gibi birçok etken vardır. Bu etkenlerin riskini en aza indirmek için ekonomistler, yatırımcılar ve akademisyenler birtakım araştırmalar yürütmüşlerdir. İnsanlar hisse senedi volatilitesini öngörebilmek için K-line diyagram analizi, Hareketli ortalamaların birbiri ile yakınlık/uzaklık diyagram analizi, hatta yazı-tura ve fal bakmak gibi çeşitli yöntemler denemişlerdir. Hisse Senetleri Borsası yoğun bir volatiliteye sahiptir. Esasında, arz ve talep zincirinin hisse senedi borsasını etkilediği söylenir ama bu durum uygulamada değişiklik gösterir ve birçok farklı etken borsayı etkiler. Ülkeler, şirketler ve yatırımcılar uzun vadeli planlarını borsanın durumuna ve analizlerine göre yön veriyor. Bunun altındaki yatan amaç uzun vadede yüksek kar olanaklarına ulaşmaktır.

Bugünlerde finansal verilerin dijital ortamlarda saklanması ve erişime açık olması bize veriler üzerinde araştırma yapma olanağı sağlıyor. Gelişen yapay zekâ kavramı ve öğrenmeye dayalı algoritmalar bu süreçte önemli katkılar sağlamıştır. Özellikle son yıllarda yoğunlaşan makine öğrenmesi metodlarının ve derin öğrenme algoritmalarının finansal tahminlerde kullanımı ve alınan olumlu sonuçlar araştırmacıların bu alana olan ilgisini artırmaktadır; ancak buna rağmen piyasalardaki fiyat davranışlarını tam olarak öngörebilen bir yöntem bulunamadığından bu alandaki çalışmalar aktif olarak devam etmektedir.

Bu çalışmada bir Linear Regresyon modeli ve de bir Recurrent Neural Network ve Long Short-Term Memory kullanılarak hisse senedi fiyatlarının tahmini amaçlanmaktadır.

1.1. İlgili Çalışmalar

Bu alanda daha önce birçok çalışma yapılmıştır. Borsa eğilimlerini tahmin edebilmek için Destek Vektör Makineleri(SVM) kullanılarak Regresyon modelleri kurulmuştur. [2] Particle Swarm Optimization Algoritması Destek Vektör Makinelerinin parametrelerini optimize edip, etkili bir şekilde hisse senedi fiyatlarını tahmin etmek için kullanılmıştır. [3] Bu çalışma SVM metodunu destekler ama Particle Swarm Optimizasyonu uzun hesaplama zamanına ihtiyaç duyar. Marketin geri dönüş reflekslerini hesaplamak için LSTM ve saf Bayesian metodu birleştirilmiştir. [4] Duygu analizi ile LSTM zaman serisi modellerinin birleşiminden doğan bu model hisse açılış fiyatlarının tahmin doğruluk oranını arttırmayı başarmıştır. [5] Jia H. 'in [6] yapmış olduğu araştırma LSTM'in hisse fiyat tahmininde etkin bir metottur. Gerçek Zamanlı Dalgacık algoritması ile LSTM birleştirilerek Asya Borsa Fiyat Tahminlemesi yapılmıştır, ki bu araştırma sonucunda bu alanda yapılmış bazı mantıksal hataların fakedilmesine yol açmıştır. [7] Orijinal LSTM'e kıyasla bu model yüksek doğruluk oranı ve düşük resresyon hatası payı taşımaktadır. Çin Borsası Fiyat tahminlemesi için Bagging Metod kullanılmıştır, bu metod birçok neural network'u backpropagation metodu ve Adam Algoritması ile eğiterek her bir hisse senedi için farklı doğruluk oranına ulaşmıştır, ama tahmin tatmin edici doğruluğa ulaşamamıştır. [8] Revolutionary Metod'u hisse senetlerinin trendlerini tahmin etmek için kullanılmıştır. [9]Doğal plastikliğe sahip derin inanç ağı, hisse senedi zaman serilerini tahmin etmek için kullanıldı. [10]Convulational Neural Network, hisse senedi trendlerini tahmin amaçlı kullanıldı. [11]

Yukarıdaki ve daha birçok araştırma ve hisse senetlerinin parametre ve göstergelerinin birbiyle ilişkili olması göz önüne alınarak; bir hisse senedinin zaman bazlı değerlerini girdi olarak alıp makine öğrenmesi algoritmaları kullanarak tahmin edilmesine karar verildi. Bize ait Linear Regresyon modeli ve keras kütüphanesi LSTM bazlı bir RNN(Tekrarlayan Sinir Ağları) kullanarak borsa tahminlemesi yapmaya çalışacağız.

Öğretme işlemini yapacağımız bu modeli Apple hisse senetleri üzerinde uygulayacağız.

2. MATERYAL VE METOT (MATERIAL AND METHOD)

Bu çalışmanın amacı Linear Regresyon Algoritması ve LSTM modelini kullanarak hisse senedi tahmini etmek ve verileri karşılaştırmaktır. Bu kapsamda Apple hisse senedi geçmiş verilerini kullanarak borsa fiyatını tahmin etmeye yönelik çalışmalarda bulunduk.

2.1 Verilerin Toplanması, Temizlenmesi ve Düzenlemesi

Bu çalışmada kullanılan veriler, Python Yahoo-Finance API kullanılarak Wikipedia ve Yahoo Finance web sitesinden elde edilmiştir.

Firmalar Yahoo Finance API'inden indirildikten sonra Excel dosyası şeklinde son tarihine göre veri klasöründe kaydedilmektedir. Veriler en geçmiş tarihten en güncel tarihe göre sıralanmıştır. Son olarak Excel dokümanı işlem görmemiş günler için boş satırlardan arındırılarak hazır hale getirilmiştir.

2.2 Model Dizaynı

2.4.1 Linear Regression

Regresyon, bir bağımlı değişken (genellikle Y ile ifade edilir) ile bir dizi diğer değişken (bağımsız değişkenler -X- olarak bilinir) arasındaki ilişkinin gücünü belirlemeye çalışan finans, yatırım ve diğer disiplinlerde kullanılan istatistiksel bir ölçümdür.

Regresyon, yatırım ve finansal danışmanların varlıklara değer vermelerine ve emtia fiyatları ve bu emtialarda işlem yapan işletmelerin hisse senetleri gibi değişkenler arasındaki ilişkileri anlamalarına yardımcı olur. Daha karmaşık veri ve analiz için doğrusal olmayan regresyon yöntemleri olmasına rağmen iki temel regresyon türü öne çıkar: doğrusal regresyon ve çoklu doğrusal regresyon.

Doğrusal regresyon, Y bağımlı değişkeninin sonucunu açıklamak veya tahmin etmek için bir bağımsız değişken kullanırken, çoklu regresyon sonucu tahmin etmek için iki veya daha fazla bağımsız değişken kullanır.

Regresyon, finansmana ve yatırım profesyonellerine olduğu kadar diğer işletmelerdeki profesyonellere de yardımcı olabilir. Regresyon ayrıca hava durumu, satışlar, GSYİH büyümesi veya diğer koşullara bağlı olarak bir şirketin satışlarını tahmin etmeye yardımcı olabilir.

Sermaye varlıkları fiyatlandırma modeli (Capital Asset Pricing Model - CAPM), varlıkların fiyatlandırılması ve sermaye maliyetlerinin keşfedilmesinde finansmanda sıkça kullanılan bir regresyon modelidir.

$$y = \beta_0 + \beta_1 X + \varepsilon$$

Figure 1. Linear Regresyon Formülü

Y değeri x'e göre tahmin edilen değer, B₀ kesişim(x 0 iken y'nin tahmini değeri), B₁ regresyon katsayısı(x arttıkça y'nin ne kadar artacağı), x bağımsız değişken, e tahmin hatası(tahmin gerçek değerden ne kadar sapmıştır.)

$$\text{Error} = \sum_{i=1}^n (\text{actual_output} - \text{predicted_output}) ** 2$$

Figure 2. Hata hesaplama formülü

$$b_0 = \bar{y} - b_1 \bar{x}$$

Figure 3. Kesişim Hesaplama

$$b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Figure 4.Katsayı Hesaplama

b_0 ve b_1 değerleri hatayı minimize edecek şekilde seçilmelidir. Toplam hataların karesini bir değerlendirme ölçütü olarak kullanacağımız için, hedefimiz hatayı küçülten bir doğru seçmek olacaktır. Kullandığımız modelde b_0 slope, b_1 ise intercept olarak geçmektedir.

2.4.2 Tekrarlayan Sinir Ağları(RNN)

Tekrarlayan yapay sinir ağları (RNN), birimler arasındaki bağlantıların, yönlendirilmiş bir döngü oluşturduğu ağlardır. RNN ile dinamik zamansal davranış sergilemesine izin verilmektedir. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleklerini, girdileri işlemek için kullanabilirler. Bu öznelik RNN'leri, el yazısı tanıma ve konuşma tanımda, kullanılabilir bir yöntem yapmaktadır.

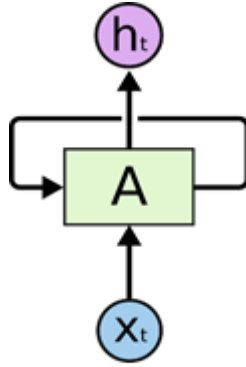


Figure 5.RNN içerisindeki döngü

Üstteki şekilde basit bir tekrarlayan sinir ağ görüntülenmektedir. 'A' ismi verilen dikdörtgen, bir yapay sinir ağındaki hücredir. Ağıın girdi değeri x_t 'dir. Yapay sinir ağıının çıktı değeri h_t 'dir. Hücreden çıkan bir değer yine kendisine gelerek, bir döngü oluşturmaktadır. Bu döngü ile önceki zamanın bilgileri de kullanılabilirdiğinden yeni bilgi, eski bilgi kullanılarak anlamlandırılabilmekte ve böylelikle sınıflandırma yapılabilir.

Geleneksel yapay sinir ağlarında, hücrelerden çıkan sonuçlar tekrardan kendilerine girdi olarak gelmemektedir. RNN'de ise hücreden çıkan sonuç, tekrardan kendisine girdi olarak gelmektedir. RNN açılırsa aşağıdaki şekildeki gibi bir mimari ortaya çıkmaktadır. Zaman diliminde, aynı hücre kendini birden fazla tekrar etmektedir. Böylelikle de kareler arasında anlamlandırma kurulabilmektedir.

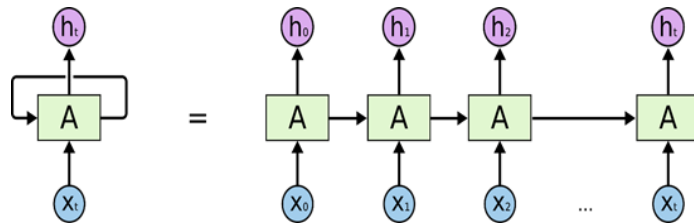


Figure 6. Açık hale getirilmiş bir RNN

RNN'ler, bir döngü oluşturabildiklerinden, zaman serisi gibi sıralı gelişen aktiviteleri birbirleriyle anlamlandırabilmektedir. Akış içerisindeki aktivitelerin anlamlandırılmasına sınıflandırabilmesinden dolayı

son yıllarda yaygın olarak kullanılmaktadır. RNN'lerin, birçok kullanım alanı bulunmaktadır; konuşma tanımlama, dil modelleme, dil çevirimi, resim başlığı oluşturma vb. [12]

Tekrarlayan Sinir Ağlarının Avantajları:

- Herhangi bir uzunluktaki girdilerin işlenmesi imkânı
- Girdi büyüklüğüyle artmayan model boyutu
- Geçmiş bilgileri dikkate alarak hesaplama
- Zaman içinde paylaşılan ağırlıklar

Tekrarlayan Sinir Ağlarının Dezavantajları:

- Yavaş hesaplama
- Uzun zaman önceki bilgiye erişme zorluğu
- Mevcut durum için gelecekteki herhangi bir girdinin düşünülmemesi [13].

LSTM

Uzun Kısa Süreli Bellek ağları, RNN'lerin uzun süreli hatırlama problemlerini çözmek amacıyla ortaya çıkmıştır. Genellikle LSTM olarak kısaltılmaktadır.

Geleneksel RNN'lerde, tekrarlayan kısımda tek bir tanjant fonksiyonu (katmanı) bulunmaktadır. LSTM'ler tek bir sinir katmanı yerine tekrarlayan zincir şeklinde dört farklı katman bulunmaktadır. Aşağıdaki şekilde, LSTM'de bulunan 4 farklı katman gösterilmektedir. LSTM mimarisinin görselleştirildiği resimlerde, birçok sembol bulunmaktadır. Bu semboller aşağıda gösterilmiştir.

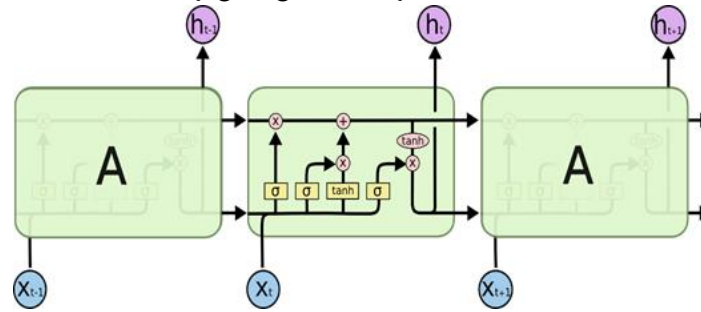


Figure 7.LSTM'deki dört etkileşimli katman

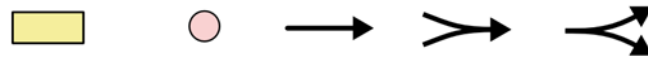


Figure 8.LSTM Mimarisinde kullanılan işaretler

Şekilde, okların çizgi kısmı düğümdeki çıkışları, baş kısımları ise diğer düğümlere girdi olarak veriyi taşımaktadır. Okun yönü, verinin taşındığı düğümü göstermektedir. Pembe daire, vektör eklemesi gibi noktasal operasyonları simgelemektedir. Sarı dikdörtgen, yapay sinir katmanını göstermektedir. İki okun birleşmesi, iki farklı vektörün birleşip, başka bir düğüme girdi oluşturmaktadır. Çatal şeklindeki ok ise düğümden çıkan bir vektörün, birden fazla düğüme girdi olarak kopyalanmakta olduğunu göstermektedir. LSTM'lerin kilit noktası, hücrenin durumudur (cell state). Bu hücre durumu şekildeki, hücrenin üst tarafında gösterilen yatay çizgidir. Hücrenin durumu, bir çeşit taşıma bandı gibidir. Bilgi, genellikle soldan sağa ('den 'ye) ilerlemektedir ancak bazı küçük doğrusal etkileşimlerde, zincirin alt tarafında doğru da ilerleyebilmektedir.

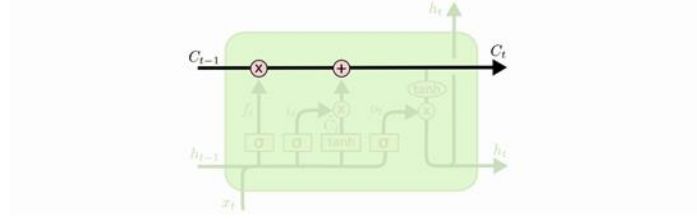


Figure 9. LSTM'lerin kilit noktası olan hücre durumu

LSTM'ler, kapılar (gate) yardımıyla, hücre durumundan, bilgi silme ve hücre durumuna bilgi ekleme yeteneğine sahiptir. Bu kapılar, isteğe bağlı olarak bilginin geçmesine olanak sağlamaktadır. Bu hücre, sigmoid sinir katmanı ve noktasal çarpma işleminden oluşmaktadır. Aşağıdaki şekilde bu işlem gösterilmektedir.

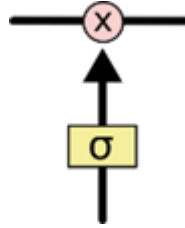


Figure 10. Noktasal çarpma ve sigmoid sinir kapısı

Sigmoid sinir katmanı, bilginin, ne kadar geçeceğini belirlemektedir. Sigmoid fonksiyonu, gelen veriye göre, çıktıda 0 ile 1 arasında bir sonuç çıkarmaktadır. Bir sigmoid fonksiyonundan, 0 çıkmışsa, “hiçbir şeyin geçmesine izin vermeyin”, eğer 1 sonucu çıkmışsa “hepsinin geçmesine izin verin” anlamındadır.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Figure 11. Sigmoid aktivasyon fonksiyonu formülü

LSTM ağları, bu kapılardan üç tanesine sahiptir. Bu kapıların görevi, korumak ve kontrol etmektir.

Kapılar(Gates)

LSTM ağlarının, sahip olduğu bazı kapılar vardır. Bu kapılar şu şekilde listelenir;

- Giriş Kapısı
- Unutma Kapısı
- Çıkış Kapısı

LSTM ağının ilk aşaması, unutma kapısı ile yapılmakta olan, hücre durumundaki bilginin, atılıp atılmayacağını (saklanıp saklanmayacağını) karar verilmesi aşamasıdır. Unutma kapısı, ve girdisi ile beslenip, hücre durumuna, 0 ile 1 arasında bir sayı vermektedir. Eğer 1 çıkarsa tamamen bunu tut, 0 çıkarsa bunu tutmana gerek yok anlamındadır.

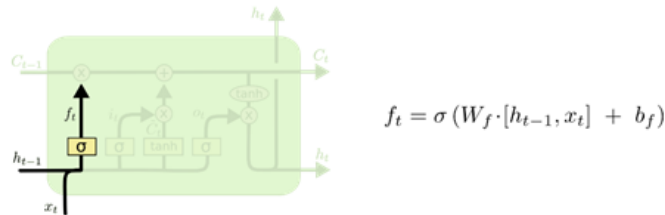


Figure 12. LSTM unutma kapısı ve formülü

LSTM'lerin bir sonraki aşaması, giriş kapısı ile yapılan hangi bilginin hücrede tutulacağı bilgisinin belirlenmesidir. Giriş kapısı, sigmoid sinir ağı ve tanjant katmanı olmak üzere ikiye ayrılmaktadır. Sigmoid

sinir katmanında, hangi değerlerin güncelleneceğine karar verilirken, tanjant katmanda ise saklanması gereken yeni vektör oluşturulmaktadır. Giriş katmanında, iki farklı değer birleşerek, yeni saklanacak olan değeri oluşturmaktadır.

Aşağıdaki şekilde giriş kapısı katmanı ve tanjant kapısı katmanı ile bu kapıların formülleri gösterilmiştir. Hücrenin eski değerini tutmakta olan C_{t-1} vektörü ile giriş kapısından çıkan sonuç olan C_t vektörlerinden hangisinin saklanacağına, unutma kapısı karar vermektedir. Unutma kapısından gelen f_t sonucu ile eski vektör çarpılmaktadır. Unutma kapısından 0 gelirse, çarpma işlemi ile beraber saklanan değer, saklama denkleminde hiç katılmayacaktır. Eğer 1 değeri gelmiş ise tamamı korunacaktır.

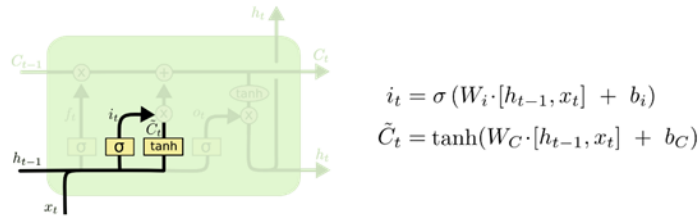


Figure 13. Giriş kapısı katmanı ve tanjant kapılarının formülü

Yeni saklanan değer ne olacağına, giriş katmanında tanjant katmanı ile karar verilmektedir. Giriş kapısından gelen vektör, sigmoid sinir katmanından gelen değer olan, i_t değeri ile tanjant katmanından gelen değer olan C_t 'nin çarpılması ile belirlenmektedir. Giriş kapısından çıkan vektör, bir önceki aşamada belirlenmiş olan ($f_t * C_{t-1}$) değeri ile toplanmaktadır.

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Figure 14. Hiperbolik tanjant aktivasyon fonksiyonu formülü

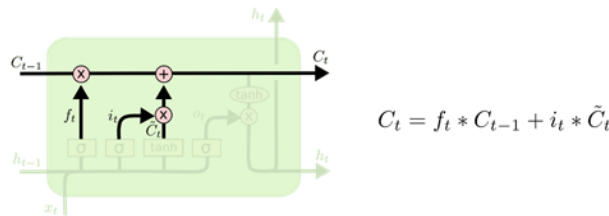


Figure 15. Hücre durumundaki eski ile yeni verinin güncellenmesi

Çıkış katmanı, tanjant ve sigmoid katmanı olmak üzere iki katmandan oluşmaktadır. Tanjant katmanında, -1 ile 1 arasında değer çıkmakta ve tanjant katmanı ile saklanan bilginin ne kadarının, kullanılacağı hesaplanmaktadır. Sigmoid katmanının çıktısı ise 0 ile 1 arasında ve bu katman yeni gelen bilginin, kullanılıp kullanılmayacağına karar veren katmandır.

Çıkış katmanı, sigmoid ve tanjant katmanlarından gelen değerlerini çarparak oluşan değeri çıktı değeri olarak belirlemektedir. Aşağıdaki şekilde, LSTM hücresinde bulunan çıkış katmanı ve formülü gösterilmiştir. [12].

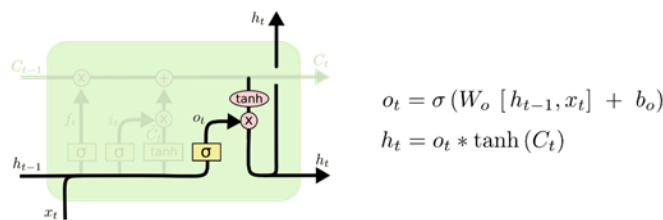


Figure 16. Çıkış kapısı ve formülleri

Loss Fonksiyonu

Loss fonksiyonu tasarlanan modelin hata oranını aynı zamanda başarımını ölçen fonksiyondur. Derin ağların son katmanı loss fonksiyonun tanımlandığı katmandır. Loss fonksiyonu, hata hesaplama işini problemi bir optimizasyon problemine dönüştürerek yaptığı için optimizasyon terminolojinde kullanılan objective function, cost function isimleriyle de tanımlanmaktadır. Loss fonksiyonu temelde modelin yaptığı tahminin, gerçek değerden (ground truth) ne kadar farklı olduğunu hesaplamaktadır. Bu nedenle iyi tahmin eden bir model oluşturmamışsak, gerçek değer (ground truth) ile tahmin edilen değer arasındaki fark yüksek olacak dolayısıyla loss değeri yüksek olacak, iyi modele sahipsek loss değeri az olacaktır. Birebir aynı olduğu durumda loss 0 olacaktır. Projede Kullanılan Loss fonksiyonu Huber Loss'tur.

Huber loss delta 0 yakın olunca MSE gibi hareket eder, delta büyüdükçe de MAE gibi bir çizgi takip eder. [14].

2.3 Değerlendirme Ölçütleri

2.4.1 Doğruluk Oranı(LSTM)

Doğruluk oranını hesaplamak için kullanılan bir fonksiyondur. Tahmin edilen veri setiyle, gerçekte olan değerlerin setini bir karşılaştırmasıdır.

Projemizde sk-learn modülünün accuracy_score fonksiyonu kullanılmıştır. Değer 1' yakınsa veri seti tahmini 100% doğruluk oranına sahiptir, aksi taktirde 0'a yakın olacaktır.

Eğer y_i^{\wedge} veri setinin i adımındaki tahmini ve y_i verinin i adımındaki gerçek değeri ise, $n_samples$ adımı bulunan bir veri setinin doğruluk oranı hesaplaması aşağıdaki gibi yapılır. [15].

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i)$$

Figure 17. LSTM için Doğruluk Oranı Hesaplaması

2.4.2 R² Skoru

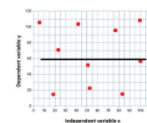
Belirleme katsayısı veya R², bir modelin uyumu hakkında bilgi sağlayan bir ölçüdür. Regresyon bağlamında, regresyon çizgisinin gerçek verilere ne kadar iyi yaklaştığının istatistiksel bir ölçüsüdür. Bu nedenle, gelecekteki sonuçları tahmin etmek için veya hipotezlerin test edilmesinde istatistiksel bir model kullanıldığında önemlidir[16].

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$
$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}.$$

Figure 18. R2 skoru formülü

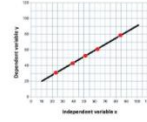
$$R^2 = 0$$

None of the variation in the y values
is accounted for by the x values



$$R^2 = 1$$

All the variation in the y values is accounted for by the x values



$$R^2 = 0.83$$

83% of the variation in the y values is accounted for by the x values

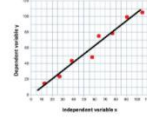


Figure 19. R2 skorları

3. ANALİZ VE UYGULAMA (ANALYSIS AND APPLICATION)

3.1 Linear Regresyon

Hazırladığımız modelde Apple firmasının 2020 mayıs ve temmuz arası fiyat verileri kullanılmıştır.

Çalışmada açılış değerleri X girdisi olarak kullanılmış, kapanış değerleri ise Y kümesini oluşturmuştur.

Yukarıda belirtilen formüller kullanarak b0 slope değeri 0.9482580952364014 olarak hesaplanmıştır. b1

İntercept değeri ise 6.1583252243522395 olarak hesaplanmıştır.

Çalışma sonunda elde edilen tahmin değerleri:

```
[130.53186047 147.03154554 143.01094047 122.31994189 119.8639569
144.86951824 135.79469579 130.01031562 140.66874182 141.7971568 ]
```

Figure 20. Linear Regresyon çalışma sonrası gerçek değerler

Gerçek değerler ise:

```
[131.16000366 148.55999756 144.32000732 122.5 119.91000366
146.27999878 136.71000671 130.61000061 141.8500061 143.03999329]
```

Figure 21. Linear Regresyon çalışma sonrası tahmini değerler

Olarak hesaplanmıştır.

$$y = \beta_0 + \beta_1 X + \varepsilon$$

Formülüne göre:

$0.9482580952364014 * 131.16000366 + 6.1583252243522395 = 130.53186047$ değerine eşittir.

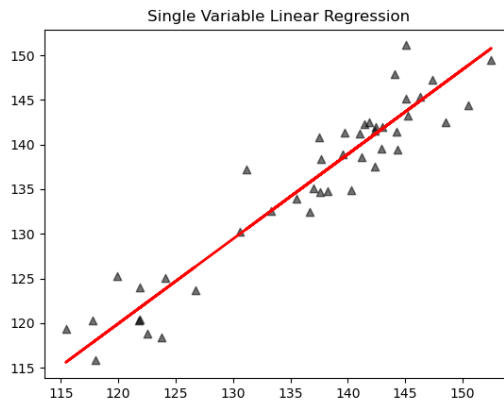


Figure 22. Linear Regresyon Apple firması grafiği

R^2 ölçüt değeri belirtilen firma ve tarih aralığında 0.8041136458498783 olarak saptanmıştır. MAE ise 0.90'dır.

3.2 LSTM

Hazırladığımız modelin test aşamasında Apple firması için birçok look_up steps, n_steps, dropout rate, loss fonksiyonu ve epoch parametresi denenmiştir.

```
# tahmin aşaması için kaç günün verilerinin kullanılacağı
N_STEPS = 250
# kaç gün sonrasının verileri tahmin edilecektir
LOOKUP_STEP = 1

# test yüzdesi
TEST_SIZE = 0.2
# hangi sütunlar kullanılacaktır
FEATURE_COLUMNS = ["adjclose", "volume", "open", "high", "low"]
# tarih ayarları
date_now = time.strftime("%Y-%m-%d")

### modelimiz için kullanılacak parametreler

N_LAYERS = 3
# RNN hücreleri(cells)
CELL = LSTM
# 256 LSTM nöronları
UNITS = 256
# 40% dropout
DROPOUT = 0.4
# Çift yönlülük
BIDIRECTIONAL = False

### Öğrenim parametreleri

# huber loss
LOSS = "huber_loss"
OPTIMIZER = "adam"
BATCH_SIZE = 64
EPOCHS = 500
```

Figure 23. LSTM hiperparametreleri

Aktivasyon fonksiyonu olarak Relu ve sigmoid fonksiyonu kullanılmış ve sigmoid fonksiyonunun reluya göre 10% daha etkili olduğu görülmüştür.

Sırasıyla her bir şirket için 1'er, 10'ar ve 30'ar günlük fiyat tahmini yapılmıştır. Uzun aralıklarda modelin daha başarılı sonuçlar verdiği gözlemlenmiştir.

Epoch sayısında yükseltmeye gidildiğinde doğruluk oranının arttığı fakat eğitim süresinin de arttığı gözlemlenmiştir. Sırasıyla her bir firmanın eğitim sonuçları:

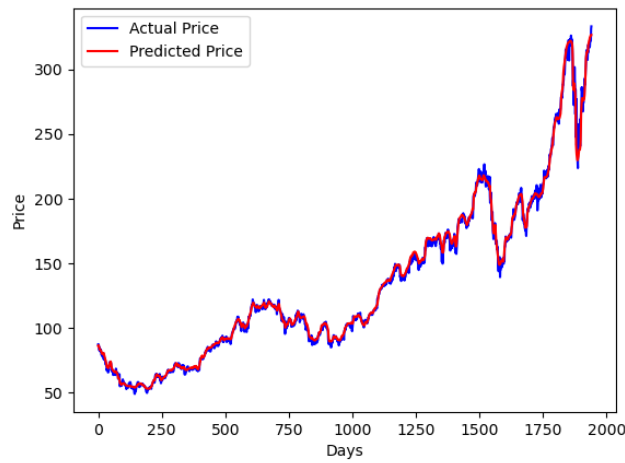


Figure 24. Apple 08.06.2020 tahmin tablosu

```

o such file or directory', flags = 0, flags = 0)
C:\Users\mear\Desktop\stock-prediction>py test.py
2020-06-09 00:42:40.395492: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2020-06-09 00:42:40.398589: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2020-06-09 00:42:44.177995: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library nvcuda.dll
2020-06-09 00:42:44.909927: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1561] Found device 0 with properties:
pciBusID: 0000:01:00:0 name: GeForce MX230 computeCapability: 6.1
coreClock: 1.531GHz coreCount: 2 deviceMemorySize: 2.00GiB deviceMemoryBandwidth: 44.76GiB/s
2020-06-09 00:42:44.913541: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2020-06-09 00:42:44.915840: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cublas64_10.dll'; dlerror: cublas64_10.dll not found
2020-06-09 00:42:44.918155: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2020-06-09 00:42:44.928317: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2020-06-09 00:42:44.931877: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusolver64_10.dll'; dlerror: cusolver64_10.dll not found
2020-06-09 00:42:44.940928: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cuspars64_10.dll'; dlerror: cuspars64_10.dll not found
2020-06-09 00:42:44.951179: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2020-06-09 00:42:44.959100: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1598] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-06-09 00:42:44.976457: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2020-06-09 00:42:44.985727: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1c46513740 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-06-09 00:42:44.989288: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2020-06-09 00:42:44.998648: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182] Device interconnect StreamExecutor with strength 1 edge matrix:
2020-06-09 00:42:45.000225: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1188]
Future price after 1 days is 326.83$
Accuracy Score: 0.6293814432989691
C:\Users\mear\Desktop\stock-prediction>

```

Figure 25. Apple 08.06.2020 tahmin doğruluk oranı

Yukarıda da görüldüğü gibi Apple hisse fiyatları 08.08.2020 tarihinde 1 gün sonraki fiyatın 326.83\$, 62%'lük bir doğruluk oranıyla tahmin etmiştir.

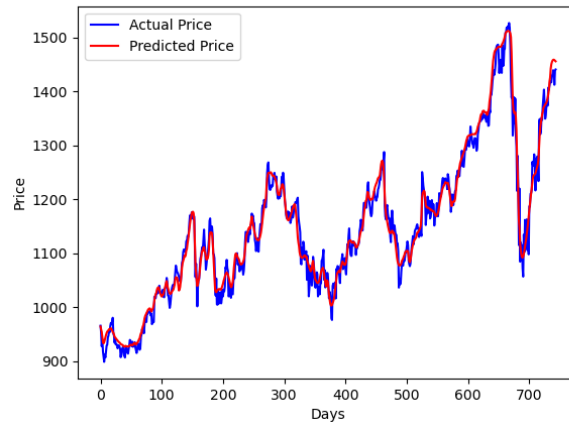


Figure 26. Apple 08.06.2020, 10 gün sonrası tahmini

```

coreClock: 1.531GHz coreCount: 2 deviceMemorySize: 2.00GiB deviceMemoryBandwidth: 44.76GiB/s
2020-06-09 01:45:51.873428: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2020-06-09 01:45:51.875819: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cublas64_10.dll'; dlerror: cublas64_10.dll not found
2020-06-09 01:45:51.878049: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2020-06-09 01:45:51.880254: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2020-06-09 01:45:51.882488: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cusolver64_10.dll'; dlerror: cusolver64_10.dll not found
2020-06-09 01:45:51.890713: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cuspars64_10.dll'; dlerror: cuspars64_10.dll not found
2020-06-09 01:45:51.893859: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2020-06-09 01:45:51.895796: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1598] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-06-09 01:45:51.907038: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2020-06-09 01:45:51.914926: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1f9af1b4960 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-06-09 01:45:51.918239: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2020-06-09 01:45:51.922887: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182] Device interconnect StreamExecutor with strength 1 edge matrix:
2020-06-09 01:45:51.924620: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1188]
Future price after 10 days is 323.11$
Accuracy Score: 0.8786936236391913

```

Figure 27. Apple 08.06.2020, 10 gün sonrası doğruluk oranı

Modelimizin uzun vadede eğitimler sayesinde daha iyi bir performans gösterdiğini söylemiştik. 10 gün sonrasının fiyatını 323.11 dolar, 87% doğruluk oranı ile tahmin ediyor.

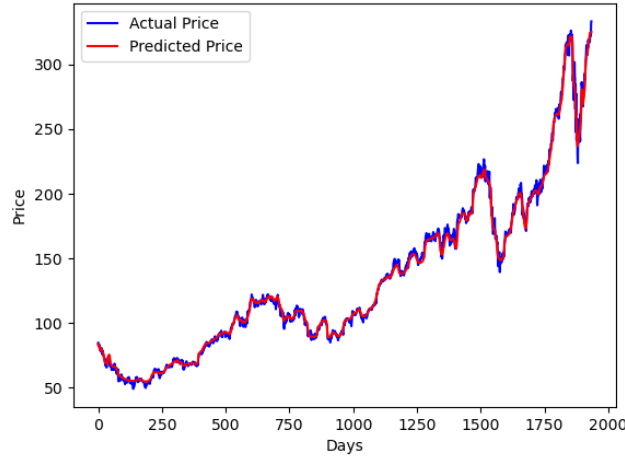


Figure 28. Apple 08.06.2020, 30 gün tahmini

```

ry 'cudart64_101.dll'; d1error: cudart64_101.dll not found
2020-06-09 11:15:54.109647: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'cublas64_10.dll'; d1error: cublas64_10.dll not found
2020-06-09 11:15:54.113232: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'cufft64_10.dll'; d1error: cufft64_10.dll not found
2020-06-09 11:15:54.117406: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'curand64_10.dll'; d1error: curand64_10.dll not found
2020-06-09 11:15:54.120859: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'cusolver64_10.dll'; d1error: cusolver64_10.dll not found
2020-06-09 11:15:54.122685: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'cusparse64_10.dll'; d1error: cusparse64_10.dll not found
2020-06-09 11:15:54.125304: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic libra
ry 'cudnn64_7.dll'; d1error: cudnn64_7.dll not found
2020-06-09 11:15:54.128206: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1598] Cannot dlopen some GPU libraries. P
lease make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the gu
ide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-06-09 11:15:54.134657: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that thi
s TensorFlow binary was not compiled to use: AVX2
2020-06-09 11:15:54.143305: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x19d2b2e5de0 initialized for
platform Host (this does not guarantee that XLA will be used). Devices:
2020-06-09 11:15:54.145599: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default
Version
2020-06-09 11:15:54.148349: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182] Device interconnect StreamExecutor
with strength 1 edge matrix:
2020-06-09 11:15:54.150957: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1188]
Future price after 30 days is 325.23$
Accuracy Score: 0.9322834645669291
C:\Users\mswr\Desktop\195112046 - Musawer Muradi - Stock Price Prediction using LSTM\stock-prediction>

```

Figure 29. Apple 08.06.2020, 30 gün doğruluk oranı

Model 30 gün sonrasının fiyatını 93% doğruluk oranı ile 325.23 dolar hesaplıyor.

4. SONUÇ (CONCLUSION)

Bu alanda birçok çalışma yapılmış ve yapılmaya devam edilmiş olmasına rağmen, en genel iki piyasa tahmin metodlarını inceledik.

Kendi geliştirmiş olduğumuz basit Linear Regreson sonucu b0 slope değeri 0.9482580952364014 olarak hesaplanmıştır. b1 İntercept değeri ise 6.1583252243522395 olarak hesaplanmıştır. Bu kapsamda Apple firmasının 10 günlük ileriye dönük tahminleri yapılmıştır. R^2 değeri 81% olarak hesaplanmıştır. MAE ise 0.90'dır

LSTM'de toplam 19910 verinin kullanıldığı Apple stock'unda 1 gün sonrası tahmini için MAE 3.7 ve doğruluk oranı 63% , 10 gün sonrası için MAE 3.8 ve doğruluk oranı 88%, 30 gün sonrası için ise MAE 2.0 ve doğruluk oranı 93% olarak karşımıza çıkmıştır.

Bu kapsamda keras kütüphanesinin de vermiş olduğu esneklik sayesinde LSTM modelinin uzun vadede başarılı sonuçlar verdiğini gözlemledik.

KAYNAKLAR

- [1] B. ATASEVEN, "YAPAY SİNİR AĞLARI İLE ÖNGÖRÜ MODELLEMESİ," *Öneri Dergisi*, vol. 10, no. 39, 2013.
- [2] "Rachael, Olomo & Misra, Sanjay & Ahuja, Ravin & Adewumi, Adewole & Ayeni, Foluso & Maskeliunas, Rytis. (2019). Image Steganography and Steganalysis Based on Least Significant Bit (LSB). 10.1007/978-3-030-30577-2_97."
- [3] Y. Xia, Y. Liu and Z. Chen, "Support vector regression for prediction of stock trend," in *6th Int Conf Inf Manag Innov Manag Ind Eng (ICIII)*, 2013.
- [4] T. Sands, D. Tayal and S. Monteiro, "Robust stock value prediction using support vector machines with particle swarm optimization," in *Congress on Evolutionary Computation (CEC)*, 2015.
- [5] J. Li, H. Bu and J. Wu, "Sentiment-aware stock market prediction: a deep learning metho," in *IEEE: 2017 international conference on service systems and service management*, 2017.
- [6] Q. Zhuge, L. Xu and G. Zhang, "LSTM neural network with emotional analysis for prediction of stock price," *Eng Letter*, p. 2(25):167–175, 2017.
- [7] H. Jia, "Investigation into the effectiveness of long short term memory networks for stock price prediction," *arXiv[cs.NE]*, pp. 1-6.
- [8] Z. Li and V. Tam, "Combining the real-time wavelet denoising and long- short-term-memory neural network for predicting stock indexes.," in *IEEE: 2017 IEEE symposium series on computational intelligence (SSCI)*, 2017.
- [9] B. Yang, Z.-J. Cong and W. Yang, "Stock market index prediction using deep neural network ensemble," in *36th Chinese Control Conference (CCC)*, 2017.
- [10] Y.-C. Tsai and C.-Y. Hong, "The application of evolutionary approach for stock trend awareness," in *IEEE: IEEE 8th international conference on awareness science and technology (iCAST)*, 2017.
- [11] X. Li, L. Yang, F. Xue and H. Zhou, "Time series prediction of stock price using deep belief networks with Intrinsic plasticity," in *IEEE: 2017 29th Chinese Control and Decision Conference (CCDC)*, 2017.
- [12] M. Gudelek, S. Boluk and M. Ozabyoglu, "A deep learning based stock trading model with 2-D CNN trend detection," in *IEEE: 2017 IEEE symposium series on computational intelligence (SSCI)*, 2017.
- [13] J. Olah, 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [14] A. Amidi, "Stanford University," [Online]. Available: <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-recurrent-neural-networks#attention>.
- [15] P. Grover. [Online]. Available: <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>.

[16] "Scikit-Learn," 2019. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html.