

T.C.

İSTANBUL AREL ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Yüksek Lisans Programı



**BORSA VERİLERİNİN YAPAY ZEKA VE DERİN
ÖĞRENME İLE SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

Ali İsa TAŞ

İstanbul, 2020

T.C.

İSTANBUL AREL ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Yüksek Lisans Programı



**BORSA VERİLERİNİN YAPAY ZEKA VE DERİN
ÖĞRENME İLE SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

Tezi Hazırlayan: **Ali İsa TAŞ**

İstanbul, 2020

KABUL VE ONAY

ALİ İSA TAŞ tarafından hazırlanan “BORSA VERİLERİNİN YAPAY ZEKA VE DERİN ÖĞRENME İLE SINIFLANDIRILMASI” adlı tez çalışmasının savunma tarihi 15.12.202 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği /oy çokluğu ile İstanbul Arel Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman

Dr. Öğr. Üyesi Gökalp TULUM

Üye

Dr. Öğr. Üyesi Necdet Tuğrul ARTUĞ

İstanbul Arel Üniversitesi

Üye:

Doç.Dr. Bülent BOLAT

Yıldız Teknik Üniversitesi

İstanbul Arel Üniversitesi Lisansüstü Eğitim Yönetim Kurulu'nun

..... tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Ali AKDEMİR

Lisansüstü Eğitim Enstitüsü Müdürü

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “**BORSA VERİLERİNİN YAPAY ZEKA VE DERİN ÖĞRENME İLE SINIFLANDIRILMASI**” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım.

Tarih/imza

Ali İsa TAŞ

ÖZET

BORSA VERİLERİNİN YAPAY ZEKA VE DERİN ÖĞRENME İLE SINIFLANDIRILMASI YÜKSEK LİSANS TEZİ

Ali İsa TAŞ

İSTANBUL AREL ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

(Danışman : Dr. Öğr. Gökalp TULUM)

İSTANBUL, 2020

Bu çalışmada Yahoo Finans sitesinden elde edilen S&P verisinin analiz edilmesi ve yapılan analizler sonucunda gelecek tahmini yapılması bilgisayar bilimi ve birçok yatırımcı için önemli bir yer tutmaktadır. Gelişen teknolojiler sayesinde bilgisayarların gücü artmış ve bu sayede daha güçlü algoritmalar geliştirilmeye başlanmıştır. Bu algoritmalar ile finansal veriler üzerinde tahminleme yapılması mümkün hale gelmiştir. Yapılan bu çalışmada derin öğrenme ve sığ öğrenme ile finansal verilerin tahminlenmesi amaçlanmıştır ve çalışmada S&P 500 endeksine ait 12.08.2000 ile 13.8.2020 tarihleri arasında günlük veriler kullanılmıştır. Bu bağlamda kullanılan verinin ilk 19 senesi (verinin 0.95'i) eğitim adımlarında, son 1 senesi ise tahmin (test verisi) olarak kullanılmıştır. Tahminleme yapılırken uzun kısa süreli bellek ve çok katmanlı algılayıcılar yöntemi kullanılmıştır. Elde edilen sonuçlar incelendiğinde çok katmanlı algılayıcılar yöntemi ile uzun kısa süreli bellek yöntemlerinin eğitim ve test hatalarının birbirlerine yakın olduğu gözlemlenmiştir. Elde edilen sonuçlar incelendiğinde sırasıyla eğitim, test ve tüm veri kök ortalama karesel hataları uzun kısa süreli bellek ağı için 17,3, 65,3 ve 22,0 dolar olarak hesaplanırken; çok katmanlı algılayıcı ağı için 16,1, 61,2 ve 20,6 dolar olarak bulunmuştur.

Anahtar Kelimeler: Derin Öğrenme, LSTM, Sığ Yapay Sinir Ağı, Zaman Serileri

ABSTRACT

FORECASTING STOCK MARKET DATA WITH DEEP LEARNING AND ARTIFICIAL INTELLIGENCE

GRADUATE SCHOOL, ISTANBUL AREL UNIVERSITY

ALİ İSA TAŞ

INSTITUTE OF GRADUATE STUDIES, ISTANBUL AREL UNIVERSITY

ELECTRICAL AND ELECTONICS ENGINEERING

(SUPERVISOR ASSIST.PROF. DR. Gökalp TULUM)

ISTANBUL, 2020

In this study, analyzing the S&P data obtained from the Yahoo Finance website and making future predictions as a result of the analysis have an important place for computer science and many investors. Thanks to the developing technologies, the power of the computers has increased and by this means more powerful algorithms have been developed. It has become possible to make predictions on financial data using these algorithms.

In the present study, it was aimed to predict financial data via deep learning and shallow learning and daily data of the S&P 500 index between 12.08.2000 and 13.8.2020 were used in the study. In this context, the first 19 years of the data (0.95 per cent of the data) were used in the training steps, and the only last 1 year of data was put to use as estimation (test data). Long short-term memory and multilayer perceptrons methods were used for estimation. During the examination of the obtained results, it was observed that the training and test errors of the multilayer perceptron and the long short-term memory methods were close to each other. Furthermore, while training, test and all-data root mean square errors are calculated as 17.3, 65.3 and 22.0 dollars respectively for the long short-term memory network; it was found to be 16.1, 61.2 and 20.6 dollars for the multilayer perceptron network.

Keywords: Deep Learning, LSTM, Shallow Artificial Intelligence

ÖNSÖZ

Bu çalışma İstanbul Arel Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği anabilim dalı Elektronik Programında yüksek lisans tezi olarak hazırlanmıştır.

Yapılan çalışmanın sonucunda görüldüğü üzere, MATLAB programı kullanılarak algoritma tasarlanmıştır. Bu algoritma ile borsa verileri üzerinde tahminleme yapılması mümkün hale gelmiştir. Yapılan bu çalışmada derin öğrenme ve sığ öğrenme ile finansal verilerin tahminlenmesi amaçlanmıştır

Bu tezin hazırlanmasında ve yaptığım çalışmalar sırasında değerli yardımlarını esirgemeyen sayın hocam Dr. Öğr. Üyesi Gökalp Tulum ve değerli eşim Ar.Gör. Pelin Gülüm'e teşekkür etmeyi borç bilirim.

Tezimin bu konuda çalışacak araştırmalara faydası olması dileğiyle.

Aralık 2020

Ali İsa TAŞ

İÇİNDEKİLER

Sayfa

ÖZET	I
ABSTRACT	II
İÇİNDEKİLER	III
ŞEKİL LİSTESİ	iv
TABLolar LİSTESİ	v
KISALTMALAR VE SEMBOL LİSTESİ	vi
ÖNSÖZ	vii
1 GİRİŞ	1
1.1 Literatür Özeti	2
2 YAPAY SİNİR AĞLARI	7
2.1 Yapay Sinir Ağlarının Tarihçesi.....	7
2.2 Yapay Sinir Ağı Mimarileri	8
2.3 Bazı Aktivasyon Fonksiyonları	10
2.4 Ağırlıklar	11
2.5 Sığ Öğrenme	11
2.6 Çok Katmanlı Algılayıcılar (MLP)	12
2.7 Algılayıcıların Yapısı	13
2.8 İleri Beslemeli Sinir Ağları	14
2.9 Öz Yinelemeli Sinir Ağları.....	14
2.10 Gradyan Yok Olması Sorunu	15
2.11 Uzun Kısa Süreli Bellek (LSTM)	16
2.12 Geçitlenmiş Özyinelemeli Birimler (GRU)	18
2.13 Çift Yönlü Özyinelemeli Sinir Ağları	19
3 ZAMAN SERİSİ ANALİZİ	21
3.1 Zaman Serisi ve Stokastik Süreçler.....	21
3.2 Ortalama ve Varyans	21

3.2.1 Zaman Serileri Bileşenleri	22
3.2.2. Trend	22
3.2.3 Mevsimler Dalgalanmalar	22
3.2.4 Döngüsel (Konjüktür) Dalgalanmalar	23
3.2.5 Düzensiz Rassal Hareketler	23
4. ZAMAN SERİSİ ANALİZ YÖNTEMİ	24
4.1 Basit Ortalama Yöntemi	24
4.2 Hareketli Ortalamalar Yöntemi	24
4.3 Üstel (Exponansiyel) Düzeltme Yöntemi	24
4.4 Trend Analiz Yöntemi	25
4.5 Mevsimler Dalgalanmalar ve Trendde Ortalamalar Yöntemi	25
5. MATERYAL VE METHOD	26
5.1 Veri	26
5.2 Open-Close-High Değerleri	26
6. SONUÇLAR VE TARTIŞMA	29
6.1 Sonuçlar	29
6.2 Tartışmalar.....	33
7. KAYNAKLAR.....	36
8. ÖZGEÇMİŞ	43

KISALTMALAR LİSTESİ

YSA:	Yapay Sinir Ağları
LSTM:	Uzun Kısa Süreli Bellek
YZGD:	Yapay Zekanın Geliştirilmesi Derneği
RMSE:	Kök Ortalama Kare Hatası
BPNN:	Geri Yayımlı Sinir Ağları
ANFIS:	Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi
RNN:	Özyinelemeli Sinir Ağları
NGSIM:	Gelecek Nesil Simülasyonu
RMS:	Karekök Ortalama
BCA:	Orta Asya Banka
MAE:	Ortalama Mutlak Hata
HMAE:	Değişkenliği Ayarlanmış Ortalama Mutlak Hata
HMSE:	Değişkenliği Ayarlanmış Ortalama Kare Hata
EMD:	Ampirik Mod Ayrışması
SVM:	Destek Vektör Makinesi
MLP:	Çok Katmanlı Algılayıcı
KOSPI:	Kore Kompozit Hisse Senedi Fiyatları Endeksi
1DCNN:	1 Boyutlu Evrişimsel Sinir Ağı
GRU:	Geçitlenmiş Özyinelemeli Birimler
ERNN:	Elman Özyinelemeli Sinir ağları

ŞEKİLLER LİSTESİ

Şekil 2.1 Yapay sinir ağı	9
Şekil 2.2 Bazı aktivasyon fonksiyonları	10
Şekil 2.3 Sığ öğrenmeli MLP yapısı	12
Şekil 2.4 Algılayıcıların yapısı	13
Şekil 2.5 Özyinelemeli sinir ağı yapısı	15
Şekil 2.6 LSTM mimarisi	16
Şekil 2.7 Geçitlenmiş özyinelemeli sinir ağı yapısı	18
Şekil 2.8 Çift Yönlü özyinelemeli sinir ağı mimarisi	19
Şekil 5.1 Tasarlanan algoritma	27
Şekil 6.1 Şekil 6.1 LSTM eğitim sonuçları	29
Şekil 6.2 Şekil 6.2 LSTM test sonuçları	30
Şekil 6.3 LSTM Tüm veri sonuçları	30
Şekil 6.4 Şekil 6.5 Sığ MLP test sonuçları	31
Şekil 6.5 Sığ MLP test sonuçları	31
Şekil 6.6 Sığ MLP tüm veri sonuçları	32

TABLULAR LİSTESİ

Tablo 1.	LSTM ve Sığ MLP yapılarının hata karşılaştırması	32
Tablo 2.	LSTM ve Sığ MLP hata histogram sonuçları.....	33
Tablo 3.	Literatür Karşılaştırması	34





1.Giriş

Tahmin etmek, hangi deęişkenlerin dięer deęişkenleri öngördüğünü anlamak olarak tanımlanabilir. Bu da birçok deęişken arasındaki olası gecikme ilişkilerinin zamanlamasının net bir şekilde fark edilmesi, bu olası gecikme ilişkilerinin istatistiksel öneminin anlaşılması ve piyasa hareketlerini tahmin etmek için sinyal olarak izlenecek deęişkenlerin hangilerinin daha önemli olduğunun öğrenilmesi demektir.

Tahmin etme, her zaman pazar profesyonellerinin ve araştırmacıların dikkatini çeken bir konu olmuştur. Bu bağlamda araştırmacılar zamanla çeşitli modeller ortaya atmış ve geliştirmişlerdir. Birçok konuda tahmin çalışmaları yapılmakta ve geliştirilen modeller bu konular özelinde çeşitlenmektedir. Finans alanı da tahmin etmenin ve yapılan tahminin gücünün son derece önem taşıdığı sektörlerin başında gelmektedir. Finansal varlıkların doğası gereği yapılan tahminlerde tutarlı sonuçlar elde etmek zordur. Son yıllarda finansal piyasalarda fiyat tahmini için makine öğrenimi algoritmalarının kullanılmasını sağlayan teknolojiler ile bilgisayarların hesaplama gücü artmıştır. Fakat tahminlerin hata oranlarındaki azalmalar bu gelişmelere nazaran daha azdır. Bu durum da özellikle finans alanında tahmin gücü yüksek algoritmaların kullanılmasını gerekli kılmaktadır.

Literatürde yer alan birçok yapay sinir ağı modeli piyasa değerini tahmin etmek için istatistiksel modeller temel alınarak çalışır. Yapılan çalışmalara bakıldığında çoğu durumda yapay sinir ağı (YSA) modellerinin dięer yöntemlerden daha iyi sonuçlar verdiği görülmektedir. Bununla birlikte yapay sinir ağı modellerinin kendi aralarındaki performanslarının mukayese edildiği az sayıda çalışma mevcuttur.

Bir yapay sinir ağı türü olan uzun kısa süreli bellek (LSTM) ile tekrarlayan sinir ağları, sıralı verilerle ilgili çeşitli öğrenme problemleri için etkili ve ölçeklenebilir bir model olarak ortaya çıkmışlardır. Yapılan bu tezde finansal verilerin uzun kısa süreli bellek yöntemi ile tahmin edilmesi üzerine çalışılmıştır. Bu tahminler gerçek yatırımlar ve dięer finansal piyasa endekslerinin genel yönünün analiz edilmesi açısından oldukça yararlıdır. Çalışmamızın amacı LSTM modelini ve çok katmanlı algılayıcıları (MLP) karşılaştırarak bu iki tekniğin üstünlüklerini tartışmak ve gelecek çalışmalara referans oluşturmaktır.

Bu araştırmanın ikincil amacı, tercih edilen yöntem hakkında ayrıntılı bilgi vermek için en iyi performans gösteren modelin özelliklerini ve eksikliklerini analiz etmektir.

Analiz edilen yöntemlerin uygulaması, Yahoo Finance S&P günlük verilerinin zaman serileri üzerinden gerçekleştirilmiştir. Çalışmanın bir sonraki adımı kıyaslanan iki teknikten galip gelen tekniğin literatürdeki diğer yöntemler ile birlikte nasıl kullanılabilceği veya tahmin performansını etkileyen parametrelerin nasıl iyileştirilebileceği olabilir.

1.1 Literatür Özeti

Geleneksel tahmin modellerine kıyasla daha az hata payı ile daha doğru sonuçlar sunan uzun kısa vadeli bellek mimarisi son yıllarda oldukça fazla kullanılmaktadır. Standart ileri beslemeli ağlardan farklı olarak LSTM mimarisinin geri bildirim ağları olması özellikle zaman serisi verileri ile sınıflandırma ve tahmin konularında oldukça etkili sonuçlar vermektedir.

Geçmiş çalışmalar incelendiğinde farklı disiplinlerden birçok tahmin probleminin LSTM mimarisi ile ele alındığı görülmektedir. Tahmin modellerinde gelecek durumları öngörmenin kritik olduğu özellikle finans sektörü ön plana çıksa da enerji, ulaşım, bilişim, veri madenciliği, dil işleme gibi farklı sektörlerde de kendine kullanım alanı bulmaktadır. Gelecek durumları tahmin ederek planlama yapmanın önemli olduğu günümüz dinamik dünyasında LSTM mimarisini çoğu tahmin gerektiren duruma adapte etmek mümkündür. Bu bağlamda yapılan çalışmalar incelendiğinde örneğin (Wang ve diğerleri,2020) yaptıkları çalışmada periyodik özelliğe sahip enerji tüketim probleminde gelecekteki tüketimin öngörülebilmesi için LSTM tabanlı bir yaklaşım ortaya koymuşlardır. Gerçek endüstri verileri kullanılarak yapılan uygulamada geliştirilen yöntemin geleneksel tahmin metotlarına kıyasla daha yüksek tahmin performansına sahip olduğu ve önerilen LSTM tabanlı yöntemin Mayıs ayı tahminlerinden elde edilen RMSE değerlerinin, BPNN(Back Propagation Neural Network), ARMA(Autoregressive Moving Avarage Model), ARFIMA(Auto Regressive Fractional Integrated Moving Average Model) yöntemlerine kıyasla %19., %54.85 ,ve %64.59 daha düşük olduğu sonucuna varılmıştır. Benzer konuda yapılmış bir diğer çalışmada (Kong ve diğerleri, 2017) şebeke planlama ve işletimi için önemli bir gösterge olmasına rağmen yüksek volatilitte ve belirsizlik yüzünden hesaplanması zor olan bireysel elektrik müşterilerinin kısa vadeli yük tahmini konusunda derin öğrenme tekniklerinden tekrarlı LSTM kullanarak tahmin yapacak bir yapı kurmuşlardır. Yaklaşım gerçek veriler ile test edilerek RBF(Radial Basis Function),

ARIMA, ANFIS(Adaptive Neural Fuzzy Inference System) gibi yük tahmininde kullanılan rakip algoritmalara göre daha iyi performans gösterdiği ortaya koyulmuştur. Enerji sektörü ile ilgili tahminlerin yanı sıra LSTM gelecek tahminini gerekli kılan ve planlamanın oldukça önemli olduğu ulaştırma sistemlerinde de kullanılmıştır. (Zhao ve diğerleri, 2017) ulaşım sistemlerinin etkin planlanması ve yönetilmesi hususunda kritik olan trafik tahmini konusunda LSTM ağına dayanan bir tahmin modeli önermişlerdir. Geleneksel tahmin modellerinden farklı olarak, önerilen LSTM ağı, birçok bellek biriminden oluşan iki boyutlu bir ağ aracılığıyla trafik sisteminde zamansal-uzamsal korelasyonu göz önünde bulundurmaktadır. Diğer temsili tahmin modelleri ile yapılan karşılaştırma, önerilen LSTM ağının daha iyi bir performans elde edebileceğini doğrulamıştır. Ulaştırma konusunda (Yanjie ve diğerleri,2016) Highways England tarafından sağlanan seyahat süresi verilerini kullanarak, veri setindeki 66 bağlantı için 66 seri tahmin LSTM sinir ağı kullanarak Seyahat süresi tahmini yapmışlardır. Eğitimli modeli kullanarak test setindeki her bir bağlantı için çok adımlı ileri hareket sürelerini tahmin etmişler ve değerlendirme sonuçlarında, 1- adım önde olan seyahat süresi tahmin hatasının nispeten küçük olduğunu, 66 bağlantı için ortalama göreceli hatanın medyanının test setinde %7.0 olduğunu göstermişlerdir. (Zhao ve arkadaşları,2018) ağ trafiğinin uzaysal-zamansal özelliklerini modellemek için LSTM RNN'lere dayanan yeni bir derin mimari önermiş ve daha sonra gelecekteki öngörülen trafik matrisini doğru bir şekilde hesaplamak için doğrusal bir regresyon modeli kullanmışlardır. Simülasyon sonuçlarında, DLTMP(Genel trafik matrisi tahmini, daha önce elde edilen ağ trafiği verileri ile gelecekteki trafik matrisinin öngörücüsünü elde eden sorunu çözmeyi amaçlar) mimarisine dayanan trafik matrisi tahmin yaklaşımının diğer tipik trafik matrisi tahmin yaklaşımlarından daha iyi performans gösterdiğini ve son teknoloji trafik matrisi tahmin performansına ulaşabildiğini göstermişlerdir.(Altché ve arkadaşları,2016) NGSIM(Next Generation Simulation) veri kümesindeki doğal sürüş verilerini kullanarak, karayolu üzerindeki gelecekteki araç yörüngelerini tahmin etmek için LSTM'lere dayanan bir sinir ağı mimarisi kullanmışlardır. Bu ağın gelecekte 10 saniye yanal konum için yaklaşık 70 cm'lik ortalama bir RMS hatası ile daha iyi tahmin doğruluğu sağladığı gösterilmiş ve önceki teknolojilerden daha iyi tahmin doğruluğu sağladığı ortaya konulmuştur. (Wang ve arkadaşları, 2020) deprem olaylarının tahmin edilmesi amacı ile uzay-zamansal açıdan yeni bir deprem tahmin sistemi önerdiler. Özellikle, deprem olayları arasındaki mekansal-zamansal korelasyonları bulabilen ve doğru deprem tahminleri

yapmak için korelasyonlardan yararlanabilen iki boyutlu girdili bir LSTM ağı tasarladılar. LSTM ağının etkinliğini ve verimliliğini artırmak için önerilen ayrıştırma yönteminin, sistem performansını önemli ölçüde artırabildiği gösterilmiş, simülasyon sonuçları ayrıca sistemin farklı zamansal ve uzamsal tahmin ayrıntılarıyla doğru tahminler yapabileceğini ortaya koymuştur.

LSTM mimarisinin sıkça kullanıldığı doğal dil işleme alanında (Wang ve arkadaşları,2016) tarafından gerçekleştirilen çalışmada doğal dil çıkarımı görevi için özel bir LSTM mimarisi önerilmiştir. Öncülün dikkat ağırlıklı vektör temsillerini elde etmek için ilk olarak nöral dikkat modellerini kullanmış daha sonra hipotezi öncül ile eşleştirmeye çalışırken hipotez kelimesini kelimeye göre işleyen bir match-LSTM tasarlamışlardır. Stanford Natural Language Inference corpus ile ilgili deneyler, ortaya konan bu match-LSTM modelinin bu veri setinde bugüne kadar bildirilen son teknoloji performanstan daha iyi performans gösterdiğini belirtmiştir. (Graves ve arkadaşları,2005) da TIDIGITS konuşma grubunun ayrık alt kümelerinde, erkeklerden, kadınlardan, kızlardan ve genç erkeklerden sözler içeren bir LSTM ağını sırayla yeniden eğitmiş ve ağı daha önce hiç görülmemiş verilere çok farklı konuşma özelliklerine sahip hızlı ve doğru bir şekilde adapte olabildiğini tespit etmişlerdir. Ayrıca, nihai performansının genellikle daha önce farklı veri kümeleri üzerinde eğitilmesiyle artırıldığını ve bu gelişmenin birden fazla yeniden eğitime devam ettiğini tespit etmişlerdir.

Mevcut durumları doğru analiz ederek gelecekteki fırsatları ve riskleri öngörmenin gerekli olduğu finans alanında yapılmış çalışmalar incelendiğinde; (Siami-Namani ve diğerleri, 2018) LSTM gibi zaman serisi verilerini tahmin etmek amacı ile geliştirilen derin öğrenme tabanlı algoritmaların finansal tahminde AR(Autoregressive Moving),MA(Moving Avarage), SES(Simple Exponential Smoothing) ve ARIMA(Autoregressive Integrated Moving Average) gibi geleneksel tahmin metotlarından daha üstün performans gösterdiğini ve LSTM kullanılarak yapılan tahminlerin ARIMA'ya kıyasla RMSE(The Root-Mean-Square Error) hata oranlarında %84-87 arasında azalmaya neden olduğunu ortaya koymuşlardır. Ayrıca çalışmada derin öğrenmedeki eğitim sürelerinin sayısının (epoch) eğitilmiş ağı tahmin performansına bir etkisi olmadığı ve rastgele bir davranış sergilediği görülmüştür. Bu çalışmanın aksine (Roondiwala ve diğerleri,2017) hisse senetleri tahmini için tekrarlı sinir ağı(RNN) ve LSTM yaklaşımlarını kullandıkları

çalışmalarının sonunda *RMSE* 0.00983 eğitim, *RMSE* 0.00859 test sonucunun 500 epoch ve açılış, kapanış, yüksek, düşük olmak üzere 4 parametre ile elde edildiği gözlemleyerek epoch sayısının önemine değinmişleridir.

(Manurung ve diğerleri,2018) hisse senedi fiyatları tahmini için Yahoo Finance 2013-2018 yılı Bank Central Asia(BCA) verilerini kullanarak verideki önemli parametreler için (Açılış, Yüksek, Düşük, Kapanış) LSTM ile tahmin çalışması gerçekleştirmişlerdir. Deneyin sonucunda LSTM'deki en doğru tahminin 3 yıllık veya 5 yıllık eğitim verisi kullanmak yerine 1 yıldan az kısa vadeli veri kullanarak elde edildiği ve LSTM'nin %56 doğruluk oranına sahip geleneksel tahmin yöntemi ARIMA'ya kıyasla %94'e kadar kısa vadeli verilerde üstün olduğunu ortaya koymuştur. Böylece LSTM mimarisinin performansının dikkate alınacak zaman aralığından etkilendiği sonucu ortaya koyulurken bunun yanında geleneksel modellerden farkı da tespit edilmiştir. (Kim ve diğerleri, 2018) tarafından finans sektöründe meydana gelen hisse senedi fiyat değişimlerini tahmin etmek üzere LSTM modelini otoregresif koşullu heteroscedastisite(GARCH) tipi modellerle birleştiren hibrit bir yaklaşım önerilmiştir. MAE(Mean Absolute Error), MSE(Mean Squared Error), HMAE(heteroscedasticity adjusted Mean Absolute Error) ve HMSE(heteroscedasticity adjusted Mean Squared Error) hata oranları ile geleneksel yöntemler mukayese edilerek önerilen hibrit yaklaşımın performans üstünlüğü ortaya koyulmuştur. Benzer konuda yaptıkları çalışma ile (Zhuge ve diğerleri,2017) hisse senedi açılış fiyatlarını tahmin etmek amacıyla duygusal analiz(emotional analysis) ve LSTM modelinden oluşan iki aşamalı bir yaklaşım önermişlerdir. Verileri sınıflandırmak için ilk adımda Bayesian temelli sınıflandırıcı kullanılmış ve ardından önceki deneylerden elde edilen duygusal veriler, uzun davranış verileri LSTM öğrenme modeli için eğitim verisi olarak gerçek davranış verileri ile birleştirilerek kullanılmıştır. Sonuç olarak tahmin doğruluğu performansının büyük ölçüde iyileştiği ve önerilen iki aşamalı tekniğin geleneksel sinir ağı yaklaşımlarından daha üstün olduğu ortaya koyulmuştur. Duygusal analiz ve bu iki aşamalı LSTM modelinin yanında (Kim ve diğerleri, 2019) ise hisse senedi fiyatlarını tahmin etmek için aynı verilerin farklı gösterimlerinden yani stok zaman serileri ve stok grafik görüntülerinden elde edilen özellikleri bir araya getiren LSTM-CNN(Convolutional Neural Network) yaklaşımını önermişlerdir. Çalışmada, bu iki tekniğin ayrı ayrı kullanılmasındansa bir arada kullanılmasının tahmin hatasını etkili bir şekilde

azaltabileceği sonucuna varılırken bir şamdan grafiğinin hisse senedi fiyatlarını tahmin etmek için en uygun grafik resmi olduğu tespit edilmiştir. Farklı tekniklerin bir arada kullanılmasının daha etkili olabileceği konusunda bir diğer örnek çalışma da (Cao ve diğerleri, 2019) tarafından gerçekleştirilen ve borsa fiyat tahmini için benzer bir hibrit yaklaşımı ampirik mod ayrışması EMD,(Empirical Mode Decomposition) ve LSTM yöntemlerini bir araya getirerek kullanmış oldukları çalışmadır. Geçmiş verilerin tahmin sonucu üzerindeki etkisini sağlamak için EMD ve CEEMDAN (Complete Ensemble Empirical Mode Decomposition) birikiminden tüm karakteristik seriler için LSTM tahmin modelleri oluşturulmuştur. Önerilen modellerin tahmin performansı küresel borsa endekslerinin doğrusal regresyon analizi ile doğrulanmakta, tekli LSTM, SVM(Support Vector Machine), MLP(Mulyi Layer Perceptron) ve diğer hibrid yaklaşım modelleri ile karşılaştırıldığında finansal zaman serileri tahmininde bir adım önde olduğu vurgulanmıştır. LSTM ağının sahip olduğu üstün özelliklerden birine değinen çalışmalarında (Thomas ve diğerleri,2017) bir LSTM ağının gürültülü finansal zaman serisi verilerinden anlamlı bilgileri etkili bir şekilde çıkarabildiğini başarıyla göstermiştir. LSTM, random forest(RF), standart derin ağlar ve lojistik regresyon ile karşılaştırıldığında, tahmini doğruluk ve işlem maliyetlerinden sonraki günlük getirilere göre tercih edilen yöntem olduğunu göstermiştir. LSTM ağları şeklinde derin öğrenme bu nedenle bu alanda da bir gelişme olarak görüldüğünü göstermişlerdir. (Sangyeon ve diğerleri,2019) KOSPI 200 endeksinin eğilimlerini tahmin etmek için MLP, 1DCNN(One-Dimensional Convolutional Neural Networks), LSTM, ve dikkat ağları modellerini test ettikleri çalışmalarında ilk olarak, LSTM, MLP ve 1D CNN'den daha fazla zamana bağlı olan sıralı verilerle iyi çalıştığını doğrulamışlardır. İkinci olarak, daha uzun geçmişe dönük günleri, LSTM ağlarını içeren modellerin daha iyi isabet oranları ve daha iyi genel sonuçlar yarattığını göstermiş ve attention network ağlarının uzun ardışık verilerle daha iyi sonuçlar sunduğunu, sezgisel analiz modelleri için görselleştirme avantajına sahip olduğunu ortaya koymuşlardır.

2. Yapay Sinir Ağları

2.1 Yapay Sinir Ağlarının Tarihçesi

Sinir ağlarının tarihi, insan beyninin aktivitesini, mevcut herhangi bir bilgisayardan daha güçlü kılan temel özellikleri yakalamak için modelleyici bir girişim olarak başlamıştır (Marini ve diğerleri.,2008). Yapay sinir ağları deneyimlerden öğrenip bunları genelleştirebilir ve verilerden gerekli bilgileri alır. YSA'lar, proses elementlerini birbirine bağlayan ağırlıkları güçlendirerek veya zayıflatarak insanlar gibi "düşünmesi" için eğitilmiştir (Wang ve diğerleri.,2007). YSA'ların ve evrimsel arama üreticilerinin kombinasyonlarına ilgi son yıllarda hızla artmıştır (Yao ve diğerleri.,1993). 1923 yılında, Karel Capek, ilk robot konseptini Rossum'un Universal Robots tiyatro oyununda tanıttı ve 1936 yılında Konrad Zuse, 64K bellek adı verilen Z1 adlı programlanabilir bir bilgisayar geliştirdi (Mijwil,2015). Sinir ağlarının ilk temel modellenmesi aslında tek bir nöron idi ve 1943'te McCulloch ve Pitts tarafından hesaplamalı bir "sinirsel aktivite" modeli olarak önerildi. (McCulloch ve diğerleri.,1943). Daha sonra Hebb 'Davranış Organizasyonu' kitabında yapay bir sinir ağının öğrenimini uygulamak için ilk matematiksel kuralı sağlayarak "mevcut bir yolun nöronlar arasındaki bağlantıları güçlendirdiğini" ileri sürdü (Hebb ve diğerleri.,1949). 1959 yılında Bilişsel Sistem Araştırma Programı Dr. Frank Rosenblatt tarafından kurulmuştur. "Algı"yı tanımlayan geniş çaplı ilk teknik raporu 1957'de Cornell Havacılık Laboratuvarı himayesinde yayınlandı(Rosenblatt,1957).

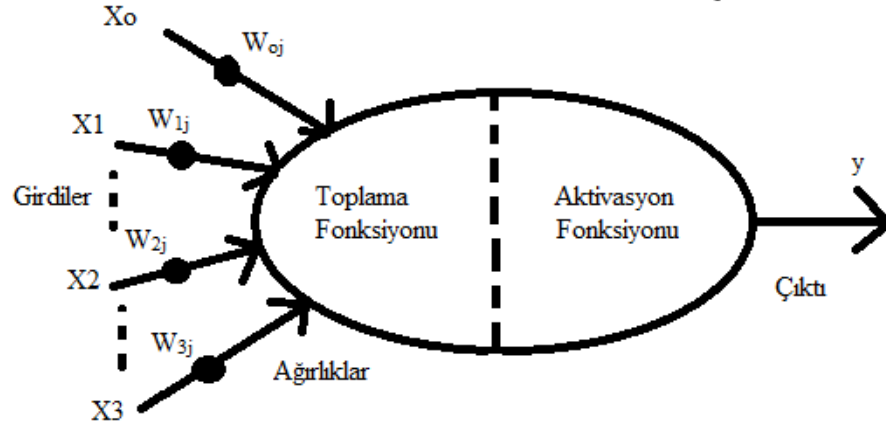
1956 yılında ise tanınmış bilim adamları ve doktora öğrencileri Dartmouth Yaz Araştırma Projesi'nde bir araya geldi ve kabaca bir beyni nasıl simule edeceğini tartıştılar. Yapay zekanın ilk destekçileri, yazılım yoluyla yetenekleri simule etmek isterken, sinir ağlarının destekçileri, sistemin en küçük parçalarını-nöronları taklit ederek sistem davranışını elde etmek istediler (Kriesel,2005:sy).

1969 yılında Minsky ve Papert adli bilim adamları tarafından yayınlanan makalede, algılayıcılar, sadece birden fazla nöron katmanı eklenerek aşılamayan ciddi sınırlamalara maruz kaldılar (Minsky ve diğerleri.,1987). 1971 yılında, Terry Winograd'ın doktora tezi (MIT), bilgisayarların çocuk bloklarının sınırlı bir dünyasında, İngilizce anlama talimatlarını uygulayan bir robot kolu ile SHRDLU'nun bir araya getirilmesiyle, çocuk bloklarının sınırlı bir dünyasında İngilizce cümleleri anlama yeteneğini gösterdi (Buchanan,2005). 1980'lerde yapay zeka araştırmacıları

yapay zeka yaratmanın ilk düşünceden çok daha karmaşık olduğunu anlamaya başlamışlardı. Bunu göz önünde bulundurarak, Brooks bilinçte ileriye giden yolun, araştırmacıların daha sonra bir araya getirilebilecek bir planlama modülü, bir bellek modülü gibi insan beyninin farklı yönlerine dayanan bireysel modüller oluşturmaya odaklanması olduğuna inanmaya başladı (Brunette ve diğerleri.,2009). 1980'lerin başında, Hopfield modelleri olarak adlandırılan tekrarlayan ağlar sınıfını eğitmek için Hebb 'in erken çalışmalarına dayanan yaklaşımlar getirmişlerdir.(Hopfield,1982.) 1985 yılında Harold Cohen tarafından oluşturulan otonom çizim programı Aaron, AAAI Ulusal Konferansı'nda gösterildi (Buchanan,2005). 1993 yılında MIT'de insan görünümlü bir robot olan Cog üretimi başladı ve 2000 yılında İletişimde jest ve mimik hareketleri kullanabilen Kısmet adlı robot tanıtıldı. Ayrıca 2010 yılında akıl gücünü kullanan ASIMO geliştirildi (Mijwil,2015). Daha sonra sinir ağlara olan ilgi daha da artmaya başladı ve gerçekten de, tahmin, sınıflandırma veya kontrol problemleri olan her yerde sinir ağları tanıtıldı (Marini, ve diğerleri.2008).

2.2 Yapay Sinir Ağı Mimarileri

Yapay Sinir Ağları (YSA) insan beynin çalışmasından ilham alır ve makine öğrenimi ve yapay zeka için kullanılabilir. Bu ağlarla bilgisayar tabanlı çeşitli problemler çözülebilir. İnsanların beyinlerinden esinlenen YSA'lardaki hesaplamalar, ağ üzerinden katmanlar halinde dağıtılan yapay nöronlar olarak bilinen birimler halinde uygulanır. Belirli bir nöronun girişleri dışarıdan veya bir önceki katmanın nöronlarının çıkışlarından toplanabilir (Maad ve diğerleri.,2019). Bir nöronun çıkışını hesaplamak için, toplanan tüm girişler, her biri, her bir girdi başına atanan belirli bir değerle çarpılarak ve aktivasyon fonksiyonu olarak bilinen doğrusal olmayan bir fonksiyondan geçirilmeden önce toplanarak hesaplanır. Bu doğrusal olmama durumu, daha karmaşık özellikleri algılama özelliğine sahip daha esnek çıktılar sağlar. Bununla birlikte, gerektiğinde bias olarak bilinen hesaplamalara yanlılık sağlamak için bir nöronun girişlerine ek değer eklenebilir (Montufar ve diğerleri.,2014)



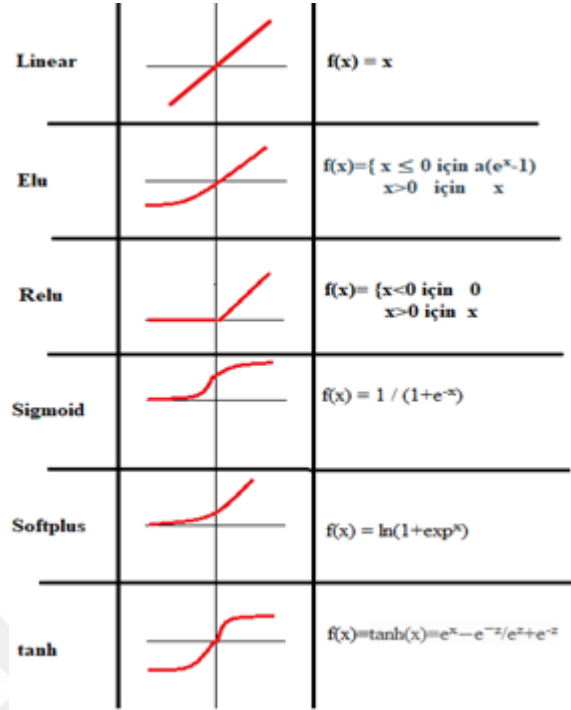
Şekil 2.1 Yapay Sinir Ağı (Aslay ve diğerleri.,2013)

Girdi, genellikle giriş katmanına gizli katmanlara dağıtacak çok boyutlu bir vektör şeklinde verilir. Gizli katmanlar daha sonra önceki katmandan kararlar alarak ve kendi içindeki tahmini bir değişimin nihai çıktığı nasıl etkilediğini veya iyileştirdiğini hesaplar (O'Shea ve diğerleri.,2015).

YSA'nın çıkışı, ileri geçişten ve amaçlanan çıkış değerleri arasındaki veri kümesinden sapma ölçülerek, çıkışın ağırlıkları türevleri hesaplanır (Maclaurin ve diğerleri, 2015).

Yapay bir sinir ağının parametrelerinden biri de aktivasyon işlevleridir. **Aktivasyon fonksiyonu**, giriş kenarlarının ağırlıklı toplamını hesaplayan ve fonksiyonun davranışına göre bir çıkış değeri üreten bir ikili eşik birimdir (Jain ve diğerleri.,1996). Aktivasyon fonksiyonları, bir nöronun harekete geçip geçmeyeceğine karar vermek için kullanılan giriş ve ağırlıklar toplamını hesaplamak için sinir ağlarında kullanılan işlevlerdir. Sunulan verileri, genellikle dereceli azaltma olan bazı gradyan işlemleriyle kullanır ve daha sonra nöral ağ için verilerdeki parametreleri içeren bir çıktı üretir. Bu aktivasyon fonksiyonlarına literatürde sıklıkla bir transfer fonksiyonu da denir (Nwankpa ve diğerleri 2018).

2.3 Bazı Aktivasyon Fonksiyonları



Şekil 2.2 Bazı aktivasyon fonksiyonları

Lineer aktivasyon fonksiyonu, yalnızca tüm gerçel sayı aralığı boyunca pozitif sayılar üretir

Relu fonksiyonu : Pozitif girişler için 1 ve negatif olanlar için 0 gradyanına sahiptir. Gradyanın sıfır olması ilk bakışta bir sorun olarak görülebilir, ancak aslında ağıın yararlı bağlantıları koruyarak seyrek olmasına yardımcı olur.

ELU: Ortalama sifıra ulaşmak için aktivasyon fonksiyonunu kullanır. Öğrenmeyi yavaşlatan şey, ReLU'larda mevcut olan bias kaymasıdır. Bunların ortalama aktivasyonu sıfırdan büyüktür ve öğrenme, sonraki katmanlar için önyargı kaymasına neden olur. Dolayısıyla ELU, ortalama aktivasyonu sifıra doğru iterek önyargı kaymasını azalttığı için ReLU'ya iyi bir alternatiftir (Pedamonti,2018)

Sigmoid, yalnızca 0 ile 1 arasında pozitif sayılar üretir. Sigmoid etkinleştirme işlevi, aynı zamanda 0 ile 1 arasında olan eğitim verileri için en yararlıdır. En çok kullanılan etkinleştirme işlevlerinden biridir(Sibi ve diğerleri,2013)

Softplus, kaybolan gradyan probleminin üstesinden gelmek için, softplus gibi yeterli gradyanları korumak için bazı sınırsız aktivasyon fonksiyonları önerilmiştir.

Tanımlama alanında sorunsuzdur. Bu özellik, pozitif ve negatif yönlerden tahmin edilirken softplus işlevini daha kararlı hale getirir.(Zheng ve diğerleri,2015)

Tahn, Hiperbolik Tanjant fonksiyonudur. Tanh işlevi sigmoid işlevine benzer, ancak orijinin etrafına simetrikdir. Tanh fonksiyonu sürekli ve türevlenebilir, değerler -1 ile 1 arasındadır.(Sharma ve diğerleri,2020)

2.4 Ağırlıklar

Girdilerin, sinir ağırları üzerinde belirledikleri katsayılar vardır. Bu katsayılara ağırlık denir. Bu sinir ağırlarının her hücresinin kendi ağırlıkları vardır. Ağırlık katsayısının büyük olması girdinin sinir hücresi için önemli olması, katsayının küçük olması ise girdinin sinir hücresine zayıf başlanması anlamına gelir (Elmas,2007:np). Ağırlıklar vektörüne yapılan atamalar ve dolayısıyla ilk kez ağ eğitimi yoluyla elde edilen hatalar istenilen gibi olmayabilir. Hatalar için en uygun değerleri bulmak için, hatalar çıkış katmanından gizli katmanlara doğru ağa “geri yayılır” ve bunun sonucunda ağırlıklar ayarlanır. Bu işlemler aynı gözlemlerle birkaç kez tekrarlanır ve tahmin edilen değerlerde ve daha sonra maliyette bir iyileşme olana kadar ağırlıklar yeniden ayarlanır ve en aza indirilene kadar model eğitilir (Namini ve diğerleri,2018).

2.5 Sığ Öğrenme

Yakın zamana kadar, çoğu makine öğrenimi ve sinyal işleme tekniği sığ yapılı mimarilerden yararlanıyordu. Bu mimariler tipik olarak tek bir doğrusal olmayan özellik dönüşümleri katmanı içerir ve çok sayıda uyarlanabilir doğrusal olmayan özellik katmanı eksikliği vardır. Sığ mimariler arasında, geleneksel gizli Markov modeller, doğrusal veya doğrusal olmayan dinamik sistemler, koşullu rastgele alanlar, maksimum entropi modelleri, destek vektör makineleri (SVM'ler), lojistik regresyon, çekirdek regresyon ve çoklu tek bir gizli katmana sahip katman algılayıcı sinir ağırları vardır. Sığ öğrenme modellerinde giriş sinyallerini veya özelliklerini gözlemlenemeyen probleme özgü bir özellik alanına dönüştürmekten sorumlu yalnızca bir katmandan oluşan basit mimariler vardır (Deng,2011).

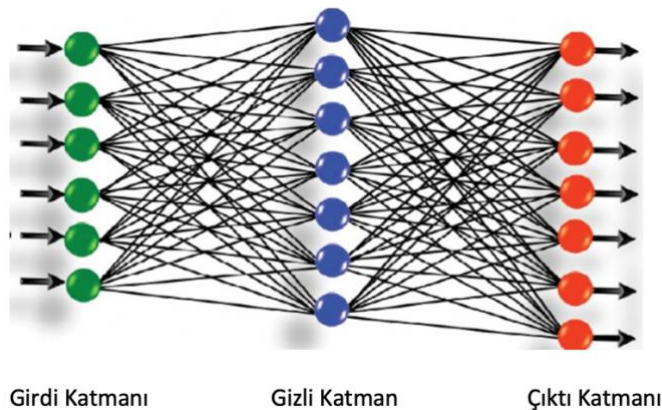
Derin modeller tipik olarak çıktıları teslim etmeden önce girdileri birçok kez dönüştürürken, sığ modeller genellikle girdileri yalnızca bir veya iki kez dönüştürür. Sonuç olarak, derin modeller daha karmaşık kalıpları öğrenebilir, manuel özellik mühendisliği olmadan uçtan uca öğrenmeye izin verir iyi performans gösterir. Derin

öğrenme, son yıllarda çok hızlı gelişen ve bilgisayarla görme ve doğal dil işleme gibi alanlarda başarılı uygulamalara sahip olan birden çok temsil düzeyini ve doğrusal olmayan işlem birimlerinin birden çok katmanını içerir. Aksine, tüm derin olmayan öğrenme yaklaşımları sığ öğrenme olarak sınıflandırılabilir. Hem derin hem de sığ öğrenme, bu durumda soğutma yükü tahmininde olduğu gibi, zaman serisi veri tahminiyle başa çıkabilir. Derin öğrenmede, Tekrarlayan Sinir Ağı, sıralı bilgileri ayıklayıp bir sonraki adım adımına geçirebilen sıralı bilgilerle başa çıkmak için tasarlanmıştır. Yüzeysel öğrenme için, sıralı bilgiler genellikle günün saati ve haftanın günü gibi zamanla ilişkili vektör değişkenlerle temsil edilir.

2.6 Çok Katmanlı Algılayıcılar (MLP)

Algılayıcı, Cornell bilim adamı Frank Rosenblatt tarafından önerilen basit bir ikili sınıflandırma algoritmasıdır. Bir dizi giriş sinyalini "evet" ve "hayır" olmak üzere iki kısma ayırmaya yardımcı olur. Ancak diğer sınıflandırma algoritmalarının aksine, algılayıcı, insan beyninin temel birimi olan nöron temel alınarak modellenmiştir ve karmaşık problemleri öğrenme ve çözme konusunda olağanüstü bir yeteneğe sahiptir. Çok katmanlı bir algılayıcı (MLP), karmaşık problemleri çözmek için birkaç katman halinde yığılmış ek algılayıcılarla bir araya gelen bir algılayıcıdır.

Aşağıdaki diyagram, üç katmanlı bir MLP'yi göstermektedir. Soldaki birinci katmandaki (giriş katmanı) her algılayıcı, ikinci katmandaki (gizli katman) tüm algılayıcılara çıktılar gönderir ve ikinci katmandaki tüm algılayıcılar çıktıları sağdaki son katmana gönderir.

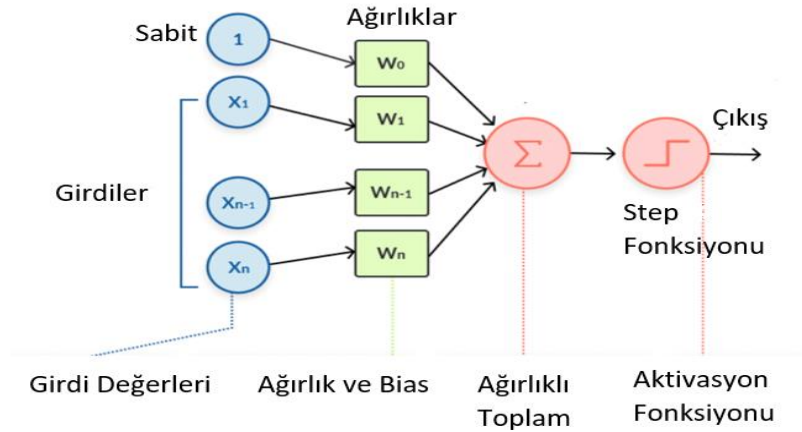


Şekil 2.3 Sığ Öğrenmeli MLP Yapısı (Serwa,2017)

Her bir algılayıcı, bir sonraki katmandaki her algılayıcıya bir sinyal giden birden çok sinyal gönderir. Her sinyal için algılayıcı, farklı ağırlıklar kullanır. Yukarıdaki diyagramda, bir katmandaki bir algılayıcıdan sonraki katmana giden her çizgi farklı bir çıktıyı temsil etmektedir. Her katman çok sayıda algılayıcıya sahip olabilir ve birden çok katman olabilir, böylece çok katmanlı algılayıcı hızla çok karmaşık bir sistem haline gelebilir. MLP'lerden gelişen sinir ağlarında, genellikle 0 ile 1 arasında veya -1 ile 1 arasında gerçek değerlerin çıktıları ile sonuçlanan diğer etkinleştirme işlevleri kullanılabilir. Bu, olasılığa dayalı tahminlere veya öğelerin çoklu etiketlere sınıflandırılmasına izin verir.

2.7 Algılayıcıların Yapısı

Bu sinir ağındaki algılayıcı veya nöron, basit ama ustaca bir yapıya sahiptir. Aşağıda gösterilen dört bölümden oluşur.



Şekil 2.4 Algılayıcıların Yapısı

Algılayıcıların yapısı,

1. Algılayıcıların girdi değerleri gerçek değeri alır.
2. Ağırlıklar, ağırlıkların her birinin sınıflandırma kararına göre göreceli önemini temsil eder. Bir "bias ağırlığı" eklenir ve 1'e eşit bir bileşenle çarpılır.
3. Giriş değerleri ağırlıklarla çarpılır ve toplanır, böylece aktivasyon fonksiyonlarına beslenen bir toplam değer oluşturulur.
4. Aktivasyon fonksiyonları sınıflandırma kararları oluşturur. (Missinglink, b.t.).

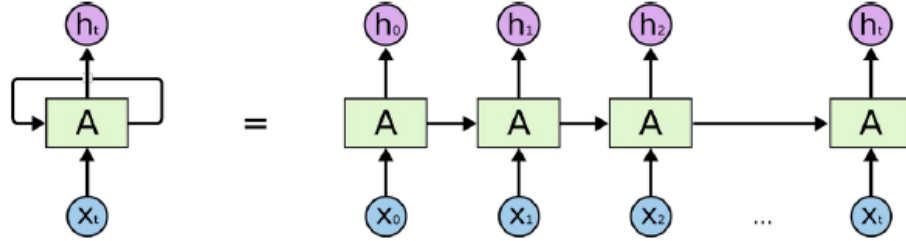
2.8 İleri Beslemeli Yapay Sinir Ağları

Yapay sinir ağı türüne bakılmaksızın, bu ağların her birinde girişten çıkış yönüne, ileri geçiş olarak bilinir, İleri geçiş, her katmanın çıktısını hesaplayarak ve ikincisinde yürütülen hesaplamalarda kullanarak, ağın çıktılarına dayanarak hesaplanması için kullanılır. Ters geçişte ise, ağırlıkların değerleri dereceli olarak azaltılarak güncellenir (Maclaurin ve diğerleri .,2015). Dereceli azaltma, konum ağırlıklarının değerinin bu hatayı azaltmak için güncellenmesi gerektiğini, yani o pozisyondaki düzgün degradenin negatifini tanımak için kullanılır. Bu güncelleme, sinir ağının girilen değerlerden istenen çıktıyı üretmesini ve böylece gerekli görevi gerçekleştirmesini sağlar. Bu işlemi birkaç tekrar için tekrarlayarak, ileri geçişten çıkış ile amaçlanan çıkış arasındaki kayıp, minimum kayba ulaşılan kadar sinir ağının performansını artıran geri yayılım kullanılarak azaltılır (Lee ve diğerleri.,2016). İleri beslemeli bir ağda, bir katmandaki her nöron sadece bir sonraki katmanın nöronları ile bağlantı kurmuştur. İleri beslemeli bir ağ için izin verilen bağlantılar düz çizgilerle temsil edilir. Her nöronun bir sonraki katmanın tüm nöronlarına bağlı olduğu ileri beslemeli ağlardır (Kriesel,2005:np)

2.9 Öz Yenilemeli Sinir Ağları (RNN)

Tekrarlayan bir sinir ağı (RNN) hedefin, dizide gözlemlenen önceki adımlara göre gözlem dizisindeki bir sonraki adımı tahmin etmek olduğu özel bir sinir ağı örneğidir. RNN'lerin arkasındaki fikir sıralı gözlemlerden faydalanmak ve gelecekteki eğilimleri tahmin etmek için önceki aşamalardan öğrenmektir (Namini ve diğerleri.,2018). Öz yinelemeli sinir ağları, dil ve konuşma gibi zaman serileri veya ardışık veriler için kullanılan başka bir derin öğrenme türüdür. Daha uzun zaman periyotlarını içermeye kabiliyetleri nedeniyle derin RNN'ler tercih edilir (Deng ve diğerleri.,2014). Öz yinelemeli sinir ağları gelen girişleri işlemek için dahili belleği kullanır ve çeşitli alanlardaki zaman serisi verilerini analiz etmek için kullanılır. Öz yinelemeli sinir ağları metin, dil çeviri uygulamaları ve sıralı veri işlemedeki bir sonraki karakteri tahmin etmede iyidir. (LeCun ve diğerleri.,2015). Bu ağın mimarisinde geciktirme elemanı vardır ve bu eleman üzerinden geri besleme yapılır. Geri besleme bir katmandaki hücreler arasında ya da katmanlar arasındaki hücreler arasında da yapılabilir ve bu yüzden doğrusal ve dinamik değildir.(Subaşı,2010). Sinir ağının bir kısmını Şekil 2.5'teki gibi gösterilmiştir. Döngü, bilginin ağ içindeki

adımlar arasında iletilmesine izin verir. Bir öz yinelemeli sinir ağı aynı ağda A'nın birden fazla kopyası olarak kabul edilebilir; burada her kopya, Şekil 2.5'te gösterildiği gibi arkasından gelene bir mesaj iletir.



Sekil 2.5 Özyinelemeli Sinir Ağı Yapısı (Olah, 2017)

İlgili bilgi ile ihtiyaç duyulan yer arasındaki boşluğun küçük olduğu durumlarda, RNN'ler geçmiş bilgileri kullanmayı öğrenebilir. Bununla birlikte, daha fazla bağlama ihtiyacı duyduğumuz durumlar vardır ve ilgili bilgi ile çok büyük olması gereken nokta arasındaki boşluğun olması mümkündür. Uygulamada RNN'ler bu durumu iyi ele alamazlar (Bengio ve diğerleri.,1994)

2.10 Gradyan Yok Olması Sorunu

Öz yinelemeli sinir ağının gradyanların hem kaybolması hem de yok olmasından dolayı dezavantajları vardır (Hochreiter,1991). Her iki soruna da, eğimi esas olarak yüksek bir güce yükseltilecek şekilde tekrarlayan ağırlık matrisine eşit olan öz yinelemeli sinir ağları yinelemeli doğasından kaynaklanır. Bu yinelenen matris güçleri, gradyanın zamana bağlı devir sayısında üstel olan bir oranda büyümesine veya küçülmesine neden olur (Jozefowicz ve diğerleri.,2015).

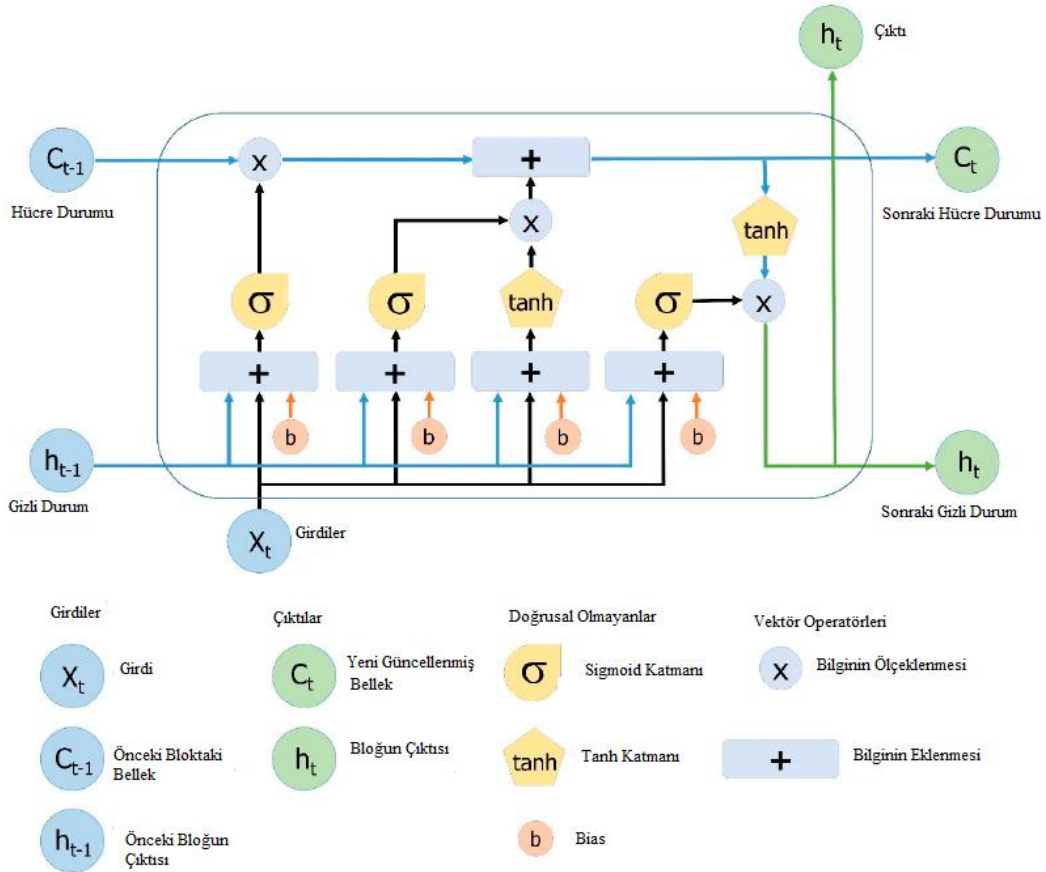
Gradient exploding yani aşırı büyümesi ağın çok büyük değerler üretmesini sağlayarak doğru sonuçtan uzaklaştıracaktır. Bunun için threshold koyarak çok yüksek değerli gradientleri kesmek basit ve etkili yollardan biridir (Mikolov ve diğerleri,2012).

Gradyan çok sık bir şekilde büyük bir faktör tarafından azaltılırsa öğrenme zarar görecektir. Gradyan kırılma çoğu zaman gradyan küçük bir norma sahip olduğunda son derece etkilidir. Kaybolan gradyan daha zordur çünkü gradyanın kendisinin küçük olmasına neden olmaz; gradyanın uzun vadeli bağımlılıklara karşılık gelen yönlerdeki bileşeni küçükken, gradyanın kısa vadeli bağımlılıklara karşılık gelen yönlerdeki bileşeni büyüktür. Sonuç olarak, RNN'ler kısa vadeli kolayca öğrenebilir, ancak uzun vadeli bağımlılıkları öğrenemez. LSTM, RNN'yi yeniden

parametrelendirerek yok olan gradyan sorununu giderir (Jozefowicz ve diğerleri.,2015).

2.11 Uzun Kısa Süreli Bellek (LSTM)

Yıllar içinde öz yinelemeli sinir ağlarının eğitime zorluklarının üstesinden gelmek için bir takım girişimlerde bulunulmuştur. Bu zorluklar, diziyeye dayalı görevlerin uzun vadeli bağımlılıklarını öğrenmek için geçitli yapıya sahip bir öz yinelemeli sinir ağları türü olan Uzun Kısa Süreli Bellek ağları (LSTM'ler) tarafından başarıyla ele alınmıştır (Hochreiter ve diğerleri.,1997). LSTM, Olah'a göre zincir yapısı şeklinde tasarlanmıştır. Ancak, yinelenen modül farklı bir yapıya sahiptir. Standart bir öz yinelemeli sinir ağları gibi tek bir sinir ağı yerine, benzersiz bir iletişim yöntemiyle etkileşen dört katmanı vardır (Olah, 2018).



Sekil 2.6 LSTM Mimarisi (Yan,2016)

LSTM ağı, hücreler adı verilen bellek bloklarından oluşur. Hücre durumu ve gizli durum bir sonraki hücreye aktarılmaktadır. Hücre durumu, verilerin temelde değişmeden ileriye doğru akmasına izin veren ana akış veri zinciridir. Ancak, bazı

doğrusal dönüşümler meydana gelebilir. Veriler hücre durumuna eklenmesi veya hücre durumundan çıkarılması gibi olaylar sigmoid kapılar yolu ile gerçekleşebilir. Bir geçit, farklı ağırlıklar içeren bir katmana veya bir dizi matris işlemine benzer. LSTM'ler, uzun süreli bağımlılık probleminden kaçınmak amacıyla ezberleme sürecini kontrol eden kapılardan meydana gelmişlerdir.

Bir LSTM ağı oluşturmanın iki adımı vardır bunlar, gerekli olmayan ve o adımda hücreden çıkarılacak bilgileri tanımlamaktır. Verileri tanımlama ve hariç tutma işlemine, t-1 zamanında son LSTM biriminin (h_{t-1}) ve t zamanında akım girişinin (X_t) çıkışını alan sigmoid işlevi karar verir. Ek olarak, sigmoid işlevi eski çıktının hangi kısmının ortadan kaldırılması gerektiğini belirler. Bu kapıya unutmaya kapısı (veya f_t) denir; burada f_t , hücre durumundaki her sayıya (C_{t-1}) karşılık gelen, 0 ile 1 arasında değişen değerlere sahip bir vektördür.

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (2.1)$$

Burada σ , sigmoid fonksiyonudur ve W_f ve b_f unutmaya geçidinin sırasıyla ağırlık matrisleri ve sapmasıdır. Bir sonraki adım ise, hücre durumunda yeni girişin (X_t) bilgisine karar vermek, depolamak ve hücre durumunu güncellemektir. Bu adım, sigmoid katman ve tanh katmanı olarak iki bölümden oluşur. Birincisi, sigmoid katman yeni bilgilerin güncellenip güncellenmeyeceğine karar verir (0 veya 1) ve ikincisi tanh fonksiyonu, önem düzeylerine (-1 ile 1) karar vererek geçen değerlere ağırlık verir.

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i), \quad e.2$$

$$N_t = \tanh(W_n[h_{t-1}, X_t] + b_n), \quad e.3$$

$$C_t = C_{t-1}f_t + N_t i_t \quad e.4$$

Burada, C_{t-1} ve C_t , t-1 ve t zamanındaki hücre durumları iken, W ve b , sırasıyla hücre durumunun ağırlık matrisleri ve sapmasıdır. N_t ise yeni hücredir. Son adımda, çıkış değerleri (h_t) çıkış hücresi durumuna (O_t) dayanır, ancak filtrelenmiş bir versiyondur. İlk olarak, sigmoid bir katman, hücre durumunun hangi bölümlerinin çıktıya ulaşacağına karar verir. Daha sonra sigmoid geçidinin (O_t) çıkışı, tanh tabakası tarafından hücre durumundan (C_t) oluşturulan yeni değerlerle çarpılır ve -1 ile 1 arasında bir değer bulunur.

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o), \quad e.5$$

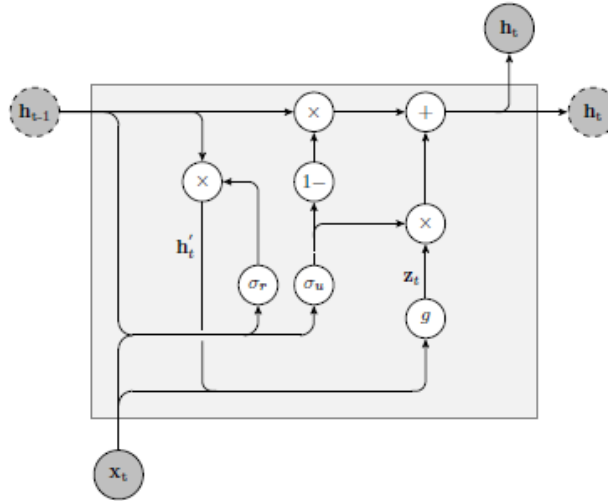
$$h_t = O_t \tanh(C_t)$$

e.6

Burada W_o ve b_o , çıkış kapısının sırasıyla ağırlık matrisleri ve sapmasıdır (Le ve diğerleri.,2019)

2.12 Geçitlenmiş Özyinelemeli Birimler(GRU)

Geçitlenmiş özyinelemeli birimler (GRU), başlangıçta Cho ve arkadaşları tarafından farklı zaman ölçeklerinde bağımlılıkları uyarlamalı olarak yakaladığını göstermiştir (Cho ve diğerleri.,2014). GRU'da, unutma ve giriş kapıları, her bir gizli ünitenin ne kadar hatırlayabileceğini veya unutabileceğini uyarlamalı olarak kontrol eden tek bir güncelleme kapısında birleştirilir. GRU'daki dahili durum, LSTM'deki çıkış kapısı gibi bir kontrol mekanizmasının olmaması nedeniyle her zaman çıktıda tam olarak ortaya çıkar. Geçitlenmiş özyinelemeli sinir ağlarının yapısı Şekil 2.7'deki gibidir.



Şekil 2.7 Geçitlenmiş Özyinelemeli sinir ağı yapısı

Düz çizgili koyu gri daireler, içeriği ağın girişi ve çıkışı ile değiştirilen değişkenlerdir. Kesikli çizgiye sahip koyu gri daireler, içeriği gizli katmanın hücreleri içinde değiştirilen iç durum değişkenlerini temsil eder. G operatörü, genellikle hiperbolik tanjant olarak uygulanan doğrusal olmayan bir dönüşümdür. "+", "-1" ve "x" olan beyaz daireler doğrusal işlemleri temsil ederken σ_r ve σ_u , sırasıyla sıfırlama ve güncelleme kapılarında kullanılan sigmoidlerdir. Yapıdaki kapılardan biri olan güncelleme kapısı, hücrenin mevcut içeriğinin yeni aday durumla ne kadar güncellenmesi gerektiğini kontrol eder. Bir diğeri olan sıfırlama kapısı, 0'a yakın bir

değerse, hücrenin belleğini etkili bir şekilde sıfırlayabilen ve o birimin bir sonraki işlenmiş girişin sekanstaki ilkmiş gibi davranmasını sağlar.

GRU'nun durum denklemleri şunlardır:

$$\text{Reset Kapısı: } r[t] = \sigma(W_r h[t-1] + R_r x[t] + b_r), \quad e.7$$

$$\text{Şimdiki Durum: } h'[t] = h[t-1] \odot r[t] \quad e.8$$

$$\text{Gelecek durum: } z[t] = g(W_z h'[t] + R_z x[t] + b_z), \quad e.9$$

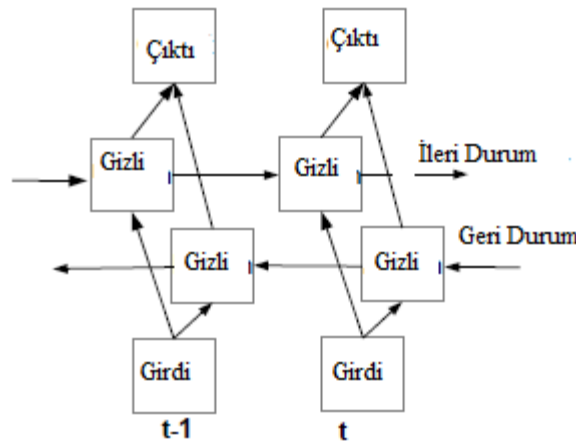
$$\text{Yenileme Kapısı: } u[t] = \sigma(W_u h[t-1] + R_u x[t] + b_u) \quad e.10$$

$$\text{Yeni durum: } h[t] = (1 - u[t]) \odot h[t-1] + u \quad e.11$$

Burada, $g(\cdot)$ genellikle hiperbolik bir tanjant tarafından uygulanan doğrusal olmayan bir fonksiyondur. Bir GRU hücresinde, parametre sayısı bir ERNN biriminden daha fazladır, ancak bir LSTM hücresinden daha azdır. Öğrenilecek parametreler dikdörtgen W_r , W_z , W_u matrisleri, R_r , R_z , R_u kare matrisleri ve b_r , b_z , b_u bu bias vektörleridir (Bianchi ve diğerleri.,2018).

2.13 Çift Yönlü Özyinelemeli Sinir Ağları

RNN'ler için potansiyel zayıflık, RNN'lerin mevcut zaman adımında girdinin aşamalı olarak işlenmemesidir (Huang ve diğerleri.,2014). Çift yönlü RNN'ler, giriş için hem geçmiş hem de gelecek bağlamdan yararlanır. Şekil 2.8'de, iki zaman aşaması için zamanla katlanan iki yönlü bir RNN mimarisi gösterilmiştir.



Şekil 2.8 Çift Yönlü Özyinelemeli Sinir Ağı Mimarisi

Şekil 2.8'de gösterildiği gibi, çift yönlü RNN'leri ileri gizli katman h_t^F , $t = 1, \dots, T$ 'den giriş sekansı boyunca yinelenerek ve geriye doğru gizlenmiş h_t^B katmanı, $t = T, \dots, 1$ 'den giriş sekansı boyunca tekrarlayarak hesaplanır. Bu iki gizli katman, aşağıdaki denklemler kullanılarak tek bir çıktı katmanında birleştirilir:

$$h_t^F = \tanh (W_{xh}^F X_t + W_{hh}^F h_{t-1}^F + b_h^F) \quad \text{e.12}$$

$$h_t^B = \tanh (W_{xh}^B X_t + W_{hh}^B h_{t+1}^B + b_h^B) \quad \text{e.13}$$

$$y_t = W_{hy}^F h_t^F + W_{hy}^B h_t^B + b_y \quad \text{e.14}$$

(Arisoy ve diğerleri.,2015.)



3. Zaman Serisi Analizi

Kronolojik olarak art arda gelen olaylardan oluşan veri grupları “zaman serisi” olarak adlandırılır. Zaman içerisinde art arda gerçekleşen olayları kaydederek toplanan veriler farklı sektörlerde ve disiplinlerde sıkça kullanılmaktadır. İş dünyasında günlük döviz kurları, saatlik borsa endeksleri, hammadde fiyatları, yıllık satış miktarları, haftalık faiz oranları gibi zamana bağlı veriler sürekli olarak takip edilmektedir. Biyolojik bir deneyde gözlemlenen parametrenin saatlik değişimi, ekonomide yıllık ithalat ve ihracat oranları, tarımda yıllık yağış miktarları gibi aslında hayatın her alanında zamana bağlı veriler ön plana çıkmaktadır.

Zaman serilerinin analiz edilmesi iki temel noktaya dayanmaktadır. İlki gözlemlenen seriye neden olan stokastik mekanizmayı anlamak ve tanımlamak, ikincisi zaman serisinin geçmiş verilerini ve davranış modellerini kullanarak geleceğe yönelik istatistiksel tahminlerde bulunmaktır. Zaman içerisinde sürekli olarak kaydedilen veriler sürekli zaman serilerini oluştururken belli zaman aralıkları ile kaydedilen verilere sahip seriler kesikli zaman serileri olarak nitelendirilirler.

3.1 Zaman Serisi ve Stokastik Süreçler

Rastgele değişkenler dizisine $\{A_t : t=0, +1, +2, +3, \dots\}$ stokastik süreç denir ve bu dizi gözlemlenmiş zaman serileri için bir model görevi görür. Bu A değerlerinin uyduğu ortak bir dağılımın tüm bilgileri (ortalama, varyans, kovaryans gibi) kullanılabilir.

3.2 Ortalama ve Varyans

Stokastik bir süreç $\{A_t : t=0, +1, +2, +3, \dots\}$ için değerlerin ortalaması aşağıdaki gibi tanımlanır.

$$u_t = E(A_t) \quad t=0, +1, +2, \dots \text{ için} \quad (e.15)$$

Burada μ_t t anındaki beklenen değeri ifade etmekte ve farklı zamanlar için t farklı değerler almaktadır.

Varyans $\{A_t : t=0, +1, +2, +3, \dots\}$ gibi bir süreç için istatistiksel yayılımın mümkün bütün değerlerin denklem (1) ile ifade edilen beklenen değerden veya ortalamadan uzaklıklarının karelerinin ortalaması şeklindeki ölçüdür.

$$VAR(x) = E((X - \mu)^2) \quad (e.16)$$

3.2.1 Zaman Serisi Bileşenleri

Zaman serileri trend, mevsimsel dalgalanmalar, döngüsel dalgalanmalar ve düzensiz terimlerden oluşmaktadır. Bunların yanında beklenen değer ile gerçekleşen değer arasındaki farkı temsil eden hata terimi de zaman serilerinin bir diğer bileşenidir.

3.2.2 Trend

Gözlemlenen bir değişkenin uzun dönemde artış veya azalış eğiliminde olması trend olarak adlandırılır. Trend doğrusal ve doğrusal olmayan trend olarak iki gruba ayrılabilir.

Doğrusal trend zamanla doğrusal bir hareket sergilerken doğrusal olmayan trend eğrisinde böyle bir durum gözlemlenmez ve trend hareketi bir eğri olarak ortaya çıkar.

Zaman serilerinde trend kalıplarının gözlemlenebilmesi için uzun bir zaman dilimindeki verilerin gözlemlenmesi gerekir. Trend zaman serisi içinde artış veya azalış eğilimi olarak kendini gösterebilmektedir. Bu noktada trendin nelerden kaynaklandığı ve hangi faktörlerin zaman serisinde böyle etkilere neden olduğu detaylı analiz edilmelidir.

3.2.2 Mevsimsel Dalgalanmalar

Mevsimsel dalgalanmalar zaman serisinin belli periyotlardaki zaman dilimlerinde aynı hareketlerin gözlemlenmesi anlamına gelmektedir. Bir meşrubat markasının yıllık satış verileri incelendiğinde yaz aylarındaki satış miktarlarının diğer aylara göre daha yüksek olduğu ve bunun her yıl benzer şekilde tekrarlandığı fark edilecektir. Doğalgaz tüketiminin yılın belli dönemlerinde azalması veya inşaat sektöründeki gelişmelere bağlı olarak inşaat işçisi talebinin yaz aylarında artış göstermesi gibi durumlar mevsimsel dalgalanmalara farklı örneklerdir.

Burada önemli olan nokta zaman serisinde meydana gelen dalgalanmaların mevsimsellikten kaynaklandığını tespit edebilmektir. Veriler doğru yorumlanır ve dalgalanmaların kök nedenlerine inilirse gelecek yılların tahmini için doğru bir kaynağa ulaşılmış olacaktır.

3.2.3 Döngüsel (Konjüktür) Dalgalanmalar

Konjüktürel dalgalanmalar daha çok ekonomi veya sektörün refah veya durgunluk dönemini içeren değişimler olarak tanımlanmaktadır. Bu terim, ekonomideki büyüme ve daralma dönemlerinin dönüşümlü olarak yaşanmasını ifade etmektedir. Mevsimsel dalgalanmaların aksine periyodik olmayan ve düzensiz bir seyir halindedir. Ayrıca mevsimsel dalgalanmalara kıyasla konjüktürün daha uzun sürmesi ve mevsimsel etkilerden daha geniş bir etki alanına sahip olması dikkat çeker. Zaman serisinde meydana gelen dalgalanmalar sınıflandırılır ve analiz edilirken bu iki temel nokta mevsimsellik ve konjüktürün birbirinden ayrılması için yol gösterici noktalar olurlar.

3.2.4 Düzensiz (Rassal) Hareketler

Bu hareketler diğer dalgalanma türlerinin aksine belli bir etki altında olmaksızın meydana gelen düzensiz artış veya azalışlardır. Bu dalgalanmalara neden gösterilebilecek bir değişim olmadığı için zaman serilerindeki yanıltıcı hareketlerdir. Tahmin yöntemlerinde veya zaman serisinin tanımlanması aşamasında karar vericiyi zorlayabilir veya yanıltabilirler (Emeç,b.t).

4.Zaman Serisi Analiz Yöntemleri

Karar verici çalıştığı veri grubunun ihtiyaç duyduğu formuna göre mevsimsellik etkisinden arındırabilir veya trend etkisini göz ardı etmek isteyebilir. Özetle zaman serilerinin detaylı analizine ihtiyaç duyulabilir. Zaman serilerinin temel bileşenlerinin(trend, mevsimsellik, konjunktür ve rassal hareketler) analiz edilmesine yönelik kullanılan birçok yöntem mevcuttur. Bu yöntemler sayesinde serinin geçmiş hareketlerinden yola çıkılarak gelecekte göstermesi muhtemel davranışlar tahmin edilmeye çalışılır.

4.1 Basit Ortalama Yöntemi

Mevcut verilerden geleceğe yönelik tahmin yapmak için en basit metot basit ortalama yöntemidir. Tahmin edilmek istenen değişkenin geçmiş dönemlerde aldığı değerlerin toplanıp ortalamasının alınması suretiyle hesaplanır.

$$\hat{A}_i = \frac{\sum_{i=1}^n A_i}{n} = A^- \quad (e.17)$$

4.2 Hareketli Ortalamalar Yöntemi

Bu yöntemde tahmin edilmek istenen değişkenin (n+1). dönemde alacağı değer, geçmiş n dönemde aldığı değerlerin ortalamasının hesaplanması ile bulunur. Bu yöntemde son dönemlerin de genel ortalamaya katkı sunması ayırt edici bir özelliktir. Burada n değerinin kaç alınacağı önemlidir ve karar vericinin ne kadar eski verileri tahminine dahil etmek istediğine göre seçilir.

$$\hat{A} = \frac{A_{i-1} + A_{i-2} + \dots + A_{i-n}}{n} \quad (e.18)$$

4.3 Üstel(Exponansiyel) Düzeltme Yöntemi

Bu yöntemde t. dönemin tahmini hesaplanırken (t-1). dönemin gerçekleşen ve tahmin değerleri kullanılır. Bu değerler için α gibi bir üstel düzeltme sabiti kullanılır ve (t-1) dönemine ait gerçekleşen değer ile α çarpılarak elde edilen değere (t-1) dönemine ait tahmin değerinin (1- α) ile çarpılmış hali eklenerek t dönemi için tahmin değeri elde edilir. Genellikle birinci dönem için tahmin ve gerçekleşen dönem verileri eşit kabul edilir.

$$\hat{A}_i = \alpha Y_{i-1} + (1 - \alpha)\hat{A}_{i-1} \quad 0 \leq \alpha \leq 1 \quad (\text{e.19})$$

4.4 Trend Analizi Yöntemi

Trend analizi yönteminin temeli basit doğrusal regresyon modeline dayanmaktadır. Aradaki fark X değişkeninin zaman serilerinde “zaman” kavramını temsil etmesidir. a ve b değerleri hesaplanarak $Y=aX+b$ denklemi bulunur. Bu denklem kullanılarak da farklı X zamanları için tahmin değerleri hesaplanabilir.

$$Y = aX + b$$

4.5 Mevsimsel Dalgalanmalar ve Trendde Ortalamalar Yöntemi

Bu yöntemde veriler herhangi bir ayırım gözetmeksizin trend analizine tutulurlar ve böylelikle trendin mevsimsel etkilerden arınmış olarak nasıl bir davranış sergilediği hesaplanmış olur. Ardından mevsimsellik dikkate alınarak her periyot için elde edilen verilerin trendin ne kadar altında veya üzerinde olduğu hesaplanır. Her mevsime ait yapılan bu hesaplamalar bir sonraki yıl için bir tahmin sunacaktır. (Yavuz, bt)

5. Materyal ve Metot

5.1. Veri

Bu çalışmada Finance Yahoo sitesinden veri setinde S&P veri seti kullanılmıştır. Verilerin open, high, low, close ve adjust gibi değerleri vardır. Tezde ise adjust değerleri kullanılarak işlemler yapılmıştır. Veriler 12.08.2000 ile 13.08.2020 arasındaki günlük verileri içerir.

5.1.2 Open-Close- High Değerleri

Finansal verilerin analizinde, akıllı tahmin sistemlerinin kullanılmasına yol açan araçlar veya finansal istatistikçiler tarafından sıklıkla kullanılan en önemli araç teknik analiz yöntemleridir. Bu yöntemler aynı zamanda makine öğrenme algoritmalarının temeli olan matematiksel algoritmalara ve sistemlere dayanmaktadır. Tüm bu istatistiksel unsurlara ek olarak, verileri anında izlemek ve görselleştirmek vazgeçilmezdir. Bu süreçte kullanılan tek görsel araç, finansal unsurların grafiksel görsel sunumlarıdır. Örnekler arasında Şamdan Grafiği, Çubuk Grafik veya OHLC grafikleri veya Çizgi Grafikler bulunur.

Bu ifadeler, anlık olarak izlenen finansal unsurdaki değişikliği okumayı kolaylaştıran bir yapıdır. Şamdan grafikleri en çok tercih edilen, ifade edilebilir parametreler ve basitlik ile değerlendirildiğinde en iyisidir.

İlgili finansal unsurun bir biriminin eyleminde oluşan durumun ifadesi için kullanılan noktalardan bahsedilmektedir. Bu birim, belirli bir çalışma zamanında ortaya çıkan etkiyi özetler.

High, Dönem içinde ulaşılan en yüksek değer,

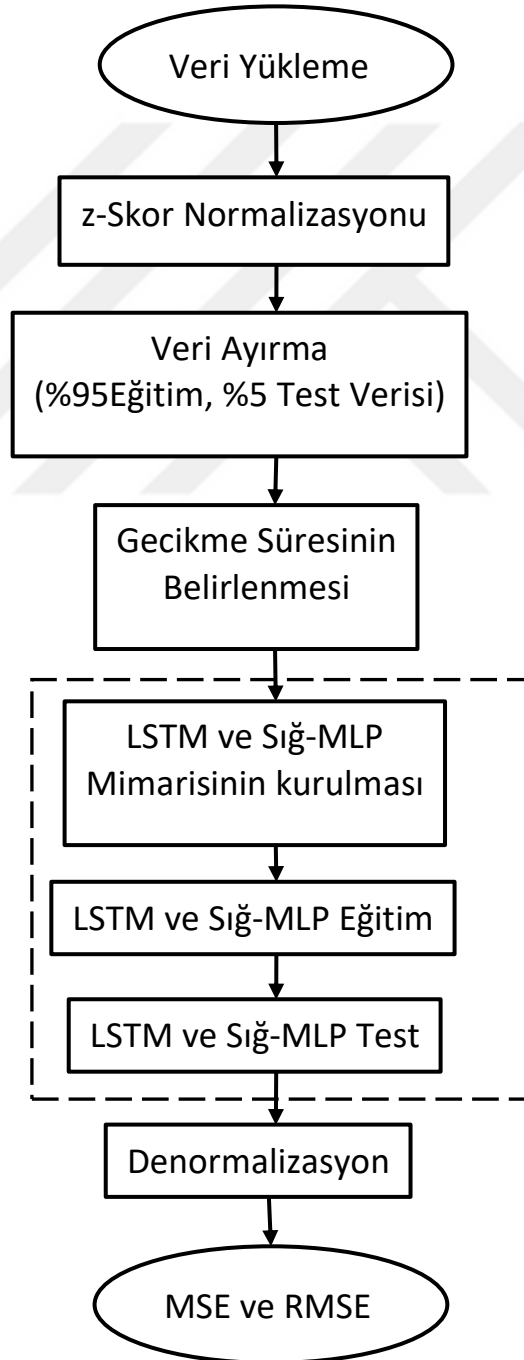
Open İşlem süresinin başlangıç zamanında aldığı ilk değerdir.

Close dönemin sonunda alınan son değeri ifade eder.

Low dönem içinde alınan en düşük değeri ifade eder (Bayındır,2017).

5.2 Metot

Çalışmada Yahoo finans verileri üzerine geliştirilen algoritma Şekil 5.1’de verildiği gibidir. Geliştirilen algorithmanda öncelikle finans verisi sisteme yüklenmiş daha sonra z-Skor Normalizasyonu uygulanarak veriler -1 ile +1 değerleri arasına getirilmiştir. Çalışmada kullanılan LSTM ve Sığ MLP ağlarının eğitimi ve eğitilmiş ağların test aşamaları için veri setinin başlangıçtan itibaren %95’lik kısmı eğitim ve %5’lik kısmı test için ayrılmıştır.



Şekil 5.1 Tasarlanan Algoritma

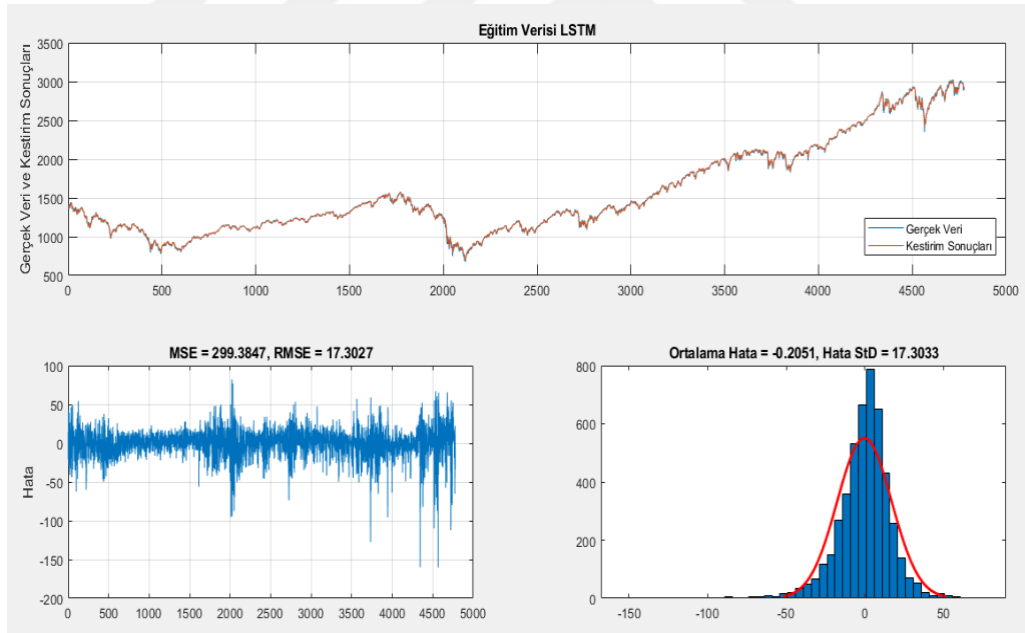
LSTM ve Sığ MLP ağlarının girişleri için Yahoo finans veri serisinin 11 farklı gün gecikme değerleri kullanılmıştır. Bu girişler 1-5, 6, 10, 11, 15, 16 ve 20. gün olacak şekilde deneysel olarak belirlenmiştir. Gecikme seçimi tasarlanırken tahmini yapılacak günün önceki 5 günü, iki, üç ve dördüncü haftalarından da aralarında beşer gün olacak şekilde iki seçim yapılmıştır.

LSTM mimarisinde iki gizli katman ve her gizli katmanlarda 300 nöron olacak şekilde tasarlanmıştır. Seyreltme değeri (dropout) 0.2 olarak seçilirken batch boyutu 120 ve iterasyon sayısı 1000 olarak belirlenmiştir. Sığ MLP mimarisinde ise gizli katman sayısı iki olarak belirlenmiş (birinci gizli katman 15 nöron, ikinci gizli katman 30 nöron) maksimum iterasyon sayısı 200 olarak seçilerek eğitim için Bayes Düzenlemeli gariyayılım (Bayesian regularization packpropagation) algoritması kullanılmıştır. Eğitim ve test aşamalarından elde edilen çıktıların z-skor denormalizasyonu uygulanarak elde edilen sonuçlar ortalama karesel hata (MSE) ve kök ortalama karesel hata (RMSE) cinsinden karşılaştırılmıştır.

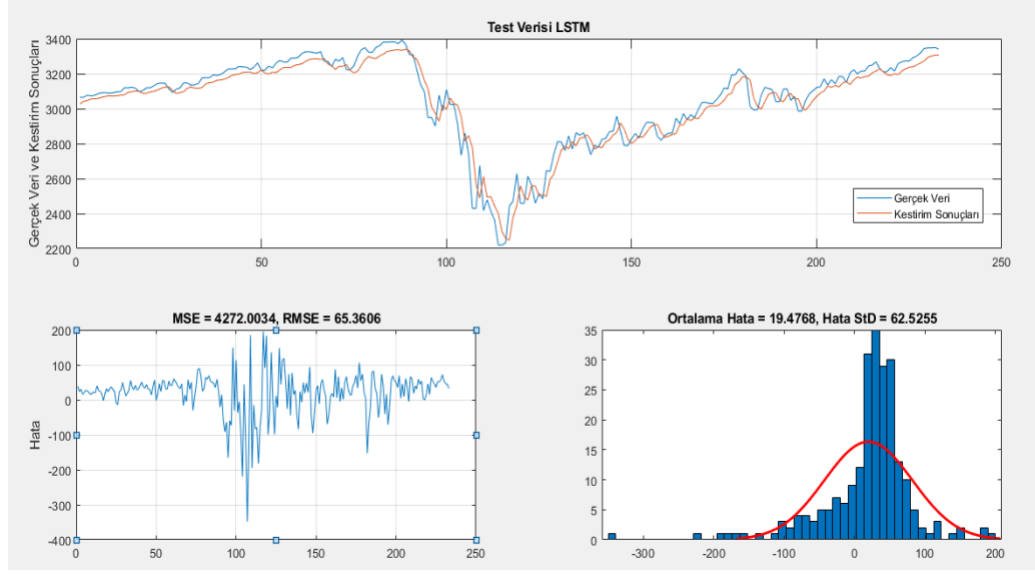
6. Sonular ve Tartışma

6.1 Sonular

alıřmada elde edilen sonular eđitim, test ve tm veri olarak ayrı ayrı incelenmiřtir. LSTM mimarisi ile elde edilen sonuların MSE ve RMSE deđerleri, ayrıca hatanın histogram ortalama ve standart sapma sonuları eđitim iin řekil 6.1'te, test iin řekil 6.2'de ve tm veri iin řekil 6.3'te verilmiřtir. řekil 6.1 incelendiđinde, ortalama karesel hatanın 300 dolar civarında olduđu ve hatanın ođunlukla -50 dolar ile 50 dolar arasında dađılım gsterdiđi gzlemlenmektedir. řekil 6.2'de verilen test bařarisında ise ortalama karesel hatanın 4727 dolar civarında olduđu ve en byk hatanın Kovid srecinin bařladıđı tarih olan 2020 Mart ayı civarında ortaya ıktıđı grlmektedir. Eđitim ve test sonuları bir arada incelendiđinde histogram eđrilerinden eđitim ařamasında ortalama hatanın 0'a yakın olduđu ve standart sapma deđerinin 17 dolar civarı olduđu test ařamasında ise ortalama hata 19 dolara yakın iken standart sapma hatasının 62 dolar mertebesinde deđiřtiđi sylenebilmektedir.

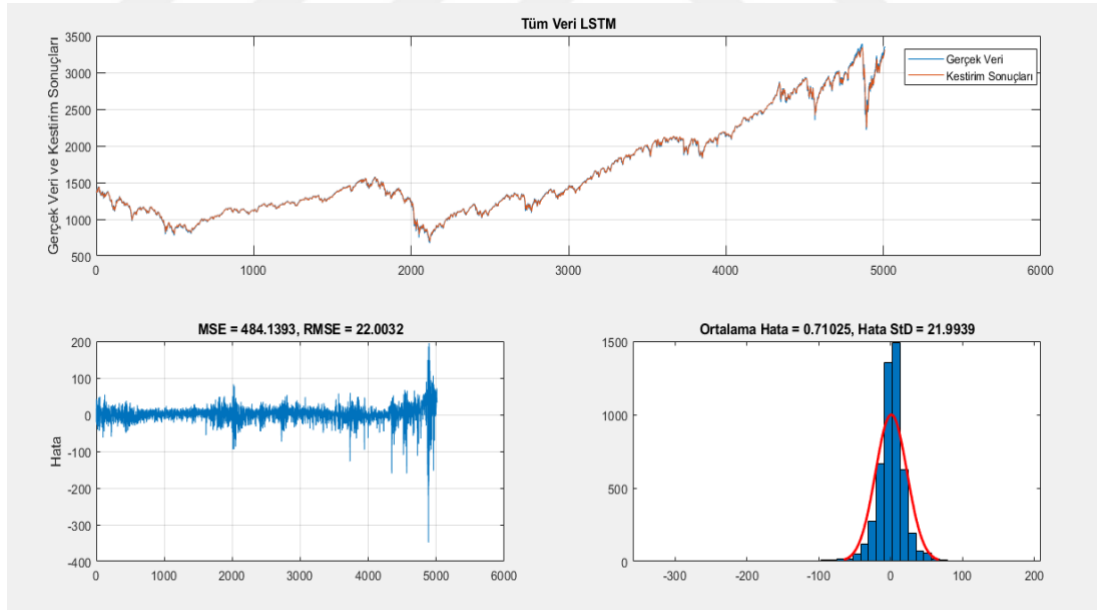


řekil 6.1 LSTM eđitim sonuları



Şekil 6.2 LSTM Test sonuçları

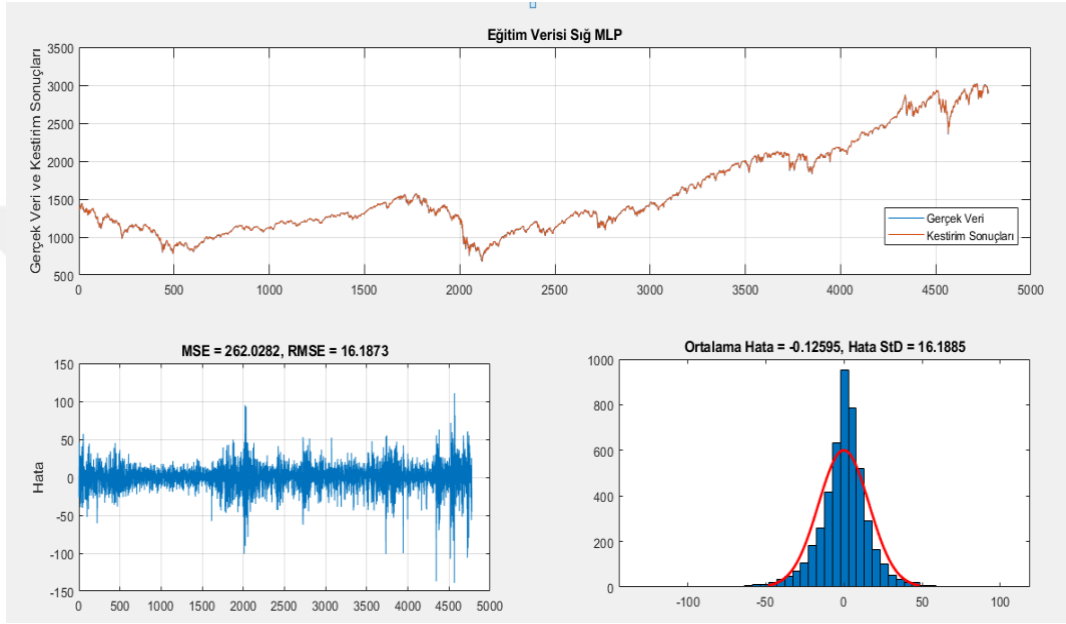
LSTM mimarisi sonucunda elde edilen eğitim ve test aşamaları birleştirildiğinde ise kök ortalama hatanın 22 dolar mertebesine yakın olduğu ve hata histogram tablosundan ortalama hatanın 0, 7 dolar civarında ve standart sapmanın 22 doları geçmediği görülmüştür.



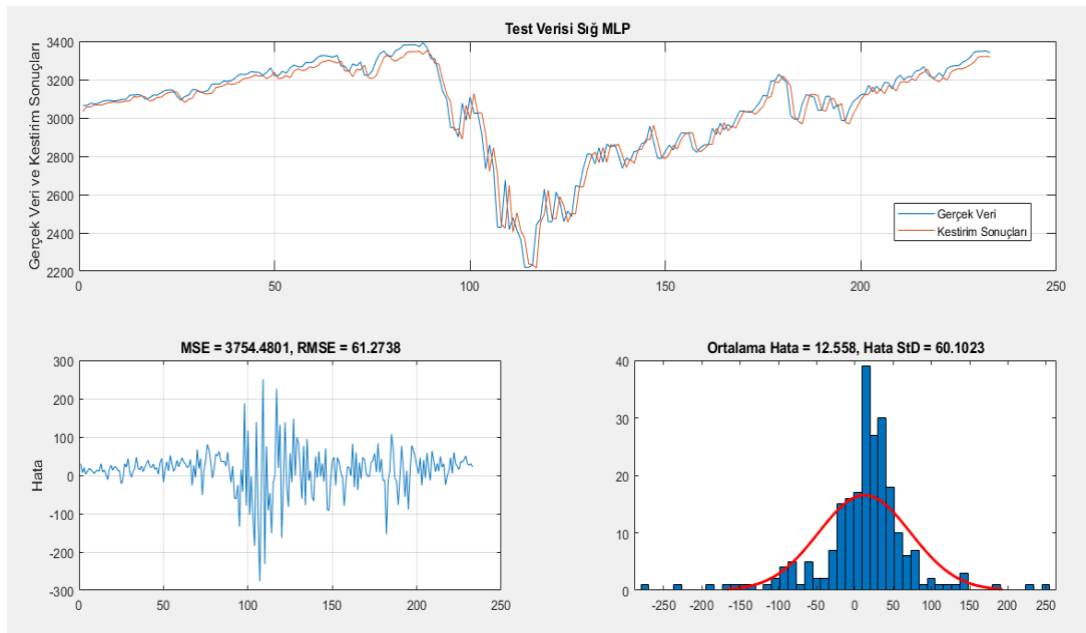
Şekil 6.3: LSTM Tüm veri sonuçları

Sığ MLP mimarisi ile elde edilen sonuçların MSE ve RMSE değerleri, ayrıca hatanın histogram ortalama ve standart sapma sonuçları eğitim için Şekil 6.4'te, test için Şekil 6.5'te ve tüm veri için Şekil 6.6'da verilmiştir. Şekil 6.4 incelendiğinde, ortalama karesel hatanın 260 dolar civarında olduğu ve hatanın çoğunlukla -40 dolar ile 40 dolar

arasında dağılım gösterdiği görülmüştür. Şekil 6.5'te verilen test başarısında ise ortalama karesel hatanın 3755 dolar civarında olduğu ve yine en büyük hatanın Kovid sürecinin başladığı tarih olan 2020 Mart ayı civarında ortaya çıktığı görülmektedir. Eğitim ve test sonuçları bir arada incelendiğinde histogram eğrilerinden eğitim aşamasında ortalama hatanın 0 dolara yakın olduğu ve standart sapma değerinin 16 dolar civarı olduğu test aşamasında ise ortalama hatanın 12,5 dolara yakın olduğu fakat standart sapma hatasının 60 dolar mertebesine yükseldiği görülebilmektedir.

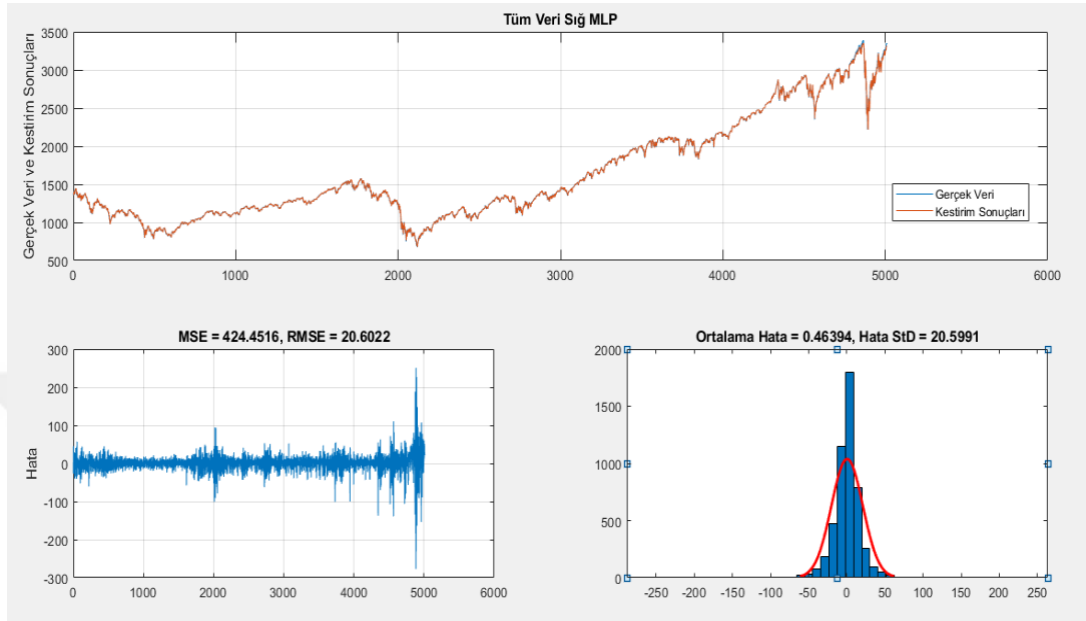


Şekil 6.4 Sığ MLP Eğitim sonuçları



Şekil 6.5: Sığ MLP Test sonuçları

Sığ MLP mimarisi sonucunda elde edilen eğitim ve test aşamaları bir arada incelendiğinde kök ortalama hatanın 20 dolar mertebesine yakın olduğu ve hata histogram tablosundan ortalama hatanın 0,5 dolar civarında ve standart sapmanın ise 20,5 dolar hesaplandığı görülmektedir.



Şekil 6.6 Sığ MLP Tüm veri sonuçları

Tasarlanan mimariler sonucunda elde edilen hata değerlerinin karşılaştırıldığı Tablo 1 incelendiğinde, eğitim başarısı açısından Sığ MLP mimarisinin LSTM mimarisine göre daha başarılı bir kestirim sonucu elde edildiği görülmektedir. Eğitilmiş ağların doğrulanması için gerçekleştirilen test adımında MSE ve RMSE değerleri açısından LSTM mimarisinin Sığ MLP'ye yakın değerler sağladığı görülmektedir. Sığ MLP ve LSTM ağlarının eğitim başarıları RMSE değerleri açısından karşılaştırıldığında Sığ MLP değerinin yaklaşık olarak 1 dolar kadar daha az hata yaptığı görülürken test aşamasında ise Sığ MLP kestirim başarısı yaklaşık olarak 5 dolar kadar daha iyi sonuç vermektedir. Eğitim ve Test adımları bir arada göz önüne alındığında ise Sığ MLP mimarisinin 1,5 dolar kadar daha iyi sonuç verdiği görülmektedir.

Tablo 1: LSTM ve Sığ MLP yapılarının hata karşılaştırması

HATA (\$)	Tüm Veri		Eğitim Verisi		Test Verisi	
	LSTM	Sığ MLP	LSTM	Sığ MLP	LSTM	Sığ MLP
MSE	484,14	424,45	299,38	262,08	4272.0	3754,48
RMSE	22,0	20,60	17.30	16.19	65.36	61,27

Tablo 2’ de her iki mimarinin hata değerlerinden elde edilen hata histogram sonuçları karşılaştırıldığında; Eğitim için beklenen değer (ortalama hata) sonuçlarının her iki mimari içinde yaklaşık olarak 0’a yakın olduğu gözlemlenmiştir. Test verisi için ise ortalama hata olarak MLP mimarisinin yaklaşık olarak 7 dolar kadar daha düşük olduğu görülmektedir. Her iki mimari standart sapma açısından incelendiğinde LSTM ve Sığ MLP mimarilerinin benzer bir dağılıma sahip olduğu görülmektedir. Doğrulama adımında her iki mimari karşılaştırılırsa, Sığ MLP ve LSTM mimarisinin yaklaşık 20 dolarlık bir sapma değerine sahip olduğu hesaplanmıştır.

Tablo 2: LSTM ve Sığ MLP hata histogram sonuçları

HATA HİSTOGRAM	Tüm Veri		Eğitim Verisi		Test Verisi	
	LSTM	Sığ MLP	LSTM	Sığ MLP	LSTM	Sığ MLP
Ortalama Hata	0.710	0.463	-0.205	-0.1259	19.476	12.558
Hata std	21.99	20.599	17.303	16.188	62.525	60.1023

6.2 Tartışma

Literatürde S&P veri seti ile gerçekleştirilen çalışmalar incelendiğinde farklı algoritmalar ve farklı zamanlardan elde edilen zaman serisi için gerçekleştirilen çalışmalar mevcuttur. Khaled ve arkadaşları 1 Ocak 2010 ile 30 Kasım 2017 arasındaki tarihlerden oluşturdukları S&P veri setini %80 eğitim ve %20 test verisi olarak kullanmış ve Bi-LSTM ve LSTM mimarileri ile gerçek değer kestirim işlemi gerçekleştirmişlerdir. Elde ettikleri sonuçlar incelendiğinde eğitim başarısı açısından 24.452 RMSE ve 0,0119 MAPE değerleri ile en iyi sonucu Bi-LSTM mimarisi ile hesapladıkları görülmektedir. Test verisi açısından yine Bi-LSTM mimarisinin %30 oranında daha düşük bir RMSE değeri sağladığı görülmektedir [16]. Önerilen çalışmada elde edilen sonuçlar ile Khaled ve arkadaşlarının elde ettiği sonuçlar incelendiğinde eğitim işlemi elde edilen RMSE değerinin (MLP için 16.187 ve LSTM için 17.302) daha az hatalı olduğu Tablo 3’ten görülmektedir. Test verisi açısından her iki çalışma karşılaştırıldığında ise LSTM mimarilerinin benzer sonuçlar verdiği gözlemlenmektedir. Cao ve arkadaşları ise 13 Aralık 2007 ve 12 Aralık 2017 tarihleri arasında oluşturdukları 10 senelik veri setini %90 eğitim ve %10 olacak şekilde bölüp farklı algoritmalar ile kestirim sonuçlarını hesaplamış; tüm veri için 18,2 ile 4.8 arasında değişen RMSE değerleri bulmuşlardır [17]. Aynı tarih aralığında Li ve arkadaşlarının gerçekleştirdiği çalışmada ise (9 Kasım 2007 ile 25 Nisan 2017 tarihleri arası) sadece tüm veri seti için MAPE değerleri hesaplanmıştır [1]. Çalışmaların gerçekleştirildiği zaman aralığı incelendiğinde bu süreçte dünya genelinde herhangi bir kriz süreci yaşanmadığı ve S&P borsa verisinin düzenli

artış gösterdiği gözlemlenmiştir. Önerilen çalışmada RMSE ve MAPE değerini oldukça düşüren iki kriz faktörü bulunmaktadır. Bunlardan birincisi 2018’de yaşanan ekonomik kriz (Şekil 6.3 ve Şekil 6.6)tüm veri grafiği 4500. Adım civarı) ve Mart 2020 tarihli etkisini gösteren kovid sürecidir (4.800. adım civarında 3400 \$ mertebesinden 2200 \$ mertebesine düşüş). Yaşanan bu süreçlere rağmen MLP ve LSTM mimarileri ile elde edilen MAPE değerleri (Sığ MLP 0.00815 ve LSTM 0,00897) Bu çalışmalara oldukça yakındır.

Tablo 3: Literatür Karşılaştırması

	Kullanılan Algoritma	Eğitim Verisi		Test Verisi		Tüm Veri	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Khaled ve diğerleri [59]	BLSTM	24,452	0,0119	46,409	0,02302		
	LSTM	26,418	0,0130	61,393	0,03100		
Cao ve diğerleri [57]	LSTM					18,232	0,00611
	SVM					10,244	0,00399
	CEEMDAN-SVM					6,8118	0,00221
	CEEMDAN-LSTM					6,1412	0,00205
	CEEMDAN-MLP					4,8291	0,00161
Li ve diğerleri [71]	Linear Regression						0,0747
	Ridge Regression						0,0747
	Random Risk Management						0,0765
	via Anomaly						0,0742
	Circumvent Forest LSTM						
Kılıç ve diğerleri [70]	ARIMA	12.015	0,0766	33,293	0,28405		
Önerilen Çalışma	MLP	16,187	0,0078	61,273	0,01447	20,602	0,00815
	LSTM	17,302	0,0085	65,360	0,01679	22,003	0,00897

Kılıç ve Arkadaşlarının gerçekleştirdiği çalışmada ise 1 Ocak 1990 ile 1 Ocak 2011 arasındaki S&P borsa verisinden dalgacık katsayıları çıkartılarak ARIMA regresyon yöntemi kullanılmıştır. Önerilen çalışma ile Kılıç ve arkadaşlarının elde ettikleri RMSE sonuçları karşılaştırıldığında eğitim hatasının önerilen çalışmaya göre yaklaşık 5 dolarlık bir hata yaptığı gözlemlenmektedir. Fakat burada Kılıç ve arkadaşları tarafından eğitim için kullanılan veride borsa değerleri 1000 dolar civarında iken önerilen çalışmada borsa değeri 2000 dolar ile 3000 dolar mertebesinde değişmektedir. Bu neden ile dolar bazlı olarak gerçekleştirilen hata hesaplamasında değerler göz önünde bulundurulmalıdır.

Önerilen çalışmada tüm veri üzerinde başarıyı en çok etkileyen faktör test verisinin kestirilmesinde ortaya çıkmaktadır. Bunun nedeni küresel ölçekte yaşanan ve tüm dünya borsalarının %30 civarında değer kaybetmesine neden olan kovid sürecidir (Şekil 6.2 ve Şekil 6.5). Çalışmada önerilen algoritmalar bu süreçte kestirim adımını atlamayarak sadece artış trendi gösteren zaman dilimleri için değil, anlık düşüş ve yükseliş gösteren trendler için de başarılı bir kestirim tahmininde bulunduğu söylenebilir.

7. Kaynakçalar

- A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. “Artificial Neural Networks: A Tutorial”. In: *Computer* 29.3 (Mar. 1996). 02276, pp. 31–44. DOI: 10. 1109/2.485891
- Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018, April). Evaluation of bidirectional lstm for short-and long-term stock market prediction. In 2018 9th international conference on information and communication systems (ICICS) (pp. 151-156). IEEE.
- Arisoy, E., Sethy, A., Ramabhadran, B., & Chen, S. (2015, April). Bidirectional recurrent neural network language models for automatic speech recognition. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5421-5425). IEEE.
- Aslay, F., & Özen, Ü. (2013). Meteorolojik parametreler kullanılarak yapay sinir ağları ile toprak sıcaklığının tahmini. *Politeknik Dergisi*, 16(4), 139-145.
- Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE, DOI: 10.1109/72.279181, March 1994.
- Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). An overview and comparative analysis of recurrent neural networks for short term load forecasting. arXiv preprint arXiv:1705.04378.
- Brunette, Emma & Flemmer, Rory & Flemmer, Claire. (2009). A review of artificial intelligence. ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents. 385-392. 10.1109/ICARA.2000.4804025.
- Buchanan, Bruce. (2005). A (Very) Brief History of Artificial Intelligence.. *AI Magazine*. 26. 53-60
- Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*, 519, 127-139.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-

decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.)
computational abilities,” Proc. Natl. Acad. Sci., vol. 79, pp. 2554-2558, Apr. 1982

D. Maclaurin, D. Duvenaud, and R. Adams, “Gradient-based hyperparameter optimization through reversible learning,” in International Conference on Machine Learning, 2015, pp. 2113-2122)

D. Maclaurin, D. Duvenaud, and R. Adams, “Gradient-based hyperparameter optimization through reversible learning,” in International Conference on Machine Learning, 2015, pp. 2113-2122

D.O. Hebb, The Organization of Behavior: A Neuropsychological Theory, John Wiley Ed., New York, 1949

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends® in Signal Processing, 7(3–4), 197-387

Deng, Li. (2011). An Overview of Deep-Structured Learning for Information Processing. APSIPA ASC 2011 - Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2011.

Elmas, Ç. (2007). Yapay Zeka Uygulamaları. Ankara: Seçkin Yayıncılık

F. Althé and A. de La Fortelle, “An LSTM network for highway trajectory prediction,” 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 353-359.

F. Rosenblatt, “The perceptron: A Perceiving and recognizing automation,” Cornell Aeronautical Laboratory Report, 85-460-1, 1957

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669.

Graves, Alex et al. “Rapid Retraining on Speech Data with LSTM Recurrent Networks.” (2005).

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München, 91(1.)

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

http://kisi.deu.edu.tr/hamdi.emec/UygEko/ZAMAN_SERI_ANALIZINDE_TEMEL_KAVRAMLAR.pdf

<http://www.mku.edu.tr/files/339-8868446a-e4c0-4070-8530-b8beccc1b0ec.pdf>

<https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/>

Huang, P. S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2014, October). Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks. In *ISMIR* (pp. 477-482)

J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in neuroscience*, vol. 10, p. 508, 2016

J. Hopfield, "Neural networks and physical systems with emergent collective

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, June). An empirical exploration of recurrent network architectures. In *International conference on machine learning* (pp. 2342-2350)

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, June). An empirical exploration of recurrent network architectures. In *International conference on machine learning* (pp. 2342-2350)

K. Bayındır, "Financial Time Series Prediction With LSTM Recurrent Neural Networks," Yüksek lisans tezi, Bilgisayar mühendisliği bilimi, Fen Bilimleri Enstitüsü, Bahçeşehir Üniversitesi, İstanbul, 2017

Kılıç, D. K., & Uğur, Ö. (2018). Multiresolution analysis of S&P500 time series. *Annals of Operations Research*, 260(1-2), 197-216.

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103, 25-37.

Kim, Sangyeon & Kang, Myungjoo. (2019). Financial series prediction using Attention LSTM.

- Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PloS one*, 14(2).
- Kriesel, A Brief Introduction to Neural Networks, 2005
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, 11(7), 1387.
- Li, X., Li, Y., Liu, X. Y., & Wang, C. D. (2019). Risk management via anomaly circumvent: Mnemonic deep learning for midterm stock prediction. *arXiv preprint arXiv:1908.01112*.
- M. Minsky, S. Papert, *Perceptrons: An Introduction to Computational Geometry*, 3rd ed. MIT Press Ed., Cambridge 0-262-631113, 1987
- Manurung, A. H., Budiharto, W., & Prabowo, H. ALGORITHM AND MODELING OF STOCK PRICES FORECASTING BASED ON LONG SHORT-TERM MEMORY (LSTM).
- Marini, F., Bucci, R., Magrì, A. L., & Magrì, A. D. (2008). Artificial neural networks in chemometrics: History, examples and perspectives. *Microchemical journal*, 88(2), 178-185.
- Mijwil, Maad & Esen, Adam & Alsaadi, Aysar. (2019). Overview of Neural Networks. 1. 2.
- Mijwil, Maad. (2015). History of Artificial Intelligence. 3. 1-8.)
- Mikolov, Tomáš. *Statistical Language Models based on Neural Networks*. PhD
- Montufar, G. F., Pascanu, R., Cho, K., & Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in neural information processing systems* (pp. 2924- 2932).
- Olah, C. Understanding LSTM Networks. Available online: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed on 26 June 2018)
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

- Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. arXiv preprint arXiv:1804.02763.
- Q. Wang, Y. Guo, L. Yu and P. Li, "Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach," in *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 148-158, 1 Jan.-March 2020.
- Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, 6(4), 1754-1756.
- Serwa, A. (2017). Studying the Effect of Activation Function on Classification Accuracy Using Deep Artificial Neural Networks. *Journal of Remote Sensing & GIS*, 6(03), 6-11.)
- Sharma, S. (2017). Activation functions in neural networks. *Towards Data Science*, 6.
- Siame-Namini, S., & Namin, A. S. (2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM (No. 1803.06386
- Siame-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. arXiv preprint arXiv:1803.06386
- Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, 47(3), 1264-1268.
- Steel, A. (2014). Predictions in Financial Time Series Data.
- Subaşı, H. (2010). Yapay Sinir Ağları İle Atık Su Arıtma Performansının Modellenmesi. Adana: Çukurova Üniversitesi.
- Thesis, PhD thesis, Brno University of Technology, 2012. -, 2012
- Understanding LSTM Networks. Retrieved 30.01.2018 from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- W.S. McCulloch, W.H. Pitts, A logical calculus of the ideas imminent in nervous activity, *Bull. Math. Biophys.* 5 (1943) 115–133
- Wang, J. Q., Du, Y., & Wang, J. (2020). LSTM based long-term energy consumption prediction with periodicity. *Energy*, 117197.

- Wang, L., & Fu, K. (2007). Artificial neural networks. Wiley Encyclopedia of Computer Science and Engineering, 181-188.).
- WANG, Shuohang and JIANG, Jing. Learning natural language inference with LSTM. (2016). NAACL HLT 2016: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: San Diego, California, 2016 June 12-17. 1442- 1451.
- Y. Duan, Y. L.V. and F. Wang, "Travel time prediction with LSTM neural network," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 1053-1058.
- Yan, S. (2016). Understanding LSTM and its diagrams. MLReview. com.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015
- Yao, X. (1993). Evolutionary artificial neural networks. International journal of neural systems, 4(03), 203-222
- Zhao, J., Qu, H., Zhao, J., & Jiang, D. (2018). Towards traffic matrix prediction with LSTM recurrent neural network . Electronics Letters, 54(9), 566–568.
- Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11(2), 68-75.
- Zheng, H., Yang, Z., Liu, W., Liang, J., & Li, Y. (2015, July). Improving deep neural networks using softplus units. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-4). IEEE.
- Zhuge, Q., Xu, L., & Zhang, G. (2017). LSTM Neural Network with Emotional Analysis for prediction of stock price. Engineering letters, 25(2).



ÖZGEÇMİŞ

Ad : Ali İsa

Soyad : TAŞ

Eğitimi:

- İstanbul Arel Üniversitesi- Elektrik ve Elektronik Mühendisliği 2011-2016
- İstanbul Arel Üniversitesi- Fen Bilimleri Enstitüsü Yüksek lisans 2018-2020

Yabancı Dil:

- İngilizce
- Almanca

Özgeçmiş:

- Turkcell-Assistant Software Test Specialist via Testinium
- Triodor Software- Quality&Assurance Test Engineer
- Teknasyon - Software Test Specialist

