

Derin Öğrenme Mimarisi Kullanarak BİST30 İndeksinin Tahmini

Şakir SAKARYA¹

ÜMİT YILMAZ²

Manuscript information:

Received: April 20, 2019

Revised: August 13, 2019

Accepted: September 07, 2019

Öz

Derin öğrenme yöntemleri, süreci hızlandırmak ve işlem adımlarındaki doğruluğu sağlamak amacıyla, verilerdeki karmaşık kalıpları ve veriler arasındaki etkileşimleri otomatik olarak algılayıp analiz edebilmektedir. Derin öğrenme yöntemlerinin finans alanında uygulanması, bilgiye mümkün olduğunca hızlı ve doğru bir şekilde ulaşma ihtiyacını karşılama noktasında faydalı olacaktır. Ayrıca bu yöntemlerin kullanımı sayesinde, karmaşık ve etkileşimli büyük veri kümelerini bünyesinde barındıran, menkul kıymetlerin tasarlanması ve fiyatlandırılması, optimal portföyün oluşturulması ve finansal risk yönetiminin gerçekleştirilmesi gibi finansal tahmin problemlerinin çözümü de kolaylaşacaktır. Bu çalışma, derin öğrenme mimarisi yardımıyla BİST 30 Endeksinin günlük hareket tahminini elde etmeyi amaçlamaktadır.

Author 1

Prof. Dr., İktisadi ve İdari
Bilimler Fakültesi, İşletme
Bölümü, Balıkesir
Üniversitesi, TÜRKİYE.
E-mail:
sakarya@balikesir.edu.tr

Anahtar Kelimeler: Derin Öğrenme, Borsa, BİST30

Author 2

Öğr. Gör., Bigadiç Meslek
Yüksekokulu, Yönetim ve
Organizasyon Bölümü
Balıkesir Üniversitesi,
TÜRKİYE. E-mail:
umityilmaz@balikesir.edu.tr

Cite as:

Sakarya, Ş. & Yılmaz, U.(2019). Derin öğrenme mimarisi kullanarak BİST30 indeksinin tahmini. *European Journal of Educational and Social Sciences*, 4 (2), 106 – 121.



Abstract

Deep learning methods can automatically detect and analyze to complex patterns of data and interactions between data in order to expedite to process and ensure accuracy in the processing steps. Implementation of deep learning methods in the finance area will be useful in meeting the need to reach information as quickly and accurately as possible. Furthermore, through the use of these methods, the solution of financial forecasting problems such as the design and pricing of mutual funds, the creation of the optimal portfolio and the realization of financial risk management, which involve big and complex data sets, will be facilitated. This study aims to obtain the daily movement forecast of XU30 Index with the deep learning architecture.

Keywords: Deep Learning, Stock Market, XU30

1. GİRİŞ

Borsada yatırım yaparak veya alım satım yapılarak kazanılan karlar büyük ölçüde öngörülebilirliğe bağlıdır. Hisse fiyatlarının neredeyse anlık olarak değiştiği dinamik borsaların indekslerini tutarlı bir şekilde tahmin edebilecek sistemlerin varlığı, yatırımcılara bilinçli kararlar verme noktasında kolaylıklar sağlayacak, hisse senedi fiyatlarını öngörmelerindeki belirsizlik ve yanlışlığın giderilmesinde yol gösterici olacaktır. Ayrıca bu sistemlerin varlığı borsa yatırımcılarının kararlarındaki şüpheli ve tutucu tutumlarını da terk etmelerine katkıda bulunacaktır (Gurjar, Naik, Mujumdar, ve Vaidya, 2018, s. 2758; Ugwu ve OnwuachuUzochukwu, 2014, s. 112). Sistem teorisi açısından, hisse senetlerinin ve dolayısıyla borsa indekslerinin oluşum mekanizmasının yüksek derecede karmaşıklığa sahip doğrusal olmayan bir yapıya sahip olması, tahmin edilmesini de bir hayli zorlaştırmaktadır (Wang, Hu, ve Wu, 2018, s. 1).

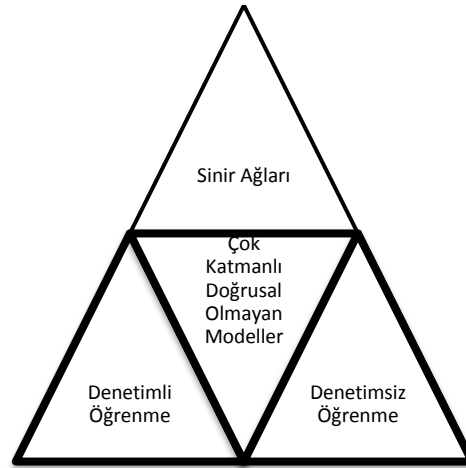
Günümüzde tahmin yapmak için genellikle üç analitik yaklaşımdan faydalanılmaktadır. Bunlardan ilki olan teknik analiz, bir finansal zaman serisinin gelecekteki davranışının geçmişteki hareketlerine bağlı olduğu fikrine dayanmaktadır. Yani teknik analiz, grafik analizi yoluyla gelecekteki hisse senedi fiyatını tahmin etmenin bir yoludur. Geçmiş n veri noktalarının ağırlıksız ortalaması olarak düşünülebilir. Bu yöntem kısa vadeli tahminler için uygundur. İkinci yaklaşım olan temel analiz ise şirketin finansal raporları ile siyasi ve ekonomik faktörler gibi dışsal bilgilere dayanmaktadır. Bahsi geçen bilgiler, gazete köşe yazıları gibi yapılandırılmamış verilerden, finansal raporlardan ve hatta analistler tarafından yayınlanan mikrobloğlardan alınmaktadır. Üçüncü yöntem ise zaman serisi verilerinin analizidir. Bu yöntem temel olarak doğrusal modeller ve doğrusal olmayan modeller biçiminde iki algoritma sınıfı içermektedir. Doğrusal modeller AR, ARMA, ARIMA ve bunların çeşitleri; doğrusal olmayan modeller ise ARCH, GARCH, TAR ve derin öğrenme gibi yöntemleri içermektedir (H. Liu, 2018, s. 2; Selvin, Vinayakumar, Gopalakrishnan, Menon, ve Soman, 2017, s. 1643; Sim, Kim, ve Ahn, 2019, s. 1).

Son yıllarda, piyasanın hareketlerini öngörebilme arzusuyla derin öğrenme teknikleri menkul kıymetler borsası tahmin problemlerinde yoğun bir şekilde kullanılmaktadır. Derin öğrenme, bir yapay sinir ağı için giriş ve çıkış katmanları arasında çoklu gizli katmanları olan genel bir terimdir (Chung ve Shin, 2018, s. 3766). Derin öğrenmenin bu tahminlerde kullanılmasının başında insanlar veya diğer geleneksel bilgisayarlar tarafından algılanamayacak kadar karmaşık ve/veya tahmini bilgilerden bağlam çıkarma konusunda çarpıcı bir kapasiteye sahip olmasındandır.

2. DERİN ÖĞRENME

Derin öğrenme, sinirsel ağların programlamasında nispeten yeni bir gelişmedir ve derin sinir ağlarının eğitilmesini sağlar. Derin sinirsel ağların varlığı, Pitts (1943)'in ortaya attığı çok katmanlı algılayıcılar kavramına dayanmaktadır. Ancak bu karmaşık sinir ağlarını başarılı bir şekilde eğiten ilk araştırmacı Hilton (1984)'dur (J. Heaton, 2015, s. 166). 2000'li yıllarda sinir ağlarının yeniden canlandırılmasıyla birlikte derin öğrenme, modern makine öğrenmesinin önünü açan son derece aktif bir çalışma alanı haline gelmiştir (Buduma ve Locascio, 2017, s. viii).

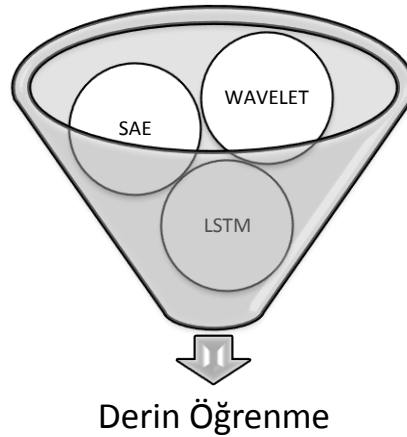
Derin öğrenme; sinir ağları, yapay zekâ, grafik modelleme, optimizasyon, örüntü tanıma ve sinyal işleme faaliyetlerinin kesişiminden ortaya çıkan bir makine öğrenimi alanıdır. Derin öğrenme ağları, sinir ağlarının devrim niteliğindeki bir gelişimidir ve daha güçlü tahminler elde etme amacıyla kullanılmaktadır. Derin öğrenme, çok katmanlı makine öğrenme modelleri kullanılarak verilerden denetimli ya da denetimsiz öğrenme ile ilgilidir. Bu modellerdeki katmanlar, verilerin özelliklerinin art arda daha yüksek, daha soyut katmanlarda temsil edildiği doğrusal olmayan veri dönüşümlerinin çoklu aşamalarından oluşmaktadır. Bu bilgilerden hareketle derin öğrenme, temel olarak Şekil 1’deki derin öğrenme piramidi şeklinde özetlenebilir (Lewis, 2016, s. 6):



Şekil 1. Derin Öğrenme Piramidi

Derin öğrenme, çok sayıda gizli katmana sahip gelişmiş bir sinirsel ağdan oluşmaktadır. Derin öğrenme, görüntü tanıma, bilgisayarla görme, el yazısı algılama, metin sınıflandırma, çoklu sınıflandırma ve regresyon problemleri gibi çeşitli uygulamalarda kullanılmaktadır (Ciaburro ve Venkateswaran, 2017, s. 38).

Çalışmada Şekil 2’de bahsi geçen yöntemlerin birbirlerine entegrasyonu sağlanarak tahmin doğruluğunun artması hedeflenmiştir.



Şekil 2. Çalışmada Derin Öğrenme Faaliyeti İçin Kullanılan Yöntemler

2.1. Dalgacık (Wavelet) Dönüşümü

Bir fiziksel büyüklüğün bir veya birden çok bağımsız değişkene göre durumunu izah eden gözlemler topluluğuna zaman serisi denir. Zaman serilerinde bağımsız değişken genellikle zamandır. Matematiksel olarak tanımlandığında (t) ye bağlı bir $g(t)$ fonksiyonu şeklinde tanımlanır. Veriler sürekli bir şekilde kaydediliyorsa elde edilen seriler sürekli seriler, belirli aralıklarda kaydediliyorsa ayrık seriler adını alır. Sürekli serilerdeki veri boyutunun fazla olması nedeniyle, içinden örnekleme yapılarak ayrık diziler oluşturulur. Günümüzde, birçok fiziksel büyüklük sürekli ve ayrık sinyal olarak gösterilmekte ve bu sinyallerin analizinde yaygın olarak dalgacık analizi kullanılmaktadır (Öner, Yeşilyurt, ve Yılmaz, 2017, s. 43).

Tanım olarak, bir dalgacık, ortalama değeri sıfır olan ve zamanla sınırlı bir dalga şeklindedir. Zaman ekseninde kaydırma ve ölçekleme parametreleri dalgacıkların temelini oluşturmaktadır (Canyılmaz, Türk, ve Güzel, 2016, s. 74). Dalgacık dönüşümü, verilerin hiyerarşik bir fonksiyonlar dizisine ayrıştırılması yöntemidir. Burada belirtilen her fonksiyon, yaklaşık olarak verileri açıklamaktadır (Derbeko, Dolev, ve Gudes, 2018, s. 109). Diğer bir ifade ile dalgacık dönüşümü, kompleks fonksiyonların, temel dalgacık olarak isimlendirilecek prototip fonksiyonun ölçeklenme ve ötelenmesiyle elde edilecek daha basit fonksiyonları birleşimi şeklinde ifade edilebilir (Aslan, Badem, Özcan, Karaboğa, ve Baştürk, 2015, s. 13).

Dalgacık dönüşümü, zaman serilerindeki yerel varyasyonları analiz ederek çeşitli veri kaynaklarından bilgi elde etmek için kullanılacak matematiksel bir araçtır. Dalgacık dönüşümü, modelleme performansı üzerinde önemli derecede olumlu etkiler yaratmaktadır. Dalgacık dönüşümleri, veri kalitesini iyileştirmek için orijinal bir zaman serisinin güvenilir şekilde ayrışmasını desteklemektedir (Mosavi, Ozturk, ve Chau, 2018, s. 20).

Dalgacık dönüşümünün sürekli dalgacık dönüşümü ve ayrık dalgacık dönüşümü olmak üzere iki türü vardır. Sürekli dalgacık dönüşümünde ölçeklendirme ve dönüşüm parametreleri sürekli değiştiğinden her bir ölçek için dalgacık katsayılarının hesaplanması zor ve zaman alıcıdır. Bu nedenle daha çok ayrık dalgacık dönüşümü kullanılmaktadır (Toprak, Çağlar, ve Merdan, 2007). Ele alınan çalışmada da ayrık dalgacık dönüşümü dikkate alınacaktır.

Normalleştirme kurallarına bağlı olarak, belirli bir fonksiyon/aile, baba dalgacık ve ana dalgacık olarak adlandırılan iki tür dalgacıktan oluşmaktadır. Baba dalgacıklar bir sinyalin düz ve düşük frekanslı kısımlarını, ana dalgacıklar ise detaylı ve yüksek frekanslı kısımlarını betimlemektedir. Aşağıdaki denklemlerde Denklem 1, baba dalgacı, Denklem 2 ise ana dalgacı temsil etmektedir (Hsieh, Hsiao, ve Yeh, 2011, s. 2512).

$$\Phi_{j,k} = 2^{-\frac{j}{2}} \Phi\left(\frac{t-2^j k}{2^j}\right) \quad (1)$$

$$\int \Phi(t) dt = 1$$

$$\Psi_{j,k} = 2^{-\frac{j}{2}} \Psi\left(\frac{t-2^j k}{2^j}\right); j = 1 \dots J \quad (2)$$

$$\int \Psi(t) dt = 0$$

Matematiksel gelişimi ve analizi kolaylaştırmak için, yukarıdaki ifadeler s ölçek parametresini 2^j diyadik ölçek ile sınırlandırmıştır. Bu temel fonksiyonların ailesi göz önüne alındığında, incelenen fonksiyonun öngörülen temel üzerine gösterimi için bir dizi katsayı tanımlanabilir. Bu katsayılar Denklem 3 ve Denklem 4'te gösterilmiştir (Ramsey, 2002, ss. 4-5):

$$s_{j,k} = \int f(t) \Phi_{j,k} \quad (3)$$

$$d_{j,k} = \int f(t) \Psi_{j,k}; j = 1 \dots J \quad (4)$$

Denklem 3'te gösterilen $s_{j,k}$ “düzgün katsayılar” olarak bilinen 2^j maksimal ölçekte baba dalgacığının katsayılarıdır. Denklem 4'te yer alan $d_{j,k}$ ise 1 ile J arasındaki tüm ölçeklerde ana dalgacıktan elde edilen detay katsayılarıdır. Katsayılar göz önüne alındığında f fonksiyonu Denklem 5'teki gibi formüle edilebilir:

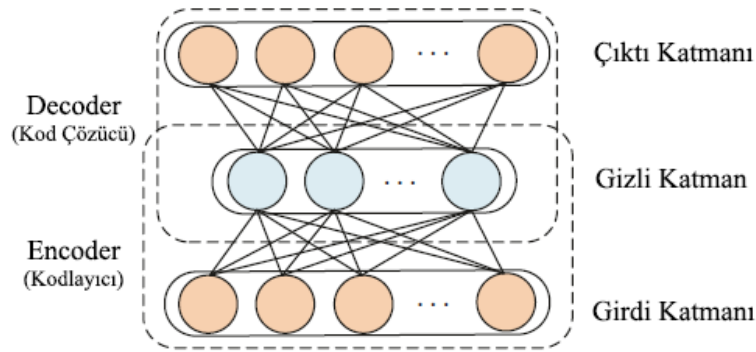
$$f(t) = \sum_k s_{j,k} \Phi_{j,k}(t) + \sum_k d_{j,k} \Psi_{j,k}(t) + \dots + \sum_k d_{j,k} \Psi_{j,k}(t) + \dots + \sum_k d_{1,k} \Psi_{1,k}(t) \quad (5)$$

$f(t)$ fonksiyonu ise $S_j = \sum_k s_{j,k} \Phi_{j,k}(t)$ ve $D_j = \sum_k d_{j,k} \Psi_{j,k}(t)$ olmak üzere Denklem 6'daki gibi özet bir şekilde gösterilebilir:

$$f(t) = S_j + D_j + D_{j-1} + \dots + D_1 \quad (6)$$

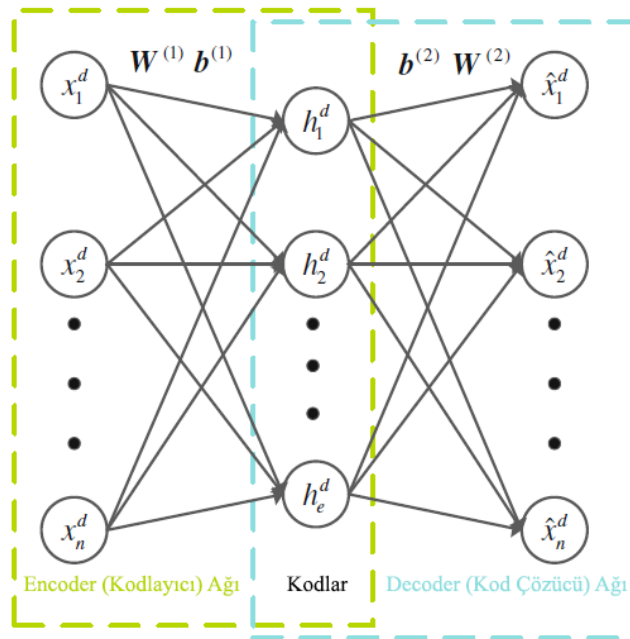
2.2. Derin (Yığın) Otokodlayıcılar

Oto kodlayıcılar, yapay sinir ağı modelinden türetilmiş denetimsiz bir makine öğrenme yöntemidir. Bir oto kodlayıcı, ileri beslemeli bir sinir ağı olarak giriş katmanı, gizli katman ve çıktı katmanı olmak üzere üç katmana sahiptir. Oto kodlayıcılar, temelde iki kısımdan oluşurlar. Birinci kısım, kodlayıcı (encoder) olarak adlandırılan ve gizli katmana giren ağırlıklar, ikinci kısım ise kod çözücü (decoder) olarak adlandırılan ve gizli katmandan çıkan ağırlıklardır. Oto kodlayıcı modelinde girdi katmanındaki nöron sayısı genellikle gizli katmandaki nöron sayısından daha fazladır. Bu modeli ileri beslemeli yapay sinir ağlarından ayıran en önemli ikinci özellik ise giriş veri setiyle çıkış veri setinin aynı olması dolayısıyla çıktı katmanındaki nöron sayısının girdi katmanındaki nöron sayısına eşit olmasıdır (Çalışan ve Talu, 2017, s. 2; Kannadasan, Edla, ve Kuppili, 2018, s. 2; Kaynar, Görmez, ve Işık, 2016, s. 47). Oto kodlayıcı, ağıın çıkışında girdi veri setinin aynısını kullandığından dolayı veri setindeki sınıf bilgisi içeren etiketlere ihtiyaç duymaz, yani denetimsiz bir makine öğrenmesi yöntemidir. Ağ, giriş veri setini çıkışta verilen aynı veri setine uydurmak amacıyla eğitim sırasında geri yayılım algoritmasını kullanarak en uygun ağırlık değerlerini belirler. Bu nedenle yöntem öğreticisi olmayan geri yayılım algoritması olarak da anılmaktadır (Kaynar, Aydın, ve Görmez, 2017, s. 323). Şekil 3'te genel bir oto kodlayıcı yapısı gösterilmiştir (G. Liu, Bao, ve Han, 2018, s. 3):



Şekil 3. Oto Kodlayıcı Yapısı

Oto kodlayıcının detaylı yapısı ise Şekil 4'te gösterilmiştir.



Şekil 4. Oto Kodlayıcının Detaylı Yapısı

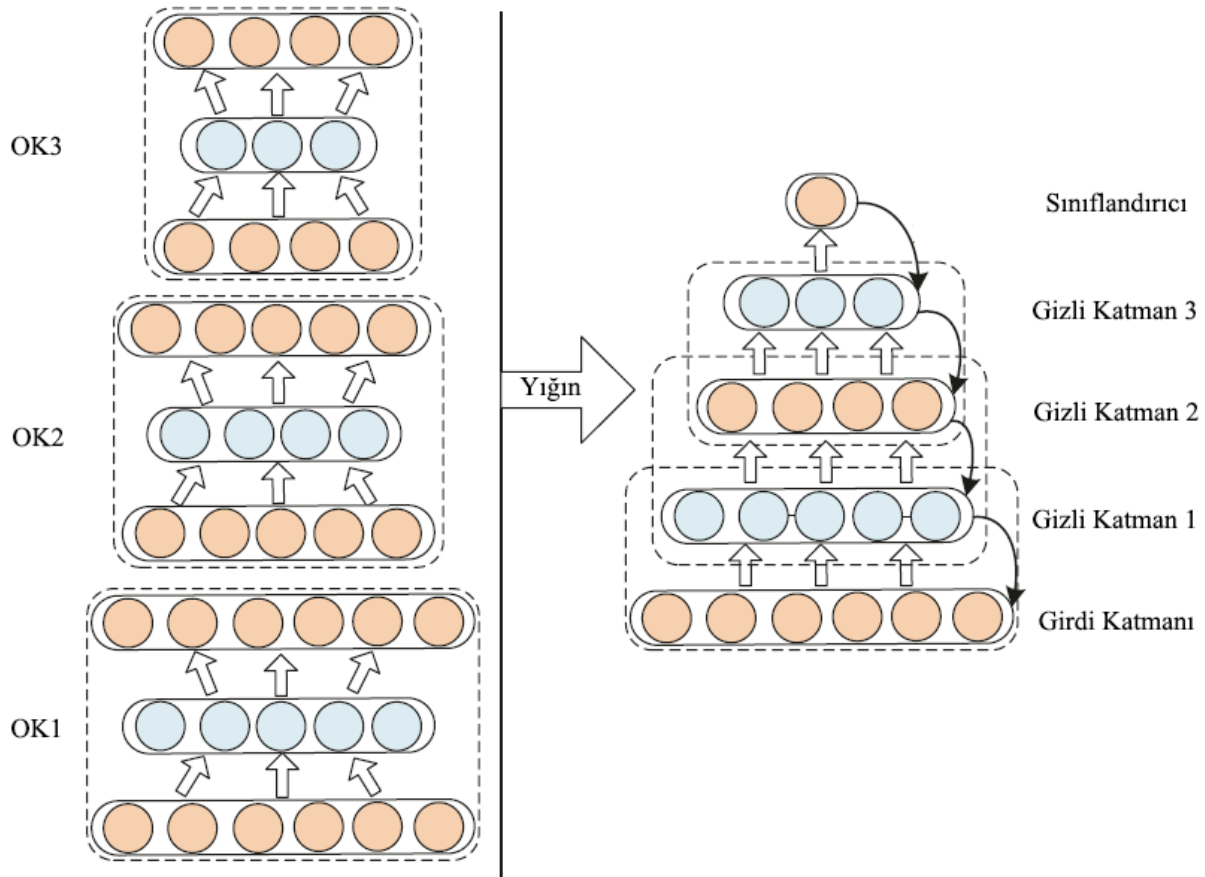
$W^{(1)}$ ile $W^{(2)}$ 'nin ağırlık matrisi, $b^{(1)}$ ile $b^{(2)}$ 'nin bias vektörü, f 'ninse aktif fonksiyon olduğu $\{x^d\}_{d=1}^M$ veri kümelerindeki her bir x^d girdi vektörü için gizli vektör h^d ve yeniden oluşturulmuş vektör \hat{x}^d olarak gösterilmiş ve Denklem 7'deki gibi tanımlanmıştır. $J(W, b)$ genel maliyet fonksiyonu Denklem 8'deki gibi tanımlanabilir. Burada λ ağırlık bozulma parametresini, n_l katman sayısını, S_l l katmanındaki nöronların sayısını, $W_{ji}^{(l)}$ ise sırasıyla $l+1$ ve l katmanlarındaki i ve j birimleri arasındaki bağlantı ağırlığını göstermektedir (Qi ve diğ., 2017, s. 369).

$$\mathbf{h}^d = f(\mathbf{W}^{(1)}\mathbf{x}^d + \mathbf{b}^{(1)}) \quad (7)$$

$$\hat{\mathbf{x}}^d = f(\mathbf{W}^{(2)}\mathbf{h}^d + \mathbf{b}^{(2)})$$

$$J(\mathbf{W}, \mathbf{b}) = \left[\frac{1}{M} \sum_{d=1}^M \left(\frac{1}{2} \|\mathbf{x}^d - \hat{\mathbf{x}}^d\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} \left(\mathbf{W}_{ji}^{(l)} \right)^2 \quad (8)$$

Derin oto kodlayıcılar, birden fazla gizli katmanlı oto ilişkili (öz-ilişkili) ağlardır. Genel olarak, tek katmanlı bir oto kodlayıcı, ayırt edici ve ham verileri temsil edici özelliğe sahip değildir. Bu nedenle derin oto kodlayıcılar, diğer bir adıyla yığın oto kodlayıcılar kavramı öne sürülmüştür. Daha fazla katman eklemek, oto kodlayıcının daha karmaşık kodlamalar öğrenmesine yardımcı olmaktadır. Ancak oto kodlayıcının çok uzmanlaşmamasına dikkat edilmelidir. Eğer oto kodlayıcı çok uzmanlaşırsa, böyle bir oto kodlayıcı eğitim verilerini mükemmel bir şekilde yeniden yapılandıracaktır, ancak süreçte herhangi bir faydalı genelleştirici veri meydana getiremeyecek ve yeni durumlara genelleme olasılığı düşük olacaktır (Wani, Bhat, Afzal, ve Khan, 2020, s. 88).



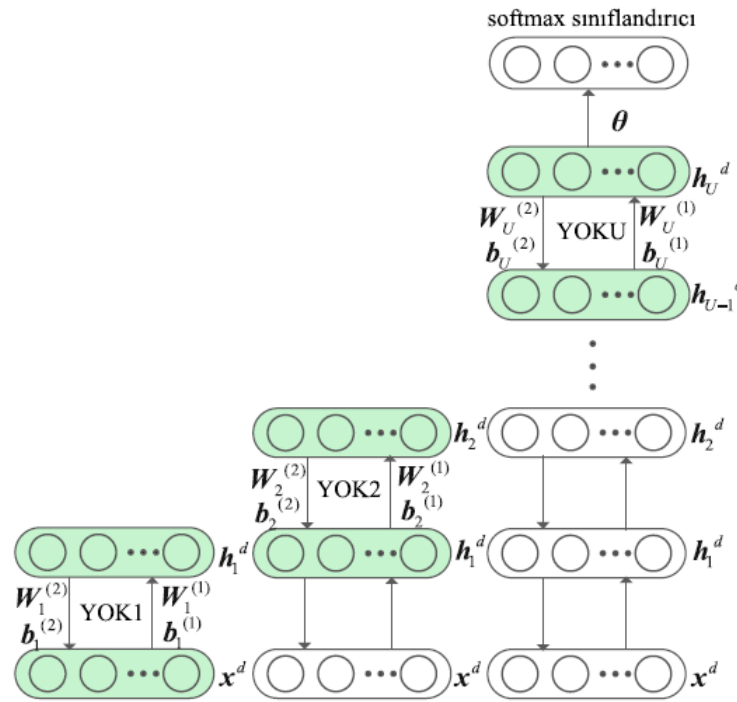
Şekil 5. Yığın Oto Kodlayıcı Yapısı

Yığın oto kodlayıcı, oto kodlayıcının $J(\mathbf{W}, \mathbf{b})$ maliyet fonksiyonuna bir seyreklik cezası uygulanarak elde edilebilir. Yığın oto kodlayıcının genel maliyet fonksiyonu Denklem 9'daki gibi

gösterilebilir. Burada ρ seyreklik parametresini, $\hat{\rho}_g$ g gizli biriminin ortalama çıktısını, β ise seyreklik ceza parametresini temsil etmektedir.

$$J_{yığın}(\mathbf{W}, \mathbf{b}) = J(\mathbf{W}, \mathbf{b}) + \beta \sum_{g=1}^{S_2} \left(\rho \log \frac{\rho}{\hat{\rho}_g} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_g} \right) \quad (9)$$

Yığın oto kodlayıcılarının eğitim süreci, orijinal girdilerin daha seyrek ve temsili özelliklerini otomatik olarak öğrenmeyi amaçlar. Parametrelerin başlatılmasından sonra, yığın oto kodlayıcının genel maliyet fonksiyonunu minimize etmek ve maksimum iterasyona ulaşana kadar parametreleri güncellemek için geri yayılma algoritması kullanılır. Optimal parametre kümeleri $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{b}^{(1)}$ ve $\mathbf{b}^{(2)}$ eğitim sürecinde aynı anda eğitilebilir ve iyi eğitilmiş bir yığın oto kodlayıcı elde edilebilir.



Şekil 6. Yığın Oto Kodlayıcının Detaylı Yapısı

U yığın oto kodlayıcılar Şekil 6'da gösterildiği gibi U adet gizli katmanlı bir yığın oto kodlayıcı ağını oluşturmak için yığınlanmaktadır. Softmax sınıflandırıcısı, sınıflandırma işlemi için çıkış katmanına eklenir. Yığın oto kodlayıcılarının eğitim süreci, ön eğitim ve ince ayar olmak üzere iki ana prosedürden oluşmaktadır. Ön eğitim, her yığın oto kodlayıcısını ve softmax sınıflandırıcısını, her bir modelin iç ilişkisini öğrenmek ve bir dizi optimal ağırlık başlatma işlemini öğrenmeyi amaçlarken; ince ayar ise katmanlar arasındaki ilişkiyi öğrenmeyi ve öğrenilen özellikler ve e etiketler arasında en uygun ilişkiyi kurmayı amaçlamaktadır. İlk olarak geri yayılım algoritması, U. Yığın oto kodlayıcısının (YOKU) iyi bir şekilde eğitilmesi için YOK1, YOK2 ... yardımıyla ön eğitim işleminin gerçekleştirilmesi için kullanılmaktadır. Daha sonra softmax sınıflandırıcısı, denetimli öğrenme yoluyla iyi bir şekilde eğitilebilir. Son olarak geri yayılım

algoritması, öğrenilen özellikler ve etiketler arasında en uygun ilişkiyi kurmak üzere tüm ağın ince ayarını yapmak için kullanılır (Qive diğ., 2017, s. 370).

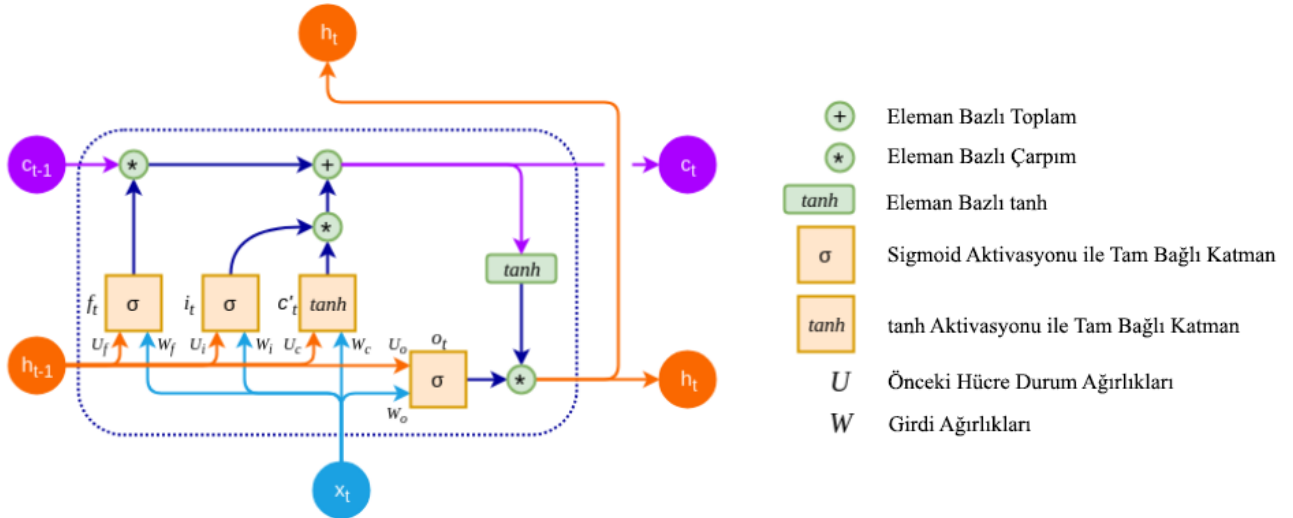
Çalışmada, yığın oto kodlayıcısı için günlük girdi değişkenlerinin sayısı 13, gizli katmanın boyutu 8, derinliği ise 5 olarak ayarlanmıştır.

2.3. Uzun / Kısa Süreli Bellek (LSTM – Long Short Term Memory)

Tekrarlayan Sinir Ağı (Recurrent Neural Network RNN), birimler arasındaki bağlantıların yönlendirilmiş bir döngü oluşturduğu yapay sinir ağı sınıfıdır. Bu döngü ile dinamik zamansal davranış sergilemesine olanak tanıyan bir ağ iç durumu oluşturulmuştur. İleri beslemeli sinir ağların aksine, tekrarlayan sinir ağları kendi giriş belleğini girdilerin rastgele dizilerini işlemek için kullanabilmektedirler. Tekrarlayan sinir ağındaki temel düşünce sıralı bilgileri kullanmaktır (Şeker, Diri, ve Balık, 2017, s. 51). Bir tekrarlayan sinir ağında, dizinin önceki basamağında hem alt katmanda hem de kendi çıkış değerinden gelen girdileri kullanarak aktivasyonları hesaplayan gizli düğümler kullanılır. Bu özyinelemeli doğa, ağı mevcut çıktılar hakkında karar vermek için geçmiş girdilerden faydalanmasını sağlar. Tekrarlayan sinir ağının kaybolan ya da patlayan gradyan problemi nedeniyle eğitilmesi zordur. Bunun sebebi gradyanların geriye yayılmaları dolayısıyla zaman içinde ölmesi ya da patlamasıdır. Bu durum, ağı veriler yardımıyla öğrenememesine neden olmaktadır (Erdogan, Hershey, Watanabe, ve Le Roux, 2017, s. 171).

Bir derin sinir ağı mimarisi olarak ifade edilen Uzun / Kısa Süreli Bellek, gradyan bozulma ve gradyan patlama problemleriyle başa çıkmak için geliştirilmiştir. Uzun Kısa Süreli Bellek katmanının ana bileşeni bellek bloğu adı verilen bir birimdir. Bir Uzun Kısa Süreli Belleği bloğunun giriş, çıkış ve unut kapıları olmak üzere üç kapısı vardır. Bu kapılar, hücreler için yazma, okuma ve sıfırlama işlemlerini gerçekleştirmektedir. Bir Uzun Kısa Süreli Bellek hücre durumu, her bir Uzun Kısa Süreli Bellek bloğu arasında bilgi taşıyan anahtar bileşendir. Hücre durumu üzerinde yapılan değişiklikler, bahsi geçen üç kapı ile kontrol edilmektedir (Singh ve diğ., 2017, s. 269). Hücreye aktarılacak verilerin seçilmesi, çıktı hesaplanırken bu bilgilerinin ne kadarının kullanılacağı gibi seçimlere olanak sağlayan kapı mekanizmalarını sisteme ekleyerek kaybolan ve patlayan gradyanlar sorununu çözmekte ve geçmiş hesaplamalardan taşınacak bilgileri rafine hale getirmektedir. Sahip olduğu girdi kapısı aracılığı ile hücre belleğine eklenecek bilgileri kontrol etmekte, unut kapısı ile geçmiş bilgilerin ne kadarının yeni verilere taşınacağına karar vermekte ve çıktı kapısı ile bellekteki bilgilerin ne kadarının çıktı hesaplanırken kullanılacağı kontrol edilmektedir (Tokgöz ve Ünal, 2018, s. 3).

Şekil 7’de Bir Uzun Kısa Süreli Bellek hücre yapısı gösterilmiştir (Vasilev, Slater, Spacagna, Roelants, ve Zocca, 2019, ss. 210-211):



Şekil 7. Uzun Kısa Süreli Bellek Hücre Yapısı

Şekil 7'de x_t , c_t ve h_t ; sırasıyla t anındaki Uzun Kısa Süreli Bellek girişi, hücre belleği durumu ve çıkışıdır. c'_t ise aday hücre durumunu temsil etmektedir. x_t girdisi ve önceki hücre çıkışı h_{t-1} , her kapıya ve aday hücre vektörüne sırasıyla W ve U ağırlık kümeleriyle bağlanmaktadır. f_t , i_t ve o_t ise; Uzun Kısa Süreli Bellek hücrelerinin sırasıyla unut, girdi ve çıktı kapılarını temsil etmektedir.

Unut kapısı, kararını bir önceki h_{t-1} hücrelerinin çıkışına ve mevcut x_t girdisine dayandırmaktadır. Bahsi geçen işlem, Denklem 10'daki eşitlik yardımıyla gösterilmiştir.

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \quad (10)$$

Bir önceki hücrenin c_{t-1} vektörünün her bir elemanına eleman bazlı lojistik fonksiyon uygulanmaktadır. İşlem eleman bazlı olduğundan, bu vektörün değerleri $[0, 1]$ aralığında sıkıştırılır. 0 çıkışı, c_{t-1} hücre bloğunu tamamen siler ve 1 çıkışı bu hücre bloğundaki bilgilerin geçmesine izin verir. Bu durum, Uzun Kısa Süreli Belleğin hücre durumu vektöründeki alakasız bilgilerden kurtulabileceği anlamına gelmektedir.

Girdi kapısı, bellek hücrelerine hangi yeni bilgilerin ekleneceğine karar vermektedir. Karar verme işlemi iki bölümden oluşmaktadır. İlk bölüm bilgilerin eklenip eklenmeyeceğine karar vermektedir. Girdi kapısı da kararını bir önceki h_{t-1} hücrelerinin çıkışına ve mevcut x_t girdisine dayandırmaktadır. Hücre vektörünün her bir hücre bloğu için lojistik fonksiyon aracılığıyla 0 veya 1 değerlerini çıktı olarak belirler. 0 çıkışı, bu hücre bloğunun belleğine hiçbir bilgi eklenmeyeceği anlamına gelmektedir. 1 çıkışı ise bu hücre bloğunun belleğine bilgilerin ekleneceği anlamına gelmektedir. Sonuç olarak, Uzun Kısa Süreli Bellek hücre durumu vektöründe spesifik bilgi parçalarını saklayabilmektedir. Bahsi geçen işlem, Denklem 11'deki eşitlik yardımıyla gösterilmiştir.

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \quad (11)$$

Eklenecek aday girdisi c'_t , önceki h_{t-1} hücresinin çıktısına ve mevcut x_t girdisine dayanmaktadır. c'_t , bir tanh fonksiyonu yardımıyla dönüştürülmektedir. Bahsi geçen işlem, Denklem 12'deki eşitlik yardımıyla gösterilmiştir.

$$c'_t = \tan h(W_c x_t + U_c h_{t-1}) \quad (12)$$

Unut ve girdi kapıları, yeni ve eski durumun hangi kısımlarının dahil edileceğini seçerek yeni hücre durumuna karar vermektedir. Bahsi geçen işlem, Denklem 13'teki eşitlik yardımıyla gösterilmiştir.

$$c_t = f_t * c_{t-1} \oplus i_t * c'_t \quad (13)$$

Çıktı kapısı, toplam hücre çıktısının ne olacağına karar vermektedir. Hücre belleğinin her bir bloğu için h_{t-1} ve x_t 'yi girdi olarak alır ve lojistik fonksiyonu aracılığıyla 0 veya 1 çıktı değerini sağlar. 0 çıktısı, bloğun herhangi bir bilgi çıktısı almadığı anlamına gelirken, 1 çıktısı ise bloğun bir hücrenin çıktısı olarak geçebileceği anlamına gelmektedir. Uzun Kısa Süreli Bellek böylece hücre durumu vektöründen spesifik bilgi blokları çıkarabilmektedir. Bahsi geçen işlem, Denklem 14'teki eşitlik yardımıyla gösterilmiştir.

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \quad (14)$$

Son olarak Uzun Kısa Süreli Bellek çıktısı bir tanh fonksiyonu ile transfer edilmektedir. Bahsi geçen işlem, Denklem 15'teki eşitlik yardımıyla gösterilmiştir.

$$h_t = o_t * \tan h(c_t) \quad (15)$$

Çalışmada gizli katmanların ve gecikmelerin sayısı sırasıyla 5 ve 4 olarak ayarlanmıştır. Finansal zaman serileri ise %80 eğitim, %10 doğrulama ve %10 test için üç alt gruba ayrılmıştır. Ayrıca öğrenme oranı, yığın (batch) büyüklüğü ve döngü (epoch) sayısı ise sırasıyla 0,05, 40 ve 1000 olarak belirlenmiştir.

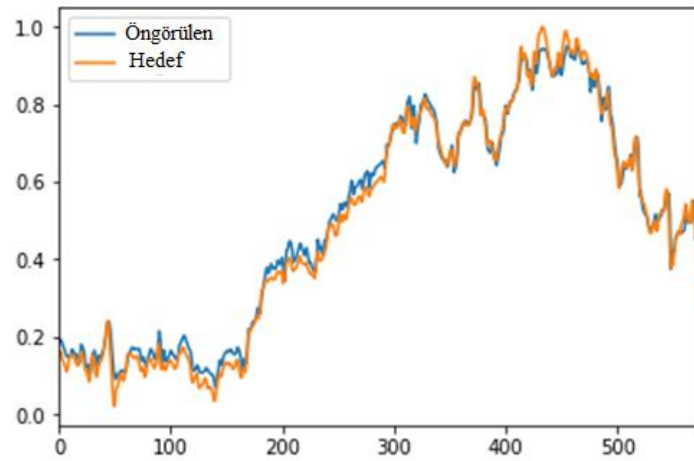
3. UYGULAMA ÖRNEĞİ

Bu çalışma, derin öğrenme mimarisi yardımıyla BİST 30 Endeksinin günlük hareket tahminini elde etmeyi amaçlamaktadır. Çalışmada girdi değişkeni olarak BİST 30 endeksinin günlük olarak açılış, yüksek, düşük, kapanış, düzeltilmiş kapanış değerleri ile MACD (Moving Average Convergence Divergence – Hareketli Ortalamaların Uyumu Uyumsuzluğu) Göstergesi, CCI (Commodity Channel Index – Emtia Kanalı Endeksi), ATR (Average True Range – Ortalama Gerçek Aralık) Göstergesi, BOLL (Bollinger Bands – Bollinger Bantları) Göstergesi, EMA20 (20 days Exponential Moving Average – 20 günlük Üstel Hareketli Aralık) Göstergesi, MA5 (5 days moving average – 5 günlük hareketli aralık) Göstergesi ve MA10 (10 days moving average – 10 günlük hareketli aralık) Göstergesi dikkate alınmıştır.

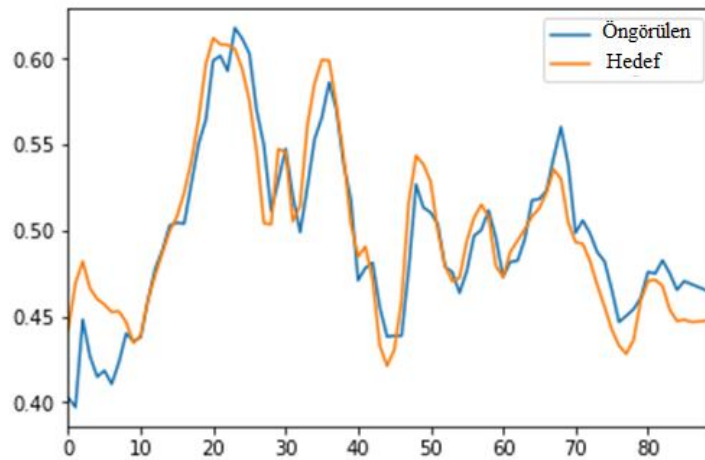
Çalışmada kullanılan 02.01.2001 ile 31.12.2018 tarihleri arasındaki BİST 30 günlük endeks bilgileri FİNNET Analiz Expert Excel Eklentisi yardımıyla elde edilmiştir.

Çalışmada ilk olarak BİST 30 Endeksinden elde edilen zaman serilerine dalgacık dönüşümü uygulanarak veri işleme faaliyeti gerçekleştirilmiş ve bu sayede zaman serisinde olabilecek gürültüler mümkün mertebe ortadan kaldırılmıştır. Bu işlem sonrasında elde edilen çıktılar ile MACD, CCI, ATR, BOLL, EMA20, MA5 ve MA10 göstergeleriyle beraber ölçeklenmiş ve elde edilen verilere denetimsiz bir şekilde eğitilmiş derin bir mimariye sahip yığın oto kodlayıcı uygulanmıştır. Son aşamada ise bir önceki aşamada elde edilen çıktı üzerine uzun kısa süreli bellek işlemi uygulanmıştır. Bahsi geçen derin öğrenme mimarisi, Anaconda Navigator'da yer alan Spyder bütünleşik geliştirme ortamında python programlama dili yardımıyla kodlanmıştır.

Yapılan analiz sonrasında elde edilen eğitim ve test tahmin sonuçları sırasıyla Şekil 8 ve Şekil 9'da gösterilmiştir.



Şekil 8. Eğitim Tahmin Sonuçları



Şekil 9. Test Tahmin Sonuçları

Yapılan analizde çapraz doğrulama aşamasında doğrulama kümesindeki kayıp 0.20 olarak elde edilmiştir. Buradan hareketle doğruluk oranının %80 olduğu görülmektedir. Bu oran çalışmanın performansta iyileştirme sağladığını ve geliştirilebileceğini göstermektedir.

4. SONUÇ

Derin öğrenme yöntemi, öngörücü performansı optimize etme amacıyla büyük veri setlerinden faydalanmaktadır. Bu nedenle, derin öğrenme yöntemi hem teorik hem de pratik olarak finans alanında birçok alan için uygundur (J. B. Heaton, Polson, ve Witte, 2017, s. 13). Bu nedenle derin öğrenme modelleri finansal piyasalardaki yatırımlarda karar almak için oldukça yararlı olabilir. Bu çalışma, derin öğrenme mimarisi yardımıyla BİST 30 Endeksinin günlük hareket tahminini mümkün olduğunca doğru gerçekleştirmeyi amaçlamaktadır. Çalışmada önerilen entegre sistem %80 gibi tatmin edici bir doğruluk oranına sahip olsa da bu oranın artırılması gerekmektedir.

Sonraki çalışmalarda girdi değişkeni olarak MTM (Momentum), ROC (Price rate of change – Fiyatlarda Yüzde Değişim), SMI (Stochastic Momentum Index), WVAD (Williams Variable Accumulation Distribution) gibi teknik göstergeler ile döviz kuru, enflasyon oranı, faiz oranı gibi makroekonomik göstergeler eklenerek tahmin doğruluğunun artırılması sağlanabilir. Bunun yanı sıra önerilen modeli eğitmek için daha fazla veri kullanarak daha doğru öngörü performansına yol açıp açmadığı belirlenebilir.

KAYNAKÇA

- Aslan, S., Badem, H., Özcan, T., Karaboğa, D., ve Baştürk, A. (2015). Çoklu Ekran Kartı ile Hızlandırılmış Ayrık Haar Dalgacık Dönüşümü Temelli Görüntü Şıkıştırma. *Kahramanmaraş Sutcu Imam University Journal of Engineering Sciences*, 18(2), 12-16.
- Buduma, N., ve Locascio, N. (2017). *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. Boston: O'Reilly Media, Inc.
- Canyılmaz, M., Türk, M., ve Güzel, E. (2016). Uzaktan Algılanan Düşük Frekanslı Sinyallerin Gürültülerinin Giderilmesinde Dalgacık Dönüşümü Ailelerinin Performanslarının İncelenmesi ve Karşılaştırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 28(1), 73-77.
- Chung, H., ve Shin, K.-s. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10), 3765-3782. doi:10.3390/su10103765
- Ciaburro, G., ve Venkateswaran, B. (2017). *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. Birmingham: Packt Publishing Ltd.

- Çalışan, M., ve Talu, M. F. (2017). *Examination of the Effect of the Basic parameters of the Auto-encoder on Coding Performance*. Paper presented at the 2017 International Artificial Intelligence and Data Processing Symposium (IDAP).
- Derbeko, P., Dolev, S., ve Gudes, E. (2018). *Privacy via Maintaining Small Similitude Data for Big Data Statistical Representation*. Paper presented at the International Symposium on Cyber Security Cryptography and Machine Learning, Beer Sheva, Israel.
- Erdogan, H., Hershey, J. R., Watanabe, S., ve Le Roux, J. (2017). Deep Recurrent Networks for Separation and Recognition of Single-Channel Speech in Nonstationary Background Audio. In *New Era for Robust Speech Recognition* (ss. 165-186): Springer.
- Gurjar, M., Naik, P., Mujumdar, G., ve Vaidya, T. (2018). Stock Market Prediction Using ANN. *International Research Journal of Engineering and Technology*, 5(3), 2758-2761.
- Heaton, J. (2015). *Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning* (Vol. 3). Chesterfield: Heaton Research, Inc.
- Heaton, J. B., Polson, N. G., ve Witte, J. H. (2017). Deep Learning for Finance: Deep Portfolios. *Applied Stochastic Models in Business Industry*, 33(1), 3-12.
- Hsieh, T.-J., Hsiao, H.-F., ve Yeh, W.-C. (2011). Forecasting Stock Markets Using Wavelet Transforms and Recurrent Neural Networks: An Integrated System Based on Artificial Bee Colony Algorithm. *Applied soft computing*, 11(2), 2510-2525.
- Kannadasan, K., Edla, D. R., ve Kuppili, V. (2018). Type 2 Diabetes Data Classification Using Stacked Autoencoders in Deep Neural Networks. *Clinical Epidemiology Global Health*.
- Kaynar, O., Aydın, Z., ve Görmez, Y. (2017). Sentiment Analizinde Öznitelik Düşürme Yöntemlerinin Oto Kodlayıcı Derin Öğrenme Makinaları ile Karşılaştırılması. *Bilişim Teknolojileri Dergisi*, 10(3), 319-326.
- Kaynar, O., Görmez, Y., ve Işık, Y. E. (2016). *Oto Kodlayıcı Tabanlı Derin Öğrenme Makinaları ile Spam Tespiti*. Paper presented at the 3. Uluslararası Yönetim Bilişim Konferansı, İzmir.
- Lewis, N. D. C. (2016). *Deep Learning Made Easy with R: A Gentle Introduction for Data Science*. California: CreateSpace Independent Publishing Platform.
- Liu, G., Bao, H., ve Han, B. (2018). A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis. *Mathematical Problems in Engineering*, 2018.
- Liu, H. (2018). Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network. *arXiv preprint arXiv:1806.06173*, 1-24.
- Mosavi, A., Ozturk, P., ve Chau, K.-w. (2018). Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11), 1536. doi:10.3390/w10111536
- Öner, İ. V., Yeşilyurt, M. K., ve Yılmaz, E. Ç. (2017). Wavelet Analiz Tekniği ve Uygulama Alanları. *Ordu Üniversitesi Bilim ve Teknoloji Dergisi*, 7(1), 42-56.
- Qi, Y., Shen, C., Liu, J., Li, X., Li, D., ve Zhu, Z. (2017). *An Automatic Feature Learning and Fault Diagnosis Method Based on Stacked Sparse Autoencoder*. Paper presented at the International Workshop of Advanced Manufacturing and Automation.

- Ramsey, J. B. (2002). Wavelets in Economics and Finance: Past and Future. *Studies in Nonlinear Dynamics & Econometrics*, 6(3).
- Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., ve Soman, K. P. (2017). *Stock Price Prediction Using LSTM, RNN and CNN-sliding Window Model*. Paper presented at the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).
- Sim, H. S., Kim, H. I., ve Ahn, J. J. (2019). Is Deep Learning for Image Recognition Applicable to Stock Market Prediction? *Complexity*, 2019, 1-10. doi:10.1155/2019/4324878
- Singh, D., Merdivan, E., Psychoula, I., Kropf, J., Hanke, S., Geist, M., ve Holzinger, A. (2017). *Human Activity Recognition Using Recurrent Neural Networks*. Paper presented at the International Cross-Domain Conference for Machine Learning and Knowledge Extraction.
- Şeker, A., Diri, B., ve Balık, H. H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Tokgöz, A., ve Ünal, G. (2018). *A RNN based time series approach for forecasting turkish electricity load*. Paper presented at the 2018 26th Signal Processing and Communications Applications Conference (SIU).
- Toprak, İ. B., Çağlar, M. F., ve Merdan, M. (2007). *Ayrık Dalgacık Dönüşümü ve Yapay Sinir Ağları Kullanarak EEG Sinyallerinden Otomatik Epilepsi Teşhisi*. Paper presented at the IEEE 15.Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Eskişehir.
- Ugwu, C., ve OnwuachuUzochukwu, C. (2014). Machine Learning Application for Stock Market Prices Prediction. *IOSR Journal of Computer Engineering*, 16(5), 112-122.
- Vasilev, I., Slater, D., Spacagna, G., Roelants, P., ve Zocca, V. (2019). *Python Deep Learning* (2. b.). Birmingham: Packt Publishing.
- Wang, Z., Hu, J., ve Wu, Y. (2018). A Bimodel Algorithm with Data-Divider to Predict Stock Index. *Mathematical Problems in Engineering*, 1-14. doi:10.1155/2018/3967525
- Wani, M. A., Bhat, F. A., Afzal, S., ve Khan, A. I. (2020). *Studies in Big Data* (Vol. 57). Singapore: Springer.