



Kahramanmaraş Sutcu Imam University

Journal of Engineering Sciences



Geliş Tarihi : 02.10.2023

Kabul Tarihi : 06.12.2023

Received Date : 02.10.2023

Accepted Date : 06.12.2023

ZAMAN SERİLERİ TAHMİNLEME ALGORİTMALARI İLE KONTÖR TÜKETİM TAHMİNLEMESİ VE KARŞILAŞTIRMALI UYGULAMASI

CREDIT CONSUMPTION ESTIMATION AND COMPARATIVE APPLICATION WITH TIME SERIES ESTIMATION ALGORITHMS

*Hakan AKÇAY*¹ (ORCID: 0000-0001-9004-8179)*Derya YILTAS-KAPLAN*^{1*} (ORCID: 0000-0001-8370-8941)¹ İstanbul Üniversitesi-Cerrahpaşa, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

*Sorumlu Yazar / Corresponding Author: Derya YILTAS-KAPLAN, dyiltas@iuc.edu.tr

ÖZET

Dijital dönüşümün hızla yaygınlaşması ile işlenen verilerin boyutları ve hacimleri de artmıştır. Büyük verileri işlemek, doğruluğu yüksek analizleri kısa sürede ve daha az kaynak kullanarak yapmak için yeni yöntem ve algoritmalar geliştirilmiştir. Bu çalışmada makine öğrenmesi ve derin öğrenme tekniklerinden ARIMA (Oto-regresif Entegre Hareketli Ortalama), SARIMA (Mevsimsel ARIMA), Prophet (Facebook), XGBoost (En Büyük Eğitim Artırma), LSTM (Uzun-Kısa Süreli Bellek), RNN (Yinelemeli Sinir Ağı) ve GRU (Geçitli Tekrarlayan Birim) algoritmaları kullanılarak tüketicilerin kontör tüketimleri zaman serileri yardımıyla tahmin edilmeye çalışılmıştır. Modelleri karşılaştırmak için MAE (Ortalama Mutlak Hata), MAPE (Ortalama Mutlak Yüzde Hata), RMSE (Kök Ortalama Karese Hata) ve Determinasyon Katsayısı (R^2) kullanılmıştır. Elde edilen ölçüm sonuçlarına göre zaman serileri tahminlemede derin öğrenme tekniklerinin makine öğrenmesi yöntemlerine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Çalışmada zaman serileri tahminlemesi üzerine dokuz farklı makine ve derin öğrenme yöntemi kullanılarak kapsamlı bir inceleme yapılmıştır. Literatürdeki benzer çalışmalar ile kıyaslandığında bu çalışmada konu oldukça geniş bir perspektiften incelenmiştir.

Anahtar Kelimeler: Derin öğrenme, gru, lstm, makine öğrenmesi, zaman serileri tahminleme

ABSTRACT

With the rapid spread of digital transformation, the sizes and volumes of the processed data have also increased. New methods and algorithms have been developed to process big data and perform high-accuracy analyzes in a short time and using less resources. In this study, by using machine learning and deep learning techniques such as ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal ARIMA), Prophet (Facebook), XGBoost (Extreme Gradient Boosting), LSTM (Long-Short Term Memory), RNN (Recurrent Neural Network) and GRU (Gated Recurrent Unit), it has been tried to estimate the credit consumption of consumers with the help of time series. Also, three hybrid models were implemented by using RNN, LSTM and GRU. MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), RMSE (Root Mean Squared Error) and Coefficient of Determination (R^2) were used to compare the model performances. According to the results, it has been observed that deep learning gives better results than machine learning. This study is a comprehensive review of time series forecasting using nine different machine and deep learning methods. Compared to similar studies in the literature, this study examined the subject from a very broad perspective.

Keywords: Deep learning, gru, lstm, machine learning, time series forecasting

GİRİŞ

Veri bilimcileri büyük ve karmaşık veri kümelerini analiz etmek, videolardan, seslerden, metinlerden veya resim verilerinden yararlanarak genellikle insanlardan daha hızlı çıktı üretmek için çeşitli yapay zekâ yöntemlerini ayrı ayrı veya birlikte kullanabilirler. Bu yöntemler arasında makine öğrenmesi ve derin öğrenme teknikleri önemli yer tutmaktadır. Verilerin boyutu ve hacmi arttıkça, günümüzdeki bilgisayarların maliyetleri ucuzladıkça ve veri işleme kapasiteleri arttıkça şirketler de perakende, sağlık, finans, üretim, lojistik gibi tüm sektörlerde yeni fırsatlar elde etmek, veriden değer üreterek rakiplerinin karşısında bir adım öne geçmek için yapay zekâ tekniklerini daha fazla kullanmakta ve bu alana yapılan yatırımlar gün geçtikçe artmaktadır. Makine öğrenmesi ve derin öğrenme, veri işleme ve analiz algoritmalarını kapsayan yapay zekânın alt çalışma alanlarıdır.

Zaman serileri verileri, makine öğrenmesi ve derin öğrenme algoritmalarının kullanıldığı ve tahminleme çalışmaları için uygun olan veri türlerindedir. Zamana göre belirli bir düzende sıralanan, periyodik zaman aralıkları boyunca dağılmış, eşit aralıklarla sıralanmış (saat, gün vb.) verilere zaman serileri verisi denir. Zaman serilerinde bilinmesi gereken bazı kavramlar vardır. Bunlar trend, mevsimsellik, döngüsel bileşen, durağanlık, fark alma, bağımlılık ve gürültüdür. Trend, serinin zaman içerisindeki artan veya azalan davranışını (eğilimini) gösterir. Trend eğilimi, artan, azalan veya durağan olabilir. Mevsimsellik, bir yıllık zaman aralığında tekrarlayan bir eğilim gösteren zaman serilerinde bulunur. Mevsimselliğe örnek olarak ısınma maliyetlerinin yazın düşerken kışın artması gösterilebilir. Hisse senedi fiyatlarındaki değişimler, artış ve düşüşler, ya da günlük yayımlanan döviz kuru oranları örnek olarak verilebilir. Döngüsel bileşenler ise mevsimsel bileşenler gibi belirli bir dönemde oluşurlar, ancak döngüsel bileşenlerin gerçekleştikleri dönemler sabit olmayan periyotlar şeklindedir. Durağanlık, serinin periyot boyunca artmadığını ve ortalama değerini sabit kaldığını gösterir. Kısaca zaman içinde belirli bir değere doğru yaklaşan sabit ortalamaya, sabit otokorelasyon yapısına ve sabit varyansa sahip olan seriler durağan serilerdir. Zaman serileri genellikle durağan değildir ve bu nedenle durağan hale getirmek için veriler üzerinden fark alma işlemine ihtiyaç duyulabilir. Çünkü durağan olmayan bir seri, zamanla farklı örüntüler sergileyebilir. Bu durumda serinin davranışı diğer dönemlere genellenemez ve tahminleme başarısını olumsuz etkiler.

Bu çalışmada geçmiş yıllara ait müşteri kontör tüketim verileri üzerinde makine öğrenmesi ve derin öğrenme algoritmaları çalıştırılarak kontör tüketim tahminlemesi yapılmıştır. ARIMA (Oto-regresif Entegre Hareketli Ortalama), SARIMA (Mevsimsel ARIMA), Prophet (Facebook), XGBoost (En Büyük Eğitim Artırma), LSTM (Uzun-Kısa Süreli Bellek), RNN (Yinelemeli Sinir Ağı) ve GRU (Geçitli Tekrarlayan Birim) algoritmaları kullanılarak tahminleme yapılmış, makine öğrenmesi ile derin öğrenme algoritmaları kıyaslanarak performans karşılaştırmasının yapılması amaçlanmıştır.

Çalışmanın devamı şöyle tasarlanmıştır: Derin öğrenme ile ilgili genel bilgi verildikten sonra zaman serileri tahminleme ile ilgili mevcut literatür çalışmalarından örnekler anlatılmaktadır. Yöntem bölümünde bu çalışma içerisinde kullanılan algoritmalar, veri seti, yazılım ve donanım bilgileri yer almaktadır. Bulgular bölümünde çalışma sonuçları ve elde edilen karşılaştırmalı grafikler gösterilmiştir. Son olarak Sonuç bölümünde ise çalışmanın sonuçlarının etkileri analiz edilmektedir.

Kapsam ve Literatür İncelemesi

Veri bilimi, teknik analiz yöntemlerini ve veri algoritmalarını kullanan birden fazla disiplini içeren bir alandır (Al-Asadi, 2019). Depolama sistemlerinin hacmi arttıkça, fiyatları düştükçe ve bilgisayarlar kapasite olarak daha güçlü ve fiyat olarak daha ucuz oldukça, büyük hacimli veriler de daha kolay saklanabilir hale gelmiştir (Amazon, 2023). Endüstriyel sektörler, finans, üretim, perakende, sağlık, taşıma, teknoloji sektörlerindeki gibi şirketler yenilik yapmak, yeni fırsatları yakalamak ve rakiplerini geçmek için derin öğrenmeye ihtiyaç duymaya başladılar ve yatırımlarını bu alana da kaydırmaya başladılar. Bu nedenlerden dolayı derin öğrenmenin hem popülerliği arttı hem de kullanım alanları genişledi (Dinçoğlu, 2022). Derin öğrenme temelde kendi içinde katmanları olan insan beynine benzer Yapay Sinir Ağı (YSA) ve makine öğrenme ile ilgili algoritmaları kapsar. YSA kullanılarak akıllı sistemlerin mevcut verilerden farklı veriler elde etmesi sağlanır (Güdelek, 2019).

Tahminleme, geçmişe dayalı verilerin analiz edilerek gelecek hakkındaki eğilimlerini ve değişim olasılıklarını belirleyen öngörü tekniğidir. İşletmeler, teknoloji şirketleri, bankalar, reklam şirketleri, finansal kuruluşlar ve e-ticaret platformları gibi kurumlar satış hacimlerini arttırmak için satış tahminlemesi yaparlar. Satış dışında, müşteri artışlarını, üretim hacimlerini, bütçelerini planlamak için tahminleme yöntemlerini kullanırlar (Microsoft, 2023). Tahminleme metodu, yöneticilerin işlerinde doğru kararlar almasına ve tüm planlamalarını doğru verilere dayanarak yapmasına olanak sağlar. Tahminler, analitik veriler üzerinden kararlar almalarına yardımcı olur. Yöntemler ve

veriler, çok değişkendir (Yaşar, 2019). Tahminleme süreçleri şirketlerin gelecek planlamasını etkilemesi açısından çok kritik öneme sahiptir.

Zaman serileri birçok alanda görülebilen verilerdir. Sinyal işlemeden ekonometriye ve istatistikten matematiksel finansa kadar birçok alandaki periyodik veriler zaman serileri formatındadır. Zaman serileri zamana göre değişen verileri içerir ve belirli bir formatta ve düzendedir. Zaman serisi üzerinde yapılan tahminleme ise geçmişte bilinen veriler ve olaylar üzerinde analizler yaparak gelecekteki olayları ve verileri tahmin etmek demektir (Kaya vd., 2020; Baş, 2019).

Bu bölümde son birkaç yılda yapılmış literatür uygulamaları incelenmiştir. Çalışmalar kronolojik sıraya göre ele alınmıştır. 2017 yılında yapılan bu çalışmada (Yang vd., 2017) durağan olmayan zaman serileri üzerinde ARIMA modeli kullanılarak uçak motoru yağlama yağının metal içerik tahmini yapılmıştır. ARIMA ve üstel yumuşatma modelleri kullanılmış ve ARIMA modelinin daha başarılı tahminlemeler yaptığı gözlemlenmiştir.

Zaman serilerini kullanarak döviz kurlarını derin öğrenme yöntemi ile tahmin etmeye yönelik yapılan çalışmada RNN, LSTM, CNN (Evrişimsel Sinir Ağı-Convolutional Neural Network) ve ARIMA yöntemleri kullanılarak saat bazında ve günlük bazda veri içeren veri setlerindeki üç adet döviz çiftinin (USD/TRY, EUR/TRY ve EUR/USD) kapanış oranları tahmin edilmeye çalışılmıştır (Al-Asadi, 2019). Modelin performans değerleri ise Ortalama Mutlak Yüzde Hata (MAPE) ile ölçülerek değerlendirilmiştir. Çalışma sonuçlarına bakıldığında CNN tabanlı modellerin diğer modellere göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Endüstri 4.0 için derin öğrenme yaklaşımlarının incelendiği bir çalışmada (Güdelek, 2019) ise endüstriyel zaman serileri kullanılarak süreçlerin iyileştirilmesinin mümkün olup olmadığı araştırılmıştır. Derin öğrenme modellerinden LSTM kullanılmıştır. Modelin performansı ise Ortalama Mutlak Hata (MAE) hesaplanarak ölçülmüştür. Diğer çalışmalara kıyasla bu çalışmadaki modelin ortalama ve standart sapma değerlerinin daha yüksek çıktığı gözlemlenmiş ve modelin katmanlarında iyileştirme ihtiyacı olduğu belirtilmiştir (Güdelek, 2019).

Bir başka çalışmada LSTM modeli kullanılarak Çorum iline ait 1983 ile 2018 yılları arasındaki güneş ışınımı verileri üzerinde tahminleme yapılmıştır. Veriler zaman serileri formatındadır. Bu çalışmada gelecek ışınım oranı tahminleme yapılmaya çalışılmıştır (Kara, 2019). Önerilen model Gradyan Güçlendirme, Karar Ağaçları, K-En Yakın Komşu Algoritması ve Rastgele Orman Regresyonu gibi makine öğrenme algoritmaları ile kıyaslanmıştır. Sonuçlara bakıldığında LSTM modeli daha başarılıdır. Optimizasyon algoritması olarak Adam, epoch (eğitim tur) sayısı 150, veri işleme batch size (yığın boyutu) 32 olarak tercih edilmiştir. Performans karşılaştırması ise MAE, Kök Ortalama Kare Hata (RMSE) ve MAPE, Determinasyon Katsayısı (Coefficient of Determination, R^2) değerleri ile ölçülmüştür. LSTM tüm bu metrikler bazında daha iyi sonuçlar üretmiştir.

Derin öğrenme yöntemleri ile zaman serisi tahminlemesini içeren farklı bir doktora tezi çalışmasında ise finansal zaman verisi üzerinde tahminleme yapılmıştır (Gündüz, 2019). Çalışmada Borsa İstanbul verileri kullanılmıştır. Dolar, altın ve birleşiminden oluşan ilk yüz endeks içinde bulunan hisse fiyatlarından yararlanılmıştır. LSTM ve CNN modelleri klasik finansal tahminleme teknikleri ile kıyaslanmıştır. Bu çalışmada derin öğrenme modelleri borsa tahmininde iyi sonuçlar vermiş ve diğer öğrenme modellerine göre iyi bir alternatif olmuştur. LSTM ve CNN modelleri lineer regresyon ve Destek Vektör Makineleri'ne (SVM) göre üstünlük sağlamıştır.

Endüstri 4.0 çalışmaları kapsamında yapılan başka bir çalışmada ise derin öğrenme modellerinin zaman serileri üzerinde uygulanması gösterilmiştir. Çalışmada LSTM modeli için en uygun parametrelerin bulunması hedeflenmiştir (Baş, 2019).

Yine benzer olarak 2019 yılında yapılan farklı bir çalışmada ise zaman serileri analizi ve derin öğrenme modelleri kullanılarak döviz kuru tahminlemesi için hibrit bir model önerilmiştir (Yaşar, 2019). Çalışma kapsamında CNN, RNN ve LSTM gibi popüler derin öğrenme modelleri kullanılarak hem Twitter üzerinde duygu analizi hem de istatistiksel metotlar kullanılarak tarihsel veriler üzerinde tahminleme yapılmıştır. Tahminleme algoritmaları olarak Basit Üstel Yumuşatma, Holt-Winters ve ARIMA modelleri kullanılmıştır. Performans sonuçlarına göre duygu analizinde LSTM ve GloVe modeli ile hibrit olarak inşa edilen model en başarılı sonuçlara sahiptir. Kur tahminlemesinde ise Holt-Winters modeli en iyi sonucu vermiştir.

Zhejiang Üniversitesi tarafından yapılan bir çalışmada yine LSTM modeline stokastik bağlantı özelliği eklenerek LSTM'nin önemli hesaplama maliyetini azaltmayı hedefleyen hibrit bir model ortaya konmuştur. Değerlendirme sonuçlarına göre bu yeni model LSTM modeline göre daha başarılı sonuçlar vermiştir (Hua vd., 2019).

2020 yılında yaşanan Covid-19 pandemi dönemi öncesi ve pandemi sonrasında Bitcoin üzerindeki fiyat dalgalanmalarının makine öğrenmesine ek olarak derin öğrenme teknikleri ile değerlendirildiği bir diğer çalışmada ise yine zaman serileri verilerinden yararlanılmıştır (Kaya vd., 2020). Bu çalışmada Covid-19 öncesi ve pandeminin sonrasında Bitcoin kripto para birimi üzerindeki kapanış fiyatları ile düşme yükselme trendi dikkate alınarak iki adet veri seti oluşturulmuştur. Veriler üzerinde uygulanan tahminleme ile veri sınıflandırma yöntemleri karşılaştırılarak bunların başarıları değerlendirilmiştir. ARIMA, LSTM ve SVM modelleri ayrı ayrı çalışılmıştır. Tüm modellere ait F1 Skoru, doğruluk, Matthews Korelasyon Katsayısı (MCC), ARIMA modeli için RMSE ve MAPE ölçümleri de ele alınarak performans karşılaştırılması yapılmıştır. Sonuçlara göre SVM pandemi öncesi veriler üzerinde, ARIMA modeli ise pandemi sonrası veriler üzerinde daha başarılı sonuçlar vermiştir. SVM ortalama 1,72 saniyede, ARIMA 7,63 saniyede ve LSTM ise 14,86 saniyede tahminleme işlemlerini gerçekleştirmişlerdir.

Avrupa ülkelerindeki Covid-19 vakalarının tahmini ve karşılaştırmalı analizi çalışmasında (Kırbaş vd., 2020) ARIMA, NARNN (Lineer Olmayan Özbağlanımlı Sinir Ağı-Nonlinear Autoregressive Neural Network) ve LSTM yaklaşımları karşılaştırılmıştır. Performans sonuçlarına göre LSTM modeli diğer modellere göre daha başarılı sonuçlar vermiştir. Benzer şekilde Hindistan Jammu Üniversitesi Bilgisayar Mühendisliği Bölümü tarafından Hindistan ve Amerika Birleşik Devletleri Covid-19 vaka tahminlemesi üzerine bir çalışma yapılmış ve LSTM, Çift Yönlü LSTM de denilen BiLSTM ve RNN modelleri karşılaştırılmıştır. Çalışma sonuçlarına göre LSTM modelinin diğer modellere göre bir aylık periyot için daha başarılı sonuçlar verdiği görülmüştür (Shastri vd., 2020).

Altın fiyatlarının tahminlemesinde XGBoost, MARS ve lineer regresyon modellerinin kullanıldığı bir başka çalışmada ise elde edilen sonuçlara göre XGBoost %99,6 başarı oranı ile kestirim yaparken, MARS modeli için bu oran %97,8 olmuştur. Lineer regresyon için oran %94,3 olarak ölçülmüştür. Görüldüğü gibi XGBoost ve MARS modellerinin ikisi de lineer regresyon modeline göre daha iyi sonuçlar vermiştir. Diğer yandan XGBoost zaman yönünden diğer modellere göre daha kötü performans göstermiştir (Abar, 2020).

Bir başka çalışmada derin öğrenme ile zaman serileri tahminlemesi üzerine derinlemesine bir araştırma yapılmış, RNN, CNN ve LSTM ile olasılıksal yöntemler incelenmiştir (Lim & Zohren, 2021).

Elektrik talep tahmini yapılan bir çalışmada ise ARIMA ve XGBoost modeli karşılaştırılmıştır (Özdemir, 2021). ARIMA modelinin parametreleri ACF (Autocorrelation Function) ve PACF (Partial Autocorrelation Function) grafiklerinden faydalanılarak oluşturulmuştur. Modeller kıyaslandığında benzer sonuçlar verdikleri görülmektedir. Ancak XGBoost modelinin min, max, peak ve step parametreleri değiştirilerek istatistiksel modellere göre performansının kayda değer arttığı gözlemlenmiştir. MAPE değeri 0,10'dan 0,05'e gerilemiştir.

Farklı bir çalışmada ise LSTM modeli kullanılarak Bitcoin fiyat tahmini yapılmıştır (Tanışman vd., 2021). LSTM modeli istatistiksel model olan ARIMA modeli ile karşılaştırılmıştır. Sonuçlara göre ARIMA sadece yakın vadede başarılı tahminleme performansı sergilemiştir. Yakın ve uzun vadede gelecek tahminlemesi için kullanılan LSTM algoritması ise daha düşük hatalar ile tahminleme yapmıştır.

Fiber internet kullanan müşterilerin şikayetlerinin makine öğrenmesi teknikleri ile analiz edilerek müşteri memnuniyetinin araştırılması hedeflenen bir başka çalışmada ise Naive Bayes, Rastgele Orman, Karar Ağacı, XGBoost ve Lojistik Regresyon algoritmaları kullanılmıştır. Naive Bayes ve Lojistik Regresyon modellerinden oluşan hibrit modelin diğer modellere göre başarılı sonuçlar verdiği gözlemlenmiştir (Kayaş, 2021).

Bir yüksek lisans tezinde Türkiye'de yaygın kullanılan operatörlerin birinden alınan mobil uygulama kullanıcı giriş sayısı verileri üzerinde modeller uygulanmış ve sonraki dönemler için kullanım tahmini yapılmaya çalışılmıştır (İnce, 2021). Çalışmada ARIMA, Naive metodu, üstel düzleştirme, basit düzleştirme yöntemleri, SVM ve YSA kullanılmıştır. Performans değerlendirmesi için MAE metriği ölçülmüştür. Deney sonuçlarına göre SVM modeli en başarılı model olarak gözlemlenmiştir. Sonrasında ise YSA yöntemi ikinci başarılı model olarak ölçülmüştür. Benzer şekilde Bitcoin, Ethereum, Ripple gibi kripto para tahminlemesi için LSTM ve Prophet modellerini kullanan bir çalışma da mevcuttur (İndulkar, 2021). Facebook tarafından geliştirilen otomatik makine öğrenmesi algoritması olan Prophet mevsim verileri üzerinde oldukça başarılı sonuçlar vermiştir. Modellerin performansları MAE metriği ile ölçülmüştür (İndulkar, 2021).

Bir başka çalışmada ise veri madenciliği ve makine öğrenmesi yöntemleri kullanılarak döviz kuru tahmini yapılmaya çalışılmıştır (Erbudak, 2022). Tahmine dayalı sınıflandırma ve regresyon modelleri çalışılmıştır. Çalışmada USD/TRY parite verileri TCMB'nin satış döviz kuru değerlerinden alınarak elde edilmiştir. Performans metrikleri olarak Kök Ortalama Kare (RMS), R-Kare, Ortalama Kare ve MAE ile ölçülmüştür. Karar Ağaçları ve SVM'nin en yüksek doğruluk ve tahmin oranlarını verdiği gözlemlenmiştir. Çalışmadan elde edilen sonuçlara bakıldığında döviz kuru ve sapmaları ile tahmini değerlerin öngörülmesinde krizlerin önemli bir gösterge olabileceği çıkarılmıştır.

Bir diğer çalışmada ise perakende sektöründe veri madenciliği yöntemleri ile satış tahminlemesi yapılmıştır (Dinçoğlu, 2022). Özel bir perakende firmasının 2018 ve 2019 yıllarına ait satış verileri kullanılarak çıkan tahmin sonuçları karşılaştırılmıştır. Çalışmada Regresyon ve ARIMA modelleri kullanılmış ve çıkan sonuçlara göre regresyon modelinin hata oranı ARIMA modeline göre daha az olmuştur. Ancak çalışmanın genelinde her iki modelin de yakın hata oranlarına sahip olduğu gözlemlenmiştir.

2022 yılında yapılan bir başka çalışmada ise makine öğrenmesi modellerinden ARIMA modeline ek olarak derin öğrenme algoritmaları çalışılmıştır. Bu çalışmada Covid-19'la bağlantılı Türkiye'deki vaka ile vefat sayıları tahmin edilmeye çalışılmıştır. LSTM, BiLSTM ve GRU (Geçitli Tekrarlayan Birim) modelleri ile makine öğrenmesi modellerinden ARIMA modeli tercih edilmiştir (Nusrat, 2022). Model performansları için RMSE ve MAPE değerleri hesaplanarak karşılaştırma yapılmıştır. Elde edilen sonuçların gerçeğe yakın olduğu görülmüştür. Derin öğrenme yöntemlerinde optimizasyon algoritması olarak Adam fonksiyonu tercih edilmiştir. Sigmoid fonksiyonu ise aktivasyon fonksiyonu olarak seçilmiştir. Performans sonuçlarına göre LSTM modeli, BiLSTM ve GRU modellerine göre daha başarılı sonuçlar vermiştir. Ayrıca en hızlı sonucu LSTM modelinin ürettiği görülmüştür. ARIMA modelinin ürettiği sonuçlar ise derin öğrenme yöntemleri ile kıyasla gerçeğe daha uygun çıkmıştır. Derin öğrenme hata oranları, ARIMA modeli hata oranlarından daha yüksek olmuştur.

(Ejder, 2022)'deki doktora tezinde ise zaman serileri için veri madenciliği tabanlı düzenlenmiş CNN geliştirilmiştir. Finansal veriler üzerinden tahminleme yapılmaya çalışılmıştır. YSA olarak CNN, zaman serilerinin gelecekteki hareketlerini tahmin etmek için Hareketli Ortalama (MA), DBMEA modelinin performansını arttırmak için CART (Classification And Regression Trees) ve GBM (Gradient Boosting Machines) algoritmaları sırası ile RFE (Recursive Feature Elimination) ve SFS (Sequential Feature Selection) ile nitelik seçimi modellenmiştir. Bu modelde finansal özelliklerin seçilmesi için tercih edilmiştir. Tüm özellikler birleştirilmiş ve 2 boyutlu görüntüler elde edilmiştir. Görüntülerin elde edilmesinin ardından veriler CNN ile sınıflandırılmıştır. Çalışmanın sonuçları oldukça başarılı olmuştur. Buna göre CNN-DBMEA algoritması literatürdeki diğer yöntemlere kıyasla daha iyi sonuçlar vermiştir.

2022 yılında güneş enerji santrallerindeki aylık elektrik üretiminin derin öğrenme ile tahminini yapan bir çalışmada ise LSTM modeli kullanılarak gelecekteki üretim miktarı tahmin edilmiştir. Çalışma sonucunda LSTM modeli ile büyük bir başarı elde edildiği görülmüştür. Epoch sayısı 100, batch size 70 alınarak verilerin eğitilmesi ile yapılan tahminlemelerin diğer epoch ve batch size değerlerine kıyasla çok daha başarılı olduğu görülmüştür (Çetin, 2022). Bir başka çalışmada da yine LSTM modeli kullanılarak ETH (Ethereum) fiyat tahminlemesi yapılmıştır. Kısa dönem tahminlemede ARIMA başarılı olurken uzun dönem tahminlemede LSTM daha başarılı sonuçlar vermiştir. ARIMA'nın kısa dönem başarı oranı %94,1 olurken uzun dönemde bu oran %78,84'e gerilemiştir (Wasee vd., 2022).

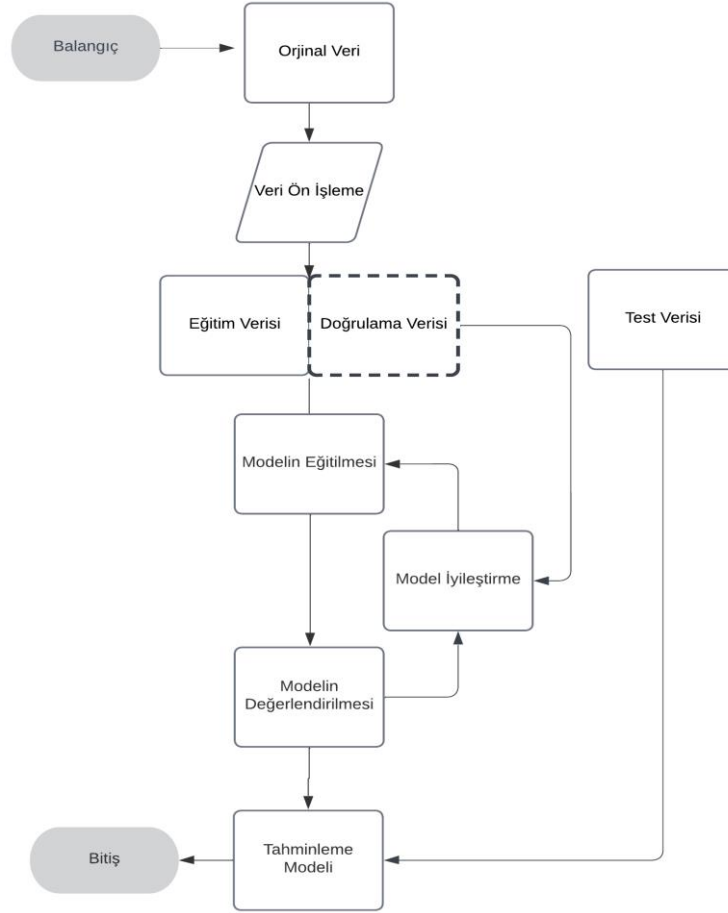
Bir diğer çalışmada LSTM ve ARIMA modelleri kullanılarak gümüş/ons paritesi üzerinde derin öğrenme ile finansal tahmin modellemesi geliştirilmiştir. MAE ve RMSE performans kriterleri kullanılarak ölçüm yapılmıştır. Sonuç olarak ARIMA algoritmasının LSTM algoritmasından daha iyi sonuçlar ürettiği görülmüştür (Üntez, 2022).

YÖNTEM

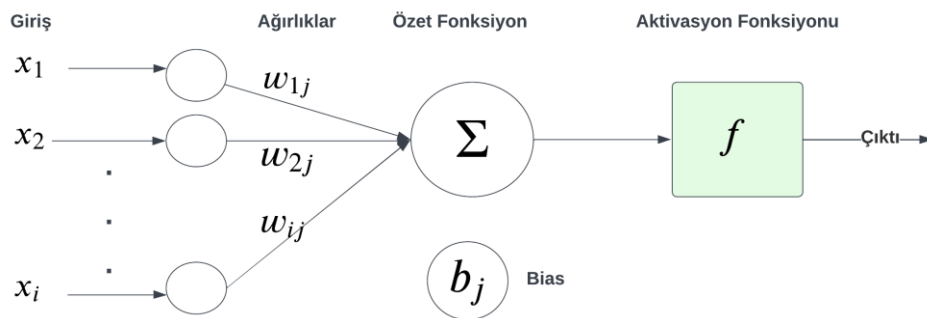
Makine öğrenmesi, bilgisayarların verileri analiz ederek insanlar gibi öğrenmelerini sağlayan bilim dalıdır. Makine öğrenmesi ve derin öğrenme için genel iş akış diyagramı (Güdelek, 2019) kaynağından yararlanarak Şekil 1'de gösterilmiştir. İş akışları, orijinal verinin okunması, veri ön işleme, eğitim ve test verisinin ayrılması, model oluşturma ve modelin eğitilmesi, modelin iyileştirilmesi ve son olarak modelin test verisi ile doğrulanması adımlarından oluşmaktadır.

YSA, insan beynini taklit eden ve öğrenme sürecine benzetilerek geliştirilmiş basit matematiksel modeller ile tanımlanan yapay zekâ yöntemlerindedir. Bu yapı, insan beynindeki sinir hücreleri gibi düşünülebilir. Sistem

girdileri ağırlıklandırılmış katsayılarla çarpılır ve ağırlıklandırılmış bir toplam oluşur. İşlem sonucunda sonuç bir eşik değer ile kıyaslanır. Sonuç eşik değerden büyükse sinir hücresi iletişime geçer, küçükse sinir hücresi bir iletim yapmaz. YSA'ların da çalışma modeli benzerdir. YSA'lardaki aktivasyon fonksiyonları sayesinde doğrusal olmayan ilişkilerin temsil edilmesi de sağlanabilmiştir (İnce, 2021). YSA mimarisi Şekil 2'de (Mohammedali, 2021) kaynağındaki gibi, yeniden çizilerek gösterilmiştir.



Şekil 1. Makine Öğrenmesi ve Derin Öğrenme Genel İş Akışı Diyagramı



Şekil 2. YSA Mimarisi

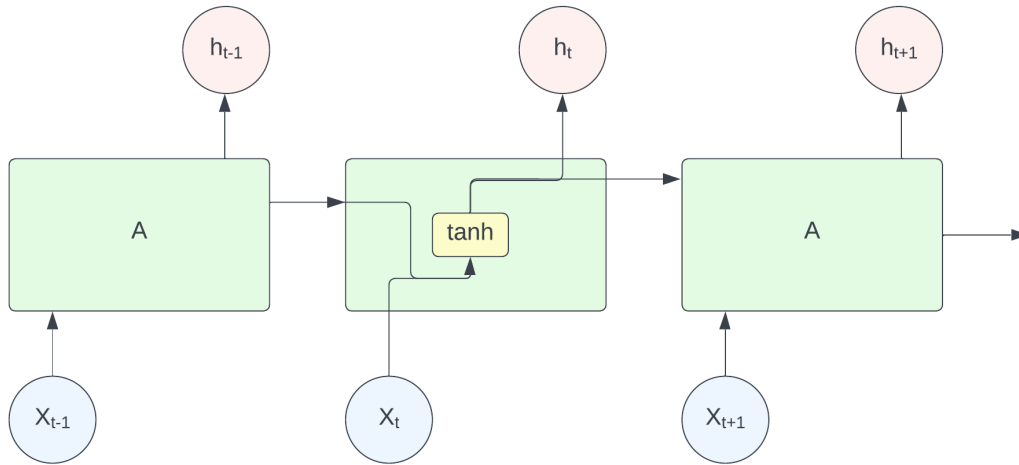
YSA, 5 farklı bileşenden oluşur. Bunlar sistem girdileri, düğüm ağırlıkları, toplama ve aktivasyon fonksiyonu ve çıktı düğümüdür. Birleştirme (özet) fonksiyonu, her X girdisini kendisine ait w ağırlığı ile çarpıp toplayarak çıkan değeri aktivasyon fonksiyonuna iletir. Aktivasyon fonksiyonu bu girdinin hücreye karşılık olarak vereceği çıktıyı hesaplar. Bu fonksiyon doğrusal değildir ve türevi kolay alınır. Özellikle türevi zor alınan aktivasyon fonksiyonlarının

kullanılmasının sistemi yavaşlattığı görülmüştür. Aktivasyon fonksiyonu olarak genellikle Sigmoid, tanjant hiperbolik fonksiyonu, eşik değer fonksiyonu ve sinüs fonksiyonu gibi fonksiyonlar tercih edilmektedir. Aktivasyon fonksiyonundan çıkan her değer o sinir hücresinin çıktısı olarak değerlendirilir (Günay, 2018). Sinir ağlarının katmanlarındaki nöronlar girdileri toplar, bir ağırlıklandırılmış bias değerini ekler ve çıktı üretir. Aktivasyon fonksiyonunun görevi, nöronun aktif veya pasif olup olmadığına karar vermektir. Aktivasyon fonksiyonu Step türünde ise sistem 1 çıktısı verdiğinde nöron aktif, 0 çıktısı verdiğinde pasif olmuş demektir (Güdelek, 2019).

Yinelemeli Sinir Ağları (Recurrent Neural Networks)

Bu ağlarda katmanlar sıralı şekilde çalışırlar. Bir sonraki katman bir önceki katmanın çıktılarını girdi olarak kullanır (Yaşar, 2019). RNN'nin geliştirilmesi 1980'li yıllarda David Rumelhart'ın çalışmalarına dayanmaktadır. Sıralı veriler üzerinde işlem yapma ve bunlardan içgörü elde etme yeteneği ile ön plana çıkan bir algoritmadır. Doğal dil işleme, video analizi, görüntü altyazıları gibi birçok alanda kullanılmaktadır (Üntez, 2022).

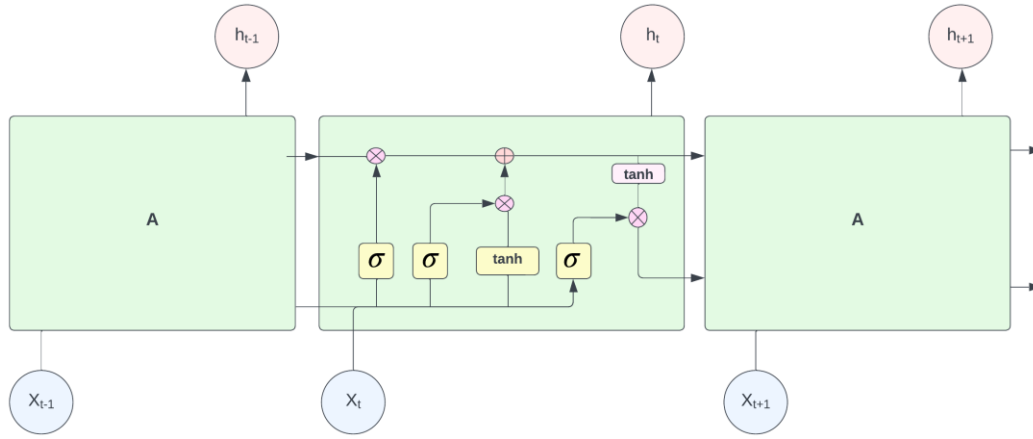
RNN algoritmasının çok popüler ve literatürde kaybolan gradyan problemi olarak geçen bir konusu vardır. Bu problem optimizasyon algoritmasının gradyan inişi ile kullanılması nedeniyle oluşur. Bağımlılıkların boşluklarının artmasına, gradyanların kaybolmasına ve ağırlıkların eğitilmesinin yavaşlamasına kadar giden bir süreci vardır. Hatta bu durumda ağırlıkların eğitilememesinden bile söz edilebilir (Mohammedali, 2021). RNN ağında tahminler katmanlardaki ağırlıklar ile ilişkilendirilerek oluşturulur. Tahminler ile gerçek değerler karşılaştırılır ve bir hata oranı hesaplaması yapılır. Hata payını düşürmek için bu işlem tekrarlanır. Bu nedenle yapıya tekrarlı sistem denmiştir. Hata payı 0'dan büyükse ağırlıklar değiştirilir ve model yeniden çalıştırılır. RNN temelde dört katmandan oluşur. Bunlar bir giriş, iki gizli katman ve bir çıkış katmanıdır (Şeker, 2017). RNN mimarisi (Güdelek, 2019) baz alınarak Şekil 3'te yeni bir gösterimle sunulmuştur.



Şekil 3. RNN Mimarisi

Uzun-Kısa Süreli Bellek (Long-Short Term Memory)

Sepp Hochreiter ve Jürgen Schmidhuber tarafından 1990'lı yıllarda geliştirilmiştir. Uzun vadeli bağımlılıkları öğrenebilme yeteneğine sahip özel bir RNN alt varyantlarından biridir. Bu nedenden dolayı zaman serisi verilerine dayanan tahminlerde oldukça performanslı çalışmaktadır (Çetin, 2022). Geleneksel RNN'lerden bu özelliği ile ayrılmaktadır. Uzun süreler boyunca ağıdaki bilginin hatırlanması işlemi LSTM modelinin en belirgin özelliğidir. LSTM bunu tek bir sinir ağı katmanı yerine birbirleri ile etkileşime giren dört katman kullanarak yerine getirir (Gündüz, 2019). Bir LSTM katmanı bir giriş kapısı, bir hücre durumu, bir unutma kapısı ve bir çıkış kapısından oluşmaktadır (Baş, 2019). LSTM mimarisi, (Hua vd., 2019)'daki modele benzer olarak Şekil 4'te gösterilmiştir.



Şekil 4. LSTM Genel Mimarisi

Unutma kapısının görevi verinin hücre katmanında ne kadar süre boyunca saklanacağını hesaplamaktır. Sigmoid fonksiyonu bu katmanda kullanılır ve 0-1 arasında normalize edilmiş bir sonuç üretir (Alpay, 2020). Giriş katmanının görevi ise verinin hücre durumundaki işlem süresini ayarlamaktır. Ayrıca bu katmanda yeni bilgiler de eklenebilir veya çıkarılabilir. Sigmoid fonksiyonu bu katmanda da kullanılmaktadır. Sonraki katmanda tanh fonksiyonu ile aday verilerin saklanması sağlanır ve bu katman ara katman olarak tanımlanır. Unutma kapısı, giriş katmanı ve tanh ara katmanı birleşerek hücre katmanını oluşturur. Hücre katmanı bir iletim hattı gibi çalışır ve anlamlı bilgileri hem hafızasında saklar hem de hat boyunca yer alan katmanlar arasında iletir. Bu özelliğinden dolayı LSTM algoritmasında kısa süreli bellek sorunu denilen sorun çözülmüş olur. Kapılardan gelen bilgiler hücre katmanının taşıyacağı formata dönüştürülür ve işlenir. Kapılar 0-1 arasında normalizasyon yapar ve bu işlem için sigmoid fonksiyonundan yararlanır. Aktivasyon işlemi sonunda değeri 0 olan bilgiler hafızadan silinirken, 1 olan bilgiler tutulmaya devam eder ve hat boyunca ilerler. Çıkış katmanı ise yine sigmoid fonksiyonu ile çalışır ve bilgileri filtreleyip nihai çıktıyı üretir (Kaya vd., 2020; Tanışman vd., 2021).

Otoregresif Entegre Hareketli Ortalama (ARIMA)

ARIMA algoritması 1970'li yıllarda Box ve Jenkins tarafından geliştirilmiştir. Başarı oranı özellikle kısa vadeler için ve durağan olmayan zaman serileri üzerinde yüksektir ve yaygın olarak kullanılır. ARIMA, AR (otoregresif), I (entegre) ve MA (hareketli ortalama) modüllerinden oluşur ve ARIMA(p,d,q) şeklinde formüle edilmiştir (Al-Asadi, 2019; Nusrat, 2022). AR modelinde önceki zaman adımları kullanılarak sonraki adımlar bu gözlemlerin bir türevi olarak modellenir. Tek değişkenli (univariate), trendi olmayan ve mevsimsellik özelliği göstermeyen zaman serileri için modellemeler yapmak için tercih edilir. Zaman serileri durağan değilse herhangi bir modelin gerçekleştirilmesi beklenmez ve tercih edilmez. Kısa bir tanımla artan veya azalan bir eğilimde olmayan ve mevsimsel değişkenlik göstermeyen zaman serileri durağan olarak nitelendirilir. Trend ve mevsimsellik, zaman serilerinin değerini etkilediği için veri kümesinde trend varsa, modeli oluşturmadan önce fark alarak ya da çıkararak trendi ortadan kaldırmak gerekir. AR modeli, önceki zaman dilimlerindeki gözlemleri kullanarak bir sonraki zaman dilimindeki değeri tahmin eder. Farklı periyotlardaki değerler arasında bir ilişki olabilir. Bu ilişkiye değişkenler arası korelasyon adı verilir ve pozitif ve negatif olmak üzere iki çeşittir. İlişki ters yönde değişiyorsa negatif korelasyona, aynı yönde değişiyorsa pozitif korelasyona işaretler. Korelasyonu hesaplamak için istatistik tabanlı metrikler kullanılır. AR (p) notasyonu aşağıdaki Denklem (1) ile ifade edilir.

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t \quad (1)$$

Denklem (1)'deki c sabit katsayıdır. Zaman serisi X_t ile ve zaman serisinin tamsayı indeksi de t ifadesi ile gösterilir. AR modelinin parametre değerleri φ_i ile gösterilir. P, modelin gecikme sayısının, ε_t ise ortalaması sıfır olan sabit varyanslı hata teriminin ifadesidir. MA modeline yuvarlanan veya hareketli ortalama denir. Veri analizinde kullanmak için varsa farklı alt kümeler diğer kümelerin ortalama hesabını yapar (Kaya vd., 2020).

$$X_t = \mu + \varepsilon_t \sum_{i=1}^q \theta_i \varepsilon_{t-i} \quad (2)$$

Denklem (2)'deki μ , modeldeki serinin ortalamasını gösterir. Hareketli ortalamanın değeri θ_i , ortalamanın sırası ise q ile ifade edilir. X_t zaman serisini belirtir. Hata terimleri de ε_t ve ε_{t-i} ile gösterilir (Kaya vd., 2020). ARMA, hem

önceki adımlardan bir sonraki adımı doğrusal olarak modeller hem de AR ile MA modellerini birleştirir. Bunun yanı sıra ön işleme adımını birleştirerek diziyi durağan hale getirir. Bu işleme entegrasyon (I) adı verilir ve aşağıdaki Denklem (3) gibi formüle edilir (Kaya vd., 2020).

$$X_t = \frac{\left(1 + \sum_{i=1}^p \theta_i L^i\right) \varepsilon_t}{\left(1 - \sum_{i=1}^p \theta_i L^i\right) (1-L)^d} \quad (3)$$

Denklem (3)'teki gecikme operatörü L ile ifade edilmiştir. Otoregresif modelin parametre değeri θ ifadesi, hareketli ortalama modelinin parametre değerleri ise θ_i ile gösterilmiştir. Benzer şekilde X_t zaman serisini belirtir. Hata terimleri de ε_t ile gösterilir (Kaya vd., 2020). ARIMA (p,d,q) modeli giriş parametreleri olarak üç bağımsız değişken alır. Buna göre p gecikme sayısını, d farklılaşma derecesini, q ise hareketli ortalama “window size (pencere boyutu)” değerini ifade etmektedir. p ve q parametrelerini belirlemek için birden fazla yöntem vardır. Ancak öncesinde zaman serisinin durağan hale getirilmesi gerekir. Bu durumda d=1 değerini alır. Bu yöntemlerden birisi otokorelasyon fonksiyonu ve kısmi korelasyon fonksiyonu grafiklerini yorumlamaktır. Grafikler incelenerek AR ve MA sayıları belirlenir. Ek olarak bu grafikler trend ve mevsimsellik hakkında da fikir verebilir. ACF bir serideki otomatik korelasyon değerlerini gösterir. Geçmişteki değerler ile günümüzdeki değerler arasında nasıl bir ilişki olduğunu belirlemesi yönünden önemlidir. ACF trend, mevsimsellik, gürültü gibi bileşenleri analiz eder ve bu nedenle tam otomatik korelasyon grafiği olarak adlandırılır. Durağan olmayan zaman serilerinde bu grafiğin zaman içerisinde düşüş eğiliminde olduğu görülür. PACF ise kısmi otomatik korelasyon grafiği olarak adlandırılır. İki nokta arasındaki zaman içerisinde değişen korelasyonu dikkate alır. AR modelinin optimal terim sayısını belirlemek için kullanılır. Terim sayısı aynı zamanda modelin sırasını belirleyen parametredir. PACF, AR modelini en iyi tanımlayan fonksiyonlardandır. SARIMA modeli ARIMA modeline mevsimsellik parametrelerinin eklenmiş türevidir. ARIMA ile benzer şekilde çalışır. Geçmişteki verileri inceler ve gelecekteki verileri doğrusal bir fonksiyon kullanarak modeller. ARIMA modelinin gerçekleştirdiği tüm işlemleri serinin mevsimsellik özelliğini de ekleyerek gerçekleştirir.

Prophet

Prophet mevsimselliği kullanarak günlük, haftalık, aylık ve yıllık bazda verileri tahmin etmeyi sağlayan otomatik bir makine öğrenimi paketidir. Meta (Facebook) şirketi tarafından geliştirilmiştir (Indulkar, 2021). Doğrusal olmayan eğilimlerin yıllık, haftalık ve günlük mevsimsellik dışında tatil etkileri ile uyumlu olduğu bir toplama modeline dayalı zaman serisi verilerini tahmin etme modelidir. Prophet, eksik verilere ve trenddeki değişimlere karşı dirençlidir ve tipik olarak aykırı değerleri kendi içinde iyi yönetir (Facebook, 2023). Prophet, güvenilir tahminler ürettiği, diğer tüm yaklaşımlardan daha iyi performans gösterdiği ve saniyeler içinde tahminleme yapabildiği için Meta'daki birçok uygulamada kullanılmaktadır. Denklem (4) ile ifade edilir. Denklem (4)'te $y(t)$ eklemeli regresif model, $g(t)$ trend faktörü, $h(t)$ tatil bileşeni, $s(t)$ sezonsallık bileşeni ve e_t ise hata terimi olarak tanımlanmıştır (Indulkar, 2021).

$$y(t) = g(t) + h(t) + s(t) + e_t \quad (4)$$

Bu model tatillerin etkisini değerlendirmemizi sağlar. Tatiller, bir zaman serisi üzerinde net bir etkiye sahip olabilecek düzensiz olaylardır. Örneğin, Amerika Birleşik Devletleri'ndeki Kara Cuma, Sevgililer Günü gibi etkinlikler, mağazalara katılımı veya bir e-ticaret web sitesindeki satışları önemli ölçüde artırabilir. Bu nedenle, bir zaman serisinde tatillerin etkisini modellemek için Prophet belirli bir ülke için bir tatil listesi tanımlamamıza izin verir. Daha sonra tatil etkileri, hepsinin bağımsız olduğu varsayılarak modele dâhil edilir. Bir veri noktası bir tatil tarihine denk gelirse, o zaman noktasındaki serideki değişikliği temsil etmek için bir parametre hesaplanır. Değişim ne kadar büyük olursa, tatil etkisi o kadar büyük olur (Peixeiro, 2022).

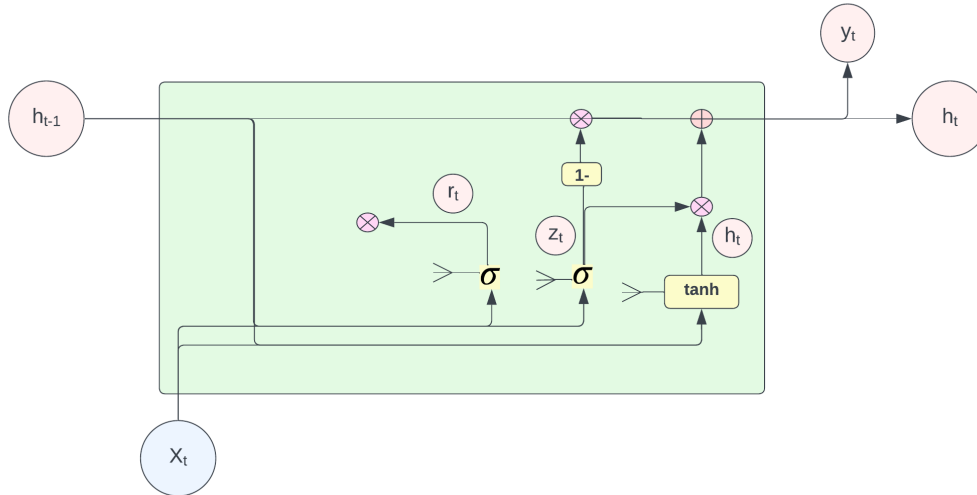
Ekstrem Gradyan Yükseltme (Extreme Gradient Boosting)

Friedman tarafından 2001 yılında geliştirilmiş sınıflandırma ve regresyon ağacına dayanan gradyan yükseltme temelli bir yöntemdir. Ancak XGBoost aşırı öğrenmeden kaçınmak için daha düzenli bir model biçimlendirmesi kullanır ve bu nedenle daha iyi performans gösterir. Performansı ve hızlı uygulanabilmesi nedeniyle son yılların en başarılı yöntemlerinden birisi olarak kabul edilir (Abar, 2020; Özdemir, 2021). Rastgele Orman algoritmasından farklı olarak burada torbalama (bagging) değil, yükseltme (boosting) kullanılmaktadır. Boosting yönteminde eğiticiler sıralı şekilde kullanılarak hatalardan ders alarak bir öğrenme ve ilerleme söz konusudur. XGBoosting algoritması, Gradient Boosting algoritmasından türemiştir ve daha hızlı, doğru ve paralel çalışabilen bir mimariye sahiptir (Kayış, 2021).

Geçitli Tekrarlayan Birim (Gated Recurrent Unit)

GRU, belirli durumlarda LSTM modeline göre avantajları olan bir RNN türüdür. GRU, daha az kapı ve bellek kullanır ve LSTM algoritmasına göre daha hızlıdır. Ancak LSTM daha uzun dizilere sahip veri kümeleri kullanıldığında daha doğru sonuçlar üretir. LSTM algoritmasının aksine üç kapı kullanır. Bunlar Güncelleme Geçidi, Sıfırlama Kapısı ve Mevcut Bellek Getirileridir. GRU, unutma kapısı olan bir LSTM gibidir. Ancak bir çıkış kapısı olmadığı için LSTM modeline göre daha az parametreye sahiptir. 2014 yılında Kyunghyun Cho tarafından tanıtılmıştır (Mohammedali, 2021).

GRU, tekrarlayan sinir ağlarının yaşadığı kaybolan gradyan problemini (ağ ağırlıklarının güncellemek için kullanılan değerler) ele alır. Derecelendirme, geri yayıldığında zamanla küçülürse öğrenmeyi etkilemek için çok küçük kalabilir ve böylece sinir ağını eğitilemez hale getirebilir. Bir sinir ağındaki katman öğrenemezse RNN'ler esasen daha uzun dizileri unutabilir. GRU modeli bu sorunu, güncelleme kapısı ve sıfırlama kapısı olmak üzere iki kapı kullanarak çözer. Bu kapılar, çıktıda hangi bilgilere izin verildiğine karar verir ve daha geriden gelen bilgileri tutmak için eğitilebilir. Bu, daha iyi tahminler yapmak için ilgili bilgileri bir olaylar zincirinden geçirmeye olanak tanır. Güncelleme Kapısı, geçmiş bilginin ne kadarının geleceğe aktarılması gerektiğini belirler. Bir LSTM tekrarlayan modelindeki Çıkış Kapısına benzer. Sıfırlama Kapısı, geçmiş bilgilerin ne kadarının unutulacağını belirler. Bir LSTM tekrarlayan birimindeki Giriş Kapısı ve Unut Kapısının kombinasyonuna benzer. Geçerli Bellek Kapısı ise Giriş Kapısının bir alt parçası olması ve girişe bir miktar doğrusal olmayan bilgi katmak, ayrıca girişi sıfır-ortalama yapmak için kullanılması nedeniyle Sıfırlama Kapısına dâhil edilmiştir. Bunu Sıfırlama geçidinin bir alt parçası yapmanın bir başka nedeni de önceki bilgilerin geleceğe aktarılmakta olan mevcut bilgiler üzerindeki etkisini azaltmaktır (Mohammedali, 2021). GRU mimarisi (Nusrat, 2022)'deki gibi çizilerek Şekil 5'te gösterilmiştir.



Şekil 5. GRU Genel Mimarisi

Zaman Serileri (Time Series)

Zaman serileri verileri, finanstan teknolojiye, sağlıktan hava durumuna kadar birbirinden farklı alanlara ait farklı ve düzenli zaman periyotlarına göre dağılım gösteren veri türleridir. Kaydedilen verilerin frekansı saat, gün, ay ya da yıl gibi değişen periyotlarda olabilir. Veriler zamana göre sıralı olduğu için aralarında zaman bağı ilişkiler oluşur. Özellikle birbirine komşu olan değerler arasında korelasyon oluşabilir. Tek değişken içeren ve çok değişken içeren seriler olmak üzere ikiye ayrılır. Tek değişkenli serilerde zamana göre değişen tek bir değişken ve bir bağımsız değişken varken çok değişkenin olduğu serilerde ise zamana bağlı yine bir değişken vardır, ancak bağımsız değişkenlerin sayısı 1'den fazladır (Güdelek, 2019). Zaman serilerinde durağanlığı tespit etmek için çeşitli yöntemler vardır. En yaygın kullanılanları ADF (Genişletilmiş Dickey Fuller Testi), Phillips-Perron, KPSS (Kwiatkowski-Phillips-Schmidt-Shin) Birim Kök Testi ve Korelogram grafiğidir (Özdemir, 2021).

ADF, sıfır ve alternatif hipotez içeren birim kök testidir. Sıfır hipotezi, birim kök içerir ve serinin durağan dışı olduğunu kabul eder. Alternatif hipotez ise serinin durağan olduğunu kabul eder ve birim kök içermez. Birim kök test örneklerinden biri Phillips-Perron yöntemidir. Bu yöntemin ADF'den farkı, t-test istatistiğine parametrik olmayan bir düzeltme yapmasıdır. Böylece otokorelasyonlar ve değişen varyans durumlarında elde ettiği sonuçlar daha düzgün olmaktadır. Serilerin durağanlık tespitinde sıklıkla kullanılan bir başka yöntem de KPSS testidir. Buradaki sıfır ve alternatif hipotez, ADF testinin tersidir. ADF testinde birim kök bulunmazsa ve KPSS testinde

bulunursa, seri fark-durağandır demektir ve yine de fark alınması gerekir. Bu testlerden biri bile birim kök bulursa seri durağandır denilemez. ACF ve PACF grafiğine bakarak da durağanlık tespiti yapılabilir. Durağan olmayan bir zaman serisinde ACF grafiğinin yavaşça düştüğü gözlemlenir. Bağımlılık aynı değişkenin önceki zaman dilimleri ile olan ilişkisini ifade eder. Gürültü (düzensiz bileşen) ise model tarafından tanımlanamayan veya açıklanamayan değişkenlere denir (Güdelek, 2019).

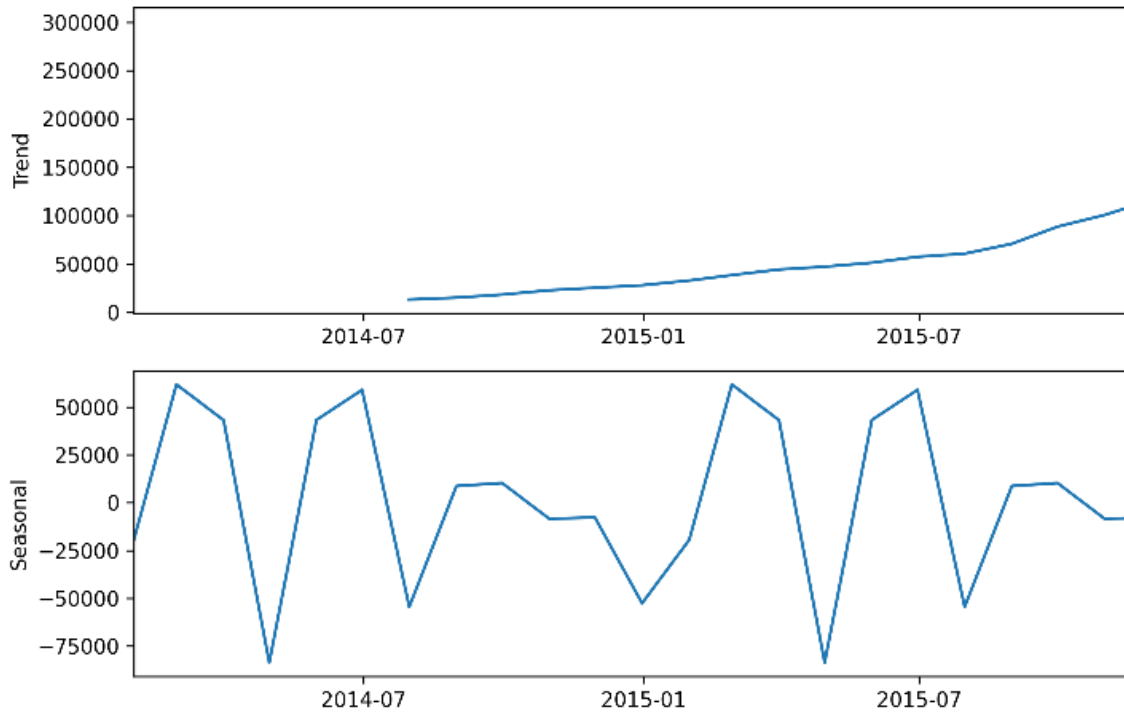
Zaman serileri tahmini ise geçmiş verilere dayanarak bir serinin gelecekteki olası değerlerini tahmin etmek için istatistiksel bir model oluşturma ve tahmin etme sürecidir. Eskiden zaman serileriyle ilgili problemlerin çözümünde lineer istatistiksel yöntemler kullanılırdı. Günümüzde ise verinin boyutu ve niteliği arttıkça lineer istatistiksel yöntemler yetersiz gelmeye başladı. Bu nedenle hem verideki özniteliklerin belirlenmesinde hem de verilerin karmaşıklıklarının analiz edilmesinde aradaki lineer olmayan ilişkileri öğrenme kabiliyetine sahip derin YSA'lar sıklıkla kullanılmaya başlanmıştır (Kara, 2019).

Zaman serileri tahminlemesi istatistik, ekonometri, matematiksel finans, deprem tahmini, hava tahmini, desen tanıma, sinyal işleme gibi birçok alanda kullanılabilir. Zaman serilerindeki unsurların toplamından oluşan "toplam modeli" ve çarpımlarından oluşan "çarpım modeli" olmak üzere iki temel model vardır. Bu modeller zaman serilerinde aşağıda gösterilen Denklem (5) ve Denklem (6) ile ifade edilirler. T_t trend, S_t sezonsallık, C_t periyodik, R_t artık parçalar demektir (Can, 2009).

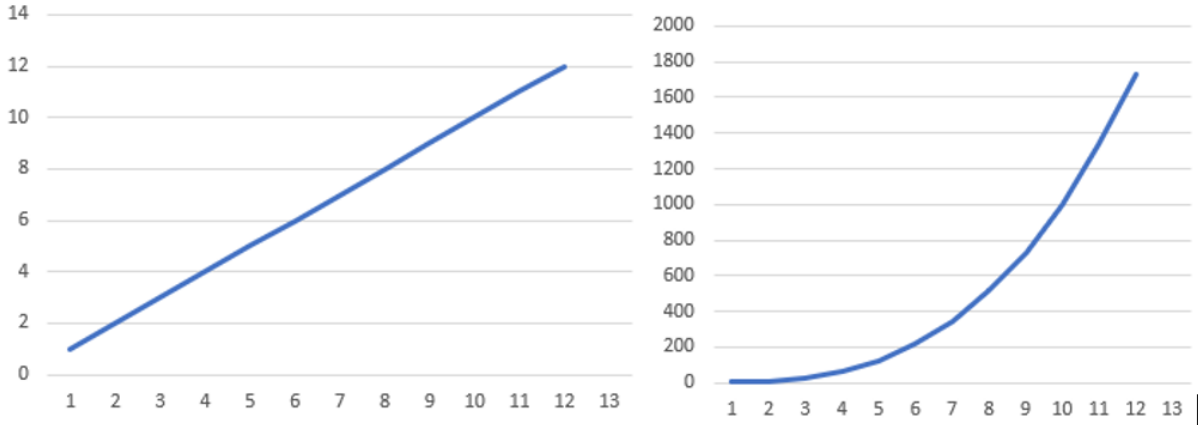
$$U_t = T_t + S_t + C_t + R_t \quad (5)$$

$$U_t = T_t \times S_t \times C_t \times R_t \quad (6)$$

Zaman serilerinde trend analizi oldukça önemlidir. Trend analizi uzun dönemlere ait verilerin analizinde kullanılır. Verinin artma, azalma ve durağanlık eğiliminde olup olmadığının tespitinin yapılması için kullanılan istatistiksel yöntemdir. Şekil 6'da trend analizi gösterilmiştir. Zaman serileri doğrusal veya doğrusal olmayan bir trend eğiliminde olabilirler (Oracle, 2023). Doğrusal trende sahip olan zaman serilerinde, eğilimi belirleyen hareket düz bir doğru şeklindedir. Doğrusal olmayan trende sahip zaman serilerinde ise eğilimi belirleyen hareket, matematiksel eğriler şeklindedir. Trend eğilimleri Şekil 7'deki gibidir (Can, 2009).



Şekil 6. Zaman Serileri Özellikleri



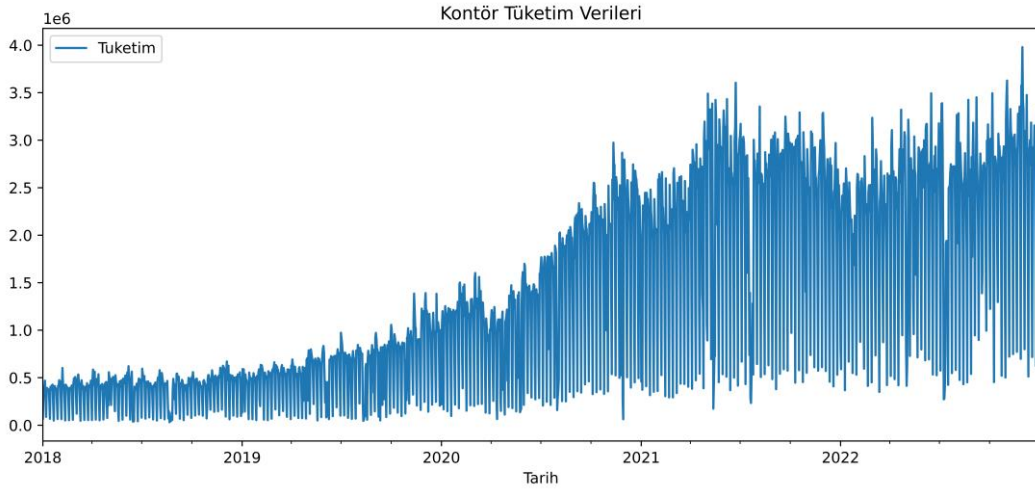
Şekil 7. Doğrusal ve Doğrusal Olmayan Trend

Bu çalışmada geliştirme ortamı (IDE) olarak Jupyter Notebook 6.5.2 ve Spyder 5.4.2 sürümü kullanılmıştır. Tüm modellerin geliştirmeleri, Python programlama dili ve Python 3.10.9 çekirdek (Kernel) sürümü kullanılarak yapılmıştır. ARIMA ve SARIMA modellerini kodlamak için Python paketlerinden “statsmodel.tsa”, Prophet için “prophet”, XGBoost için “xgboost”, RNN ve LSTM modelleri için ise “tensorflow” ve “keras” kütüphaneleri kullanılmıştır. Matematiksel işlemler, matris ve dizi işlemleri için “pandas” ve “numpy”, grafik ve görselleştirme için ise “matplotlib” ve “statsmodels.graphics” kütüphanelerinden yararlanılmıştır. Tüm model eğitme, test ve tahminleme işlemleri “11th Gen Intel Core i7-1165G7 @ 2.80GHz 2.80 GHz 32 GB Ram 64 Bit” bir bilgisayarda ve “Windows 10 Enterprise” işletim sistemi üzerinde gerçekleştirilmiştir.

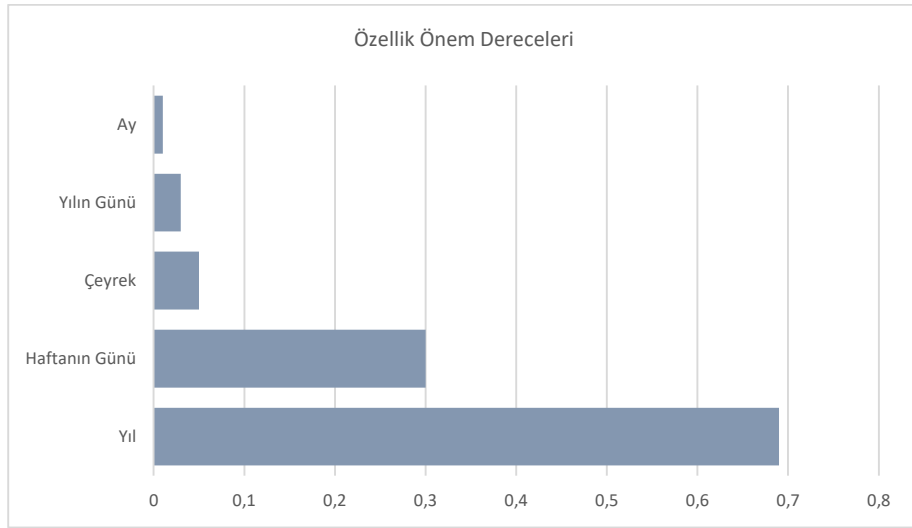
Veri Seti (Dataset)

Bu çalışmada bir teknoloji şirketinin müşterilerinin kullandıkları hizmetler için harcadıkları 9 yıllık günlük kontör tüketim verisi kullanılarak tahminleme yapılmıştır. Veri seti daha önce başka bir akademik çalışmada kullanılmamıştır. Veri seti 2014-2017 yıllarına ait tüketimler, 2018-2022 yıllarına ait tüketimler ve tüm yıllara ait (2014-2022) tüketimler olmak üzere 3 ayrı gruba ayrılmıştır. Buradaki ayırım şirketin dijital hizmetlerinin yıllara göre yaygınlaşması ve müşteri sayılarının artması kriterlerine göre belirlenmiştir. 2014 yılı hizmetlerin başlangıç yılı ve 2017 yılı ise müşteri sayısındaki artışın keskin arttığı yıl olması nedeniyle 2014-2017 yılları ayrı bir grup, 2018-2022 yılları ise ayrı bir grup olarak ele alınmıştır. Makine öğrenmesi ve derin öğrenme modelleri veri setleri üzerinde ayrı ayrı uygulanmıştır. Veriler tek değişkenli zaman serileri formatındadır ve zamana göre değişen kontör tüketim değerleri yer almaktadır. 2014-2017 yıllarına ait veri setinde 1461 adet, 2018-2022 yıllarına ait veri setinde 1825 ve 2014-2022 (tüm veri) yıllarına ait veri setinde 3286 adet satır bulunmaktadır.

Şekil 8’de görüleceği gibi, kontör tüketiminde 2020 yılına kadar artan bir trend, 2020 yılından itibaren ise keskin bir artış görülmektedir. Bu artış, bu yıllarda çıkan düzenlemelere bağlanabilir. Çıkan yasa ve yönetmelikler ile e-fatura kullanan şirket sayısının kapsamı genişlemiş ve buna bağlı olarak kontör tüketimleri de Şekil 8’deki gibi artmıştır. Şekil 9’da verideki özelliklerin önem dereceleri gösterilmiştir. Buna göre yıl ve haftanın günü özellikleri sonuca en fazla etki eden özellikler olarak öne çıkmıştır. Bunun nedeni ise şirketin hizmetlerinin sayısının, yaygınlığının ve müşteri sayısının yıllara göre artması olarak gösterilebilir.

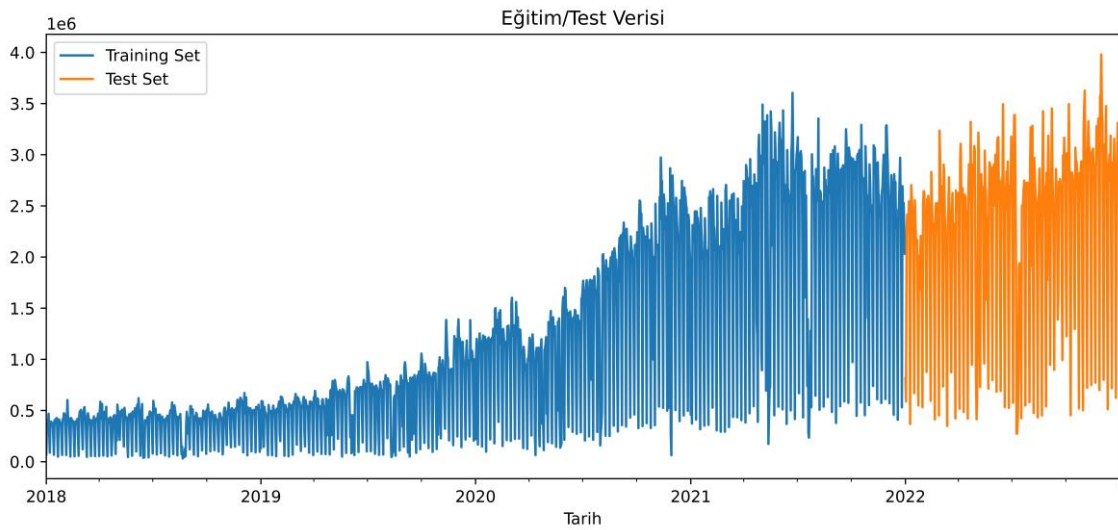


Şekil 8. 2014-2022 Yıllarındaki Müşteri Kontör Tüketim Dağılımı



Şekil 9. Sezonallık Özelliklerinin Önem Derecesi

2014-2017 yıllarına ait verinin 3 yılı eğitim verisi 1 yılı test, 2018-2022 yıllarına ait 5 yıllık verinin 4 yılı eğitim veri seti, 1 yıllık veri de test veri seti olarak kullanılmıştır. 2014-2022 yıllarına ait verinin ise 8 yıllık verisi eğitim, 1 yıllık veri ise test veri seti olarak kullanılmıştır. Şekil 10'da eğitim ve test verisi dağılımları gösterilmiştir.

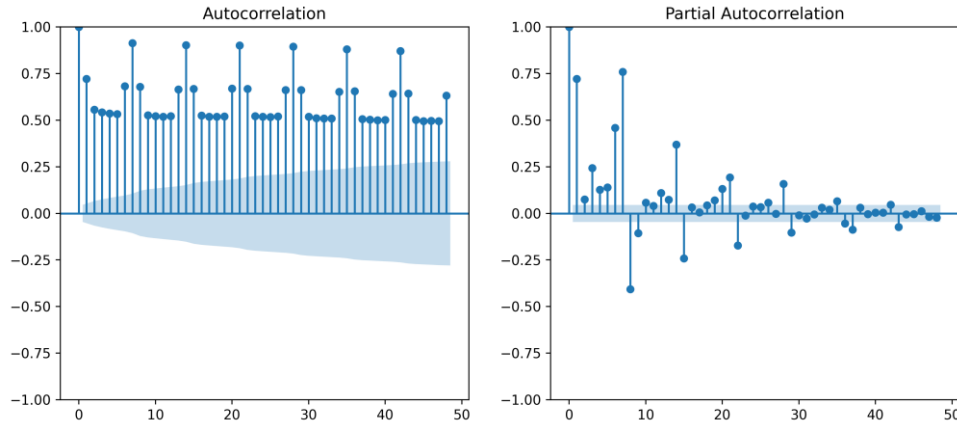


Şekil 10. Eğitim ve Test Verisi (2018-2022 yılları)

BULGULAR

Modeller sırasıyla 2014-2017, 2018-2022 ve 2014-2022 yıllarına ait üç veri seti üzerinde ayrı ayrı çalıştırılmıştır. Bu veri setlerinin nasıl ayrıldığı bir önceki bölümde detaylı anlatılmıştır. 2018-2022 yıllarına ait veri seti referans veri seti olarak alınmıştır ve sonuçlar 2014-2017 yılları ile tüm yılları içeren 2014-2022 veri setinin sonuçları ile ayrı ayrı karşılaştırılarak modellerin başarı oranları karşılaştırmalı olarak test edilmiştir. Bu karşılaştırma sonuçları bölüm sonunda gösterilmektedir. Veri setleri üzerinde önce makine öğrenmesi modelleri, sonra derin öğrenme modelleri ve son olarak da derin öğrenme modellerinden oluşturulan hibrit modeller sırasıyla çalıştırılmıştır.

Çalışmada MAE, MAPE, RMSE ve R^2 metrikleri kullanılarak performans ölçümü yapılmıştır. İlk olarak makine öğrenmesi modellerinden ARIMA ve SARIMA modelleri uygulanmıştır. Verinin durağanlığını tespit etmek için ACF ve PACF grafiklerinden yararlanılmıştır. Ayrıca ADF ve KPSS testleri uygulanmıştır. ACF grafiği zaman içerisinde bir düşüş eğilimi göstermektedir. Aynı şekilde PACF grafiği t-1 ve t-2 farkları için keskin bir düşüş göstermiştir. Bu iki veriye göre seri durağan olmadığı görülmüştür. Şekil 11’de ACF ve PACF grafikleri görülmektedir.

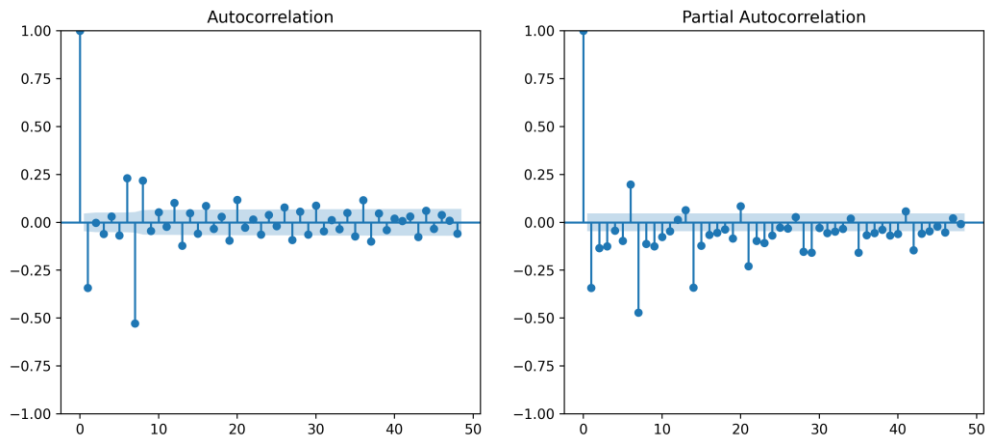


Şekil 11. ACF ve PACF Grafikleri

ADF ve KPSS testlerinin sonucuna göre de (Tablo 1) seri durağan değildir. Durağan hale getirmek için fark alma işlemi yapılmıştır. Serinin durağan olması için p-value değerinin 0,05’ten büyük olması gerekmektedir. Birinci dereceden fark alınmıştır. Fark alma işlemi sonrası ACF ve PACF grafikleri Şekil 12’de gösterilmiştir. ADF birim kök bulamaz ancak KPSS testi bulursa seri fark durağandır ve yine de fark alma işlemi uygulanarak serinin durağan hale getirilmesi gerekir.

Tablo 1. ADF ve KPSS Test Sonuçları

ADF Statistics	-0,9274541494854743
p-value	0,7787969935683057
KPSS Statistic	6,803335
p-value	0,010000



Şekil 12. Fark Alma Sonrası ACF ve PACF Grafikleri

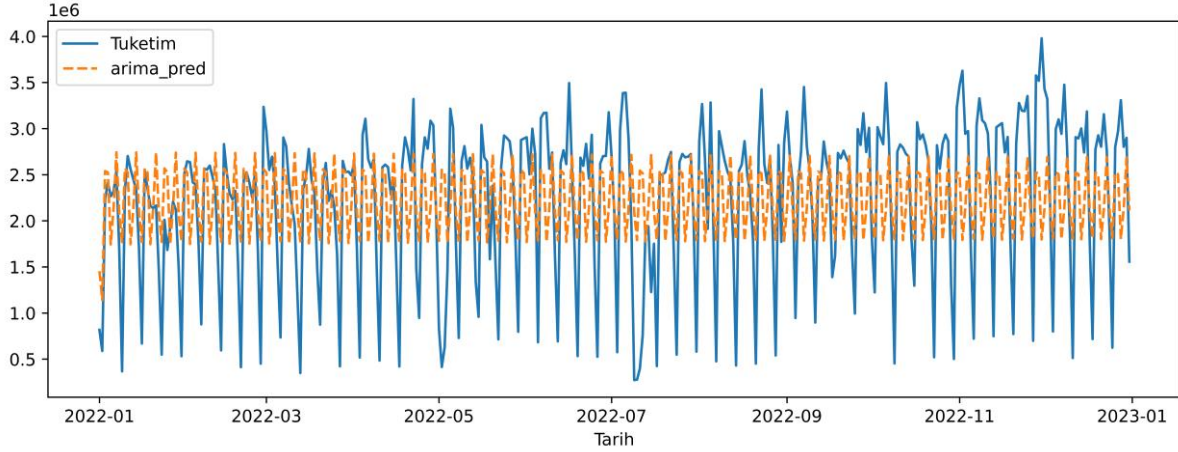
ACF ve PACF grafikleri aracılığıyla serinin durağan hale gelmesinden emin olmak için ADF ve KPSS testleri tekrar yapılmıştır. Ek ADF ve KPSS testi sonuçları ise Tablo 2’deki gibi oluşmuştur. P-value değerinden de anlaşılacağı üzere seri durağan hale getirilmiştir.

Tablo 2. Fark Alma İşlemi Sonrası ADF ve KPSS Testi Sonuçları

ADF Statistics	-
p-value	14,517242505878471
KPSS Statistic	5,598963729919661
p-value	0,036513
	0,100000

ARIMA(p,d,q) modelini uygulamak için önce p,d,q değerlerini hesaplamak gerekmektedir. Bu değerleri bulmak için iki yöntem bulunmaktadır. İlk yöntem ACF ve PACF grafiklerini yorumlamaktır. Grafik yorumlaması nedeniyle maliyetli ve hataya açık bir yöntemdir. İkinci yöntem ise Auto ARIMA dediğimiz “Grid Search” yöntemidir. Bu çalışmada Auto ARIMA yöntemi kullanılarak p,d,q değerleri hesaplanmıştır. Bu yöntemle göre en iyi model olarak ARIMA (3,1,5) önerilmiştir.

Veri sezonsallık içerdiği için ARIMA modeli iyi bir performans göstermemiştir. Performans metrikleri de bunu doğrulamaktadır. Şekil 13’te ARIMA modelinin tahminlemesi, Tablo 3’te ise model performans değerleri yer almaktadır.

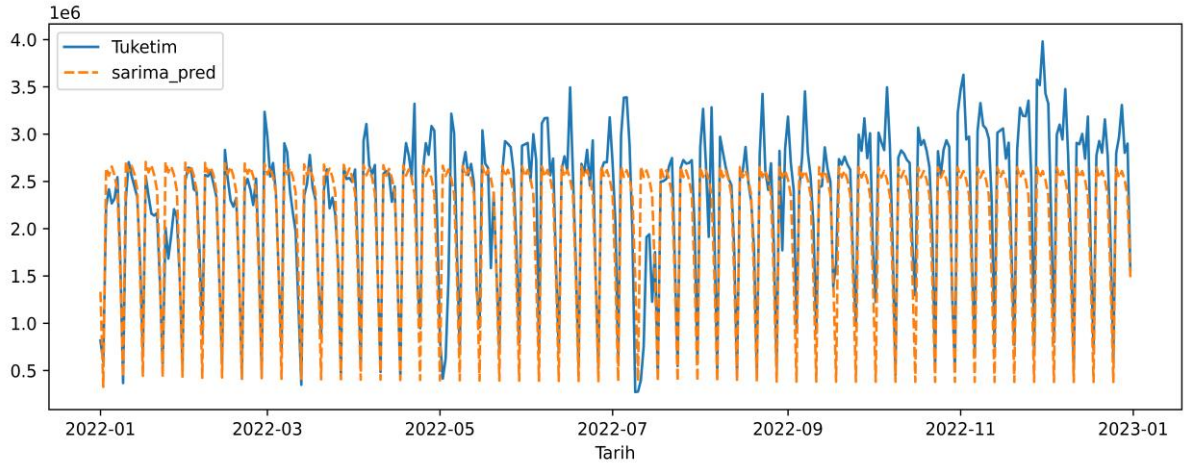


Şekil 13. ARIMA Modeli Tahminlemesi

Tablo 3. ARIMA Modeli Performansı

	ARIMA
MAE	597292,383534
MAPE	0,513970
RMSE	749371,846102
R ²	0,218284

ARIMA modelini iyileştirmek için modele mevsimsellik parametreleri eklenerek SARIMA modeli oluşturulmuştur. Auto ARIMA kullanılarak SARIMA (5,0,3) (0,1,1,7) modeli önerilmiştir. Veri sezonsallık içerdiği için, yani kontör tüketimi verisi sezonsal olarak değişkenlik gösterdiği için SARIMA modeli bu veri üzerinde ARIMA modeline göre çok daha iyi performans göstermiştir. SARIMA sezonsallık değişimlerini de dikkate alan bir modeldir. MAPE değeri ARIMA modelinde 0,51 iken SARIMA modelinde 0,21 olarak ölçülmüştür. Şekil 14’te SARIMA modelinin ARIMA modeline oranla oldukça iyi bir tahminleme yaptığı görülmektedir.



Şekil 14. SARIMA Modeli Tahminlemesi

SARIMA modeli gerçek tüketim değerlerine oldukça yakın tahminleme yapmıştır. Tablo 4’te de model performans ölçümleri yer almaktadır.

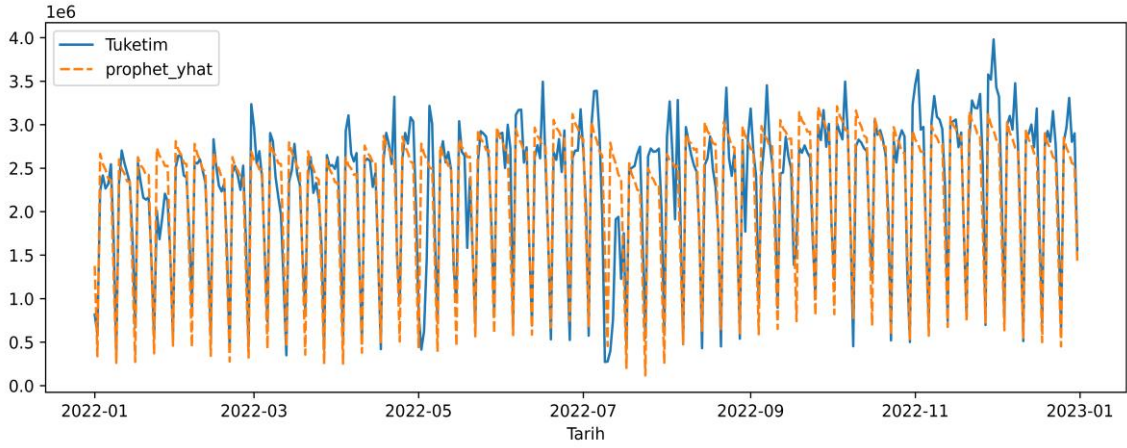
Tablo 4. SARIMA Modeli Performansı

	SARIMA
MAE	338533,575589
MAPE	0,217062
RMSE	470696,183788
R^2	0,691585

Prophet, Meta tarafından geliştirilen ve trend, mevsimsellik gibi parametreleri otomatik olarak içinde yöneten bir algoritmadır. İnsan davranışlarından kaynaklanan mevsimsel etkiler, haftalık, aylık ve yıllık döngüler, resmî tatillerdeki düşüşler ve zirveler, yeni ürünler ve Kara Cuma gibi pazar olaylarından dolayı trenddeki değişimler, aykırı değerler gibi birçok parametreyi başarılı yöneten bir algoritmadır. Model çalışma parametreleri Tablo 5’teki gibi belirlenmiştir. Şekil 15’te görüleceği üzere Prophet modeli, ARIMA ve SARIMA modellerine göre daha başarılı tahminleme yapmıştır. Prophet’in özellikle en tepe noktalarını SARIMA’ya göre daha başarılı tahmin ettiği görülmüştür. Tablo 6’da ise Prophet’in performans ölçüm değerleri yer almaktadır.

Tablo 5. Prophet Model Parametreleri

Parametre	Değeri
growth	Linear
yearly_seasonality	True
weekly_seasonality	True
daily_seasonality	True
seasonality_mode	multiplicative
seasonality_prior_scale	20
holidays_prior_scale	20
changeoint_prior_scale	0,01



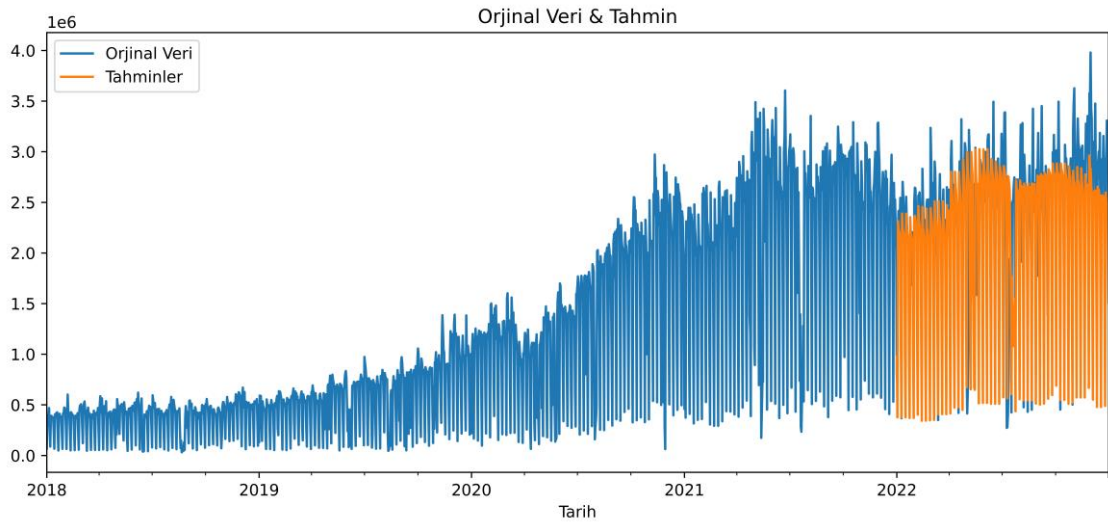
Şekil 15. Prophet Modeli Tahminlemesi

Tablo 6. Prophet Modeli Performansı

	PROPHET
MAE	283502,455315
MAPE	0,202677
RMSE	412533,996697
R^2	0,763095

Prophet, doğrusal olmayan trendlerin yıllık, haftalık ve günlük mevsimsellik ek olarak tatil etkileriyle uyumlu olduğu bir eklemeli modele dayanan zaman serisi verilerini tahmin etmeye yönelik bir yöntemdir. Güçlü mevsimsel etkilere ve birkaç mevsime ait geçmiş verilere sahip zaman serileriyle en iyi şekilde çalışır. Prophet, eksik verilere ve trenddeki değişimlere karşı dayanıklı olduğu ve genellikle aykırı değerleri iyi yönettiği için SARIMA modeline göre tahminleme başarı oranı daha yüksektir.

XGBoost, karar ağacı temelli ve Gradient Boosting bir makine öğrenmesi sistemidir. XGBoost modelinin çalışma parametreleri tahminleyici sayısı 1000, erken durdurma parametresi 50, öğrenme oranı ise 0,01 olarak belirlenmiştir. Bu parametreler karşılaştırmalı denemeler sonucu seçilmiştir. Tahminleme sonucu Şekil 16'daki gibi oluşmuştur. Tablo 7'de ise performans sonuçları yer almaktadır.



Şekil 16. XGBoost Modeli Tahminlemesi

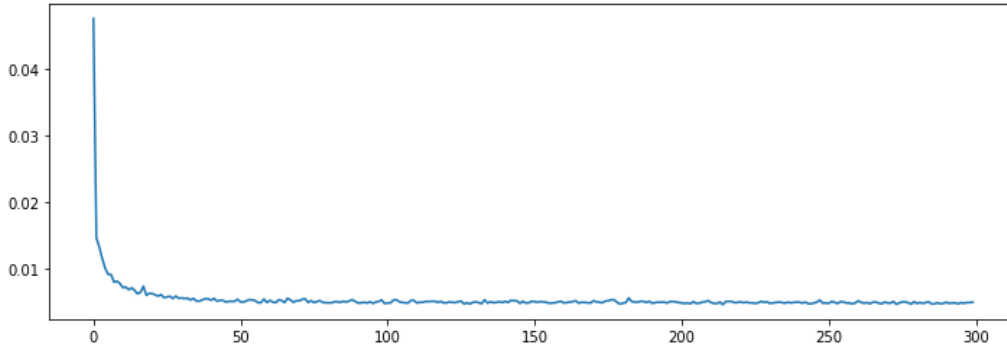
Tablo 7. XGBoost Modeli Performansı

	XGBoost
MAE	313162,62
MAPE	0,21

RMSE	465993,12
R^2	0,70

XGBoost modelinin sonuçları değerlendirildiğinde, Prophet ile tahminleme oranları benzer çıkmıştır ve SARIMA modeline göre tepe noktaları bulmada daha iyi sonuçlar vermiştir.

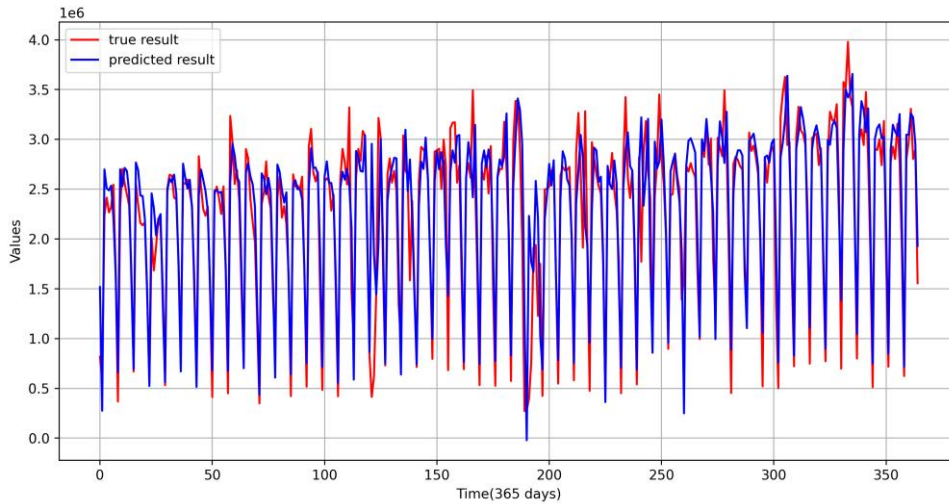
Bu bölümde ise derin öğrenme temelli algoritmaların sonuçlarına yer verilmiştir. Veri seti üzerinde ilk olarak RNN yöntemi uygulanmıştır. RNN modelinde 64 düğümden oluşan SimpleRNN, Dropout, Flatten ve Dense katmanı olmak üzere 4 katmanlı bir YSA kullanılmıştır. Aktivasyon fonksiyonu olarak “ReLU”, optimizasyon fonksiyonu olarak “adam” ve hata kaybı fonksiyonu için ise “ortalama kare hata (mse)” kullanılmıştır. Epoch sayısı 50, batch size 32 ve window size 90 olarak seçilmiştir. Epoch sayısı Şekil 17’deki kayıp fonksiyonuna göre belirlenmiştir. Tablo 8’de model performans sonuçları gösterilmiştir. Şekil 18’de ise tahminleme sonucu gösterilmiştir.



Şekil 17. RNN Kayıp Fonksiyonu

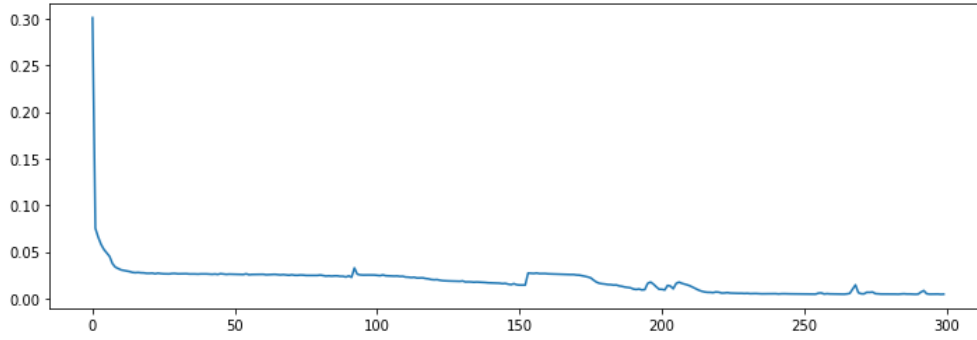
Tablo 8. RNN Modeli Performansı

	RNN
MAE	258768,70
MAPE	0,18
RMSE	368595,47
R^2	0,81

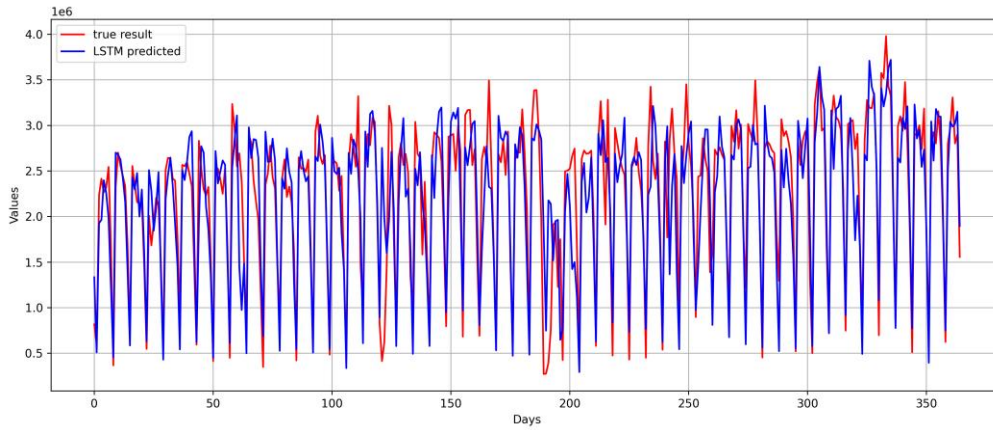


Şekil 18. RNN Modeli Tahminlemesi

LSTM modelinde ise 128 düğümden oluşan giriş ve çıkış katmanı olmak üzere iki katmanlı bir YSA modeli kullanılmıştır. Epoch sayısı 300, batch size 32 ve window size 90 olarak belirlenmiştir. RNN ile eşdeğer olarak aktivasyon fonksiyonu olarak “ReLU”, optimizasyon fonksiyonu olarak “adam” ve hata kaybı fonksiyonu için ise “ortalama kare hata (mse)” kullanılmıştır. Şekil 19’deki LSTM Kayıp Fonksiyonu grafiğine göre 300 epoch sayısında kayıp oranının stabil hale geldiği görülmektedir. Bu nedenle LSTM modelinde epoch sayısı 300 olarak alınmıştır. Şekil 20’de LSTM modelinin tahminleme sonucu gösterilmiştir.



Şekil 19. LSTM Kayıp Fonksiyonu



Şekil 20. LSTM Modeli Tahminlemesi

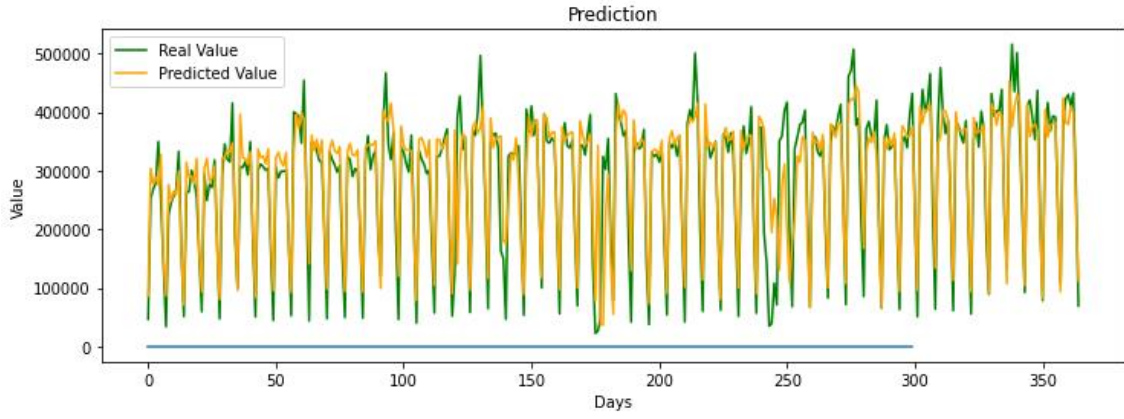
LSTM modelinde ayrıca farklı epoch, batch ve window size'lara göre model çalıştırılmış ve Tablo 9'daki sonuçlar elde edilmiştir. Bu tablo epoch 300, batch size 32 ve window size 90 parametreleri ile elde edilmiş en başarılı sonuçları vermiştir.

Tablo 9. LSTM Modeli Performansı

	LSTM
MAE	253316,13
MAPE	0,17
RMSE	383641,75
R^2	0,79

RNN ve LSTM sonuçları kıyaslandığında LSTM modelinin daha başarılı olduğu görülmektedir. Aralarındaki temel fark, LSTM'nin bilgiyi RNN'ye göre daha uzun süre hafızada tutabilmesinden kaynaklanmaktadır. Bu nedenle LSTM RNN'ye göre daha avantajlıdır, çünkü LSTM, RNN'ye kıyasla hafızadaki bilgileri uzun süre işleyebilir. Ayrıca LSTM uzun vadeli zamansal bağımlılıkları koruyabilir ve bilgiyi uzun süre hatırlayabilir. RNN, uzun bağımlılıklara sahip karmaşık görevler için uygun değil iken LSTM uzun vadeli bağımlılıkların modellenmesini gerektiren görevler için çok daha uygundur.

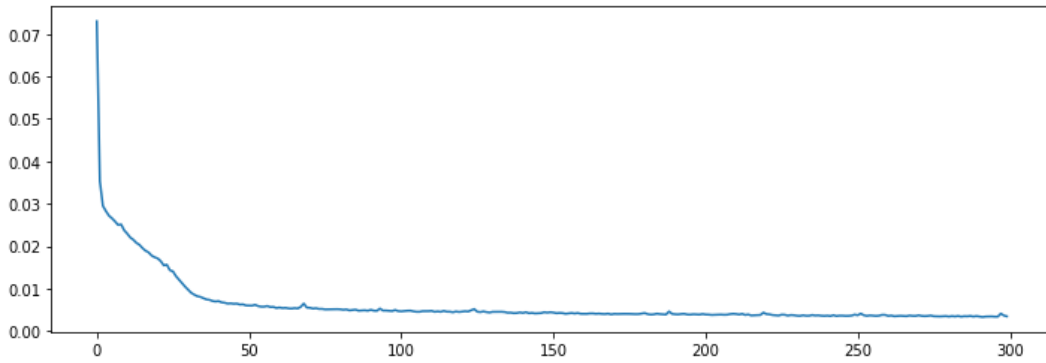
Şekil 21'de GRU modelinin tahminleme sonuçları gösterilmiştir. Tablo 10'da ise modelin performans sonuçları yer almaktadır. Şekil 22, GRU Kayıp Fonksiyonunu göstermektedir.



Şekil 21. GRU Modeli Tahminlemesi

Tablo 10. GRU Modeli Performansı

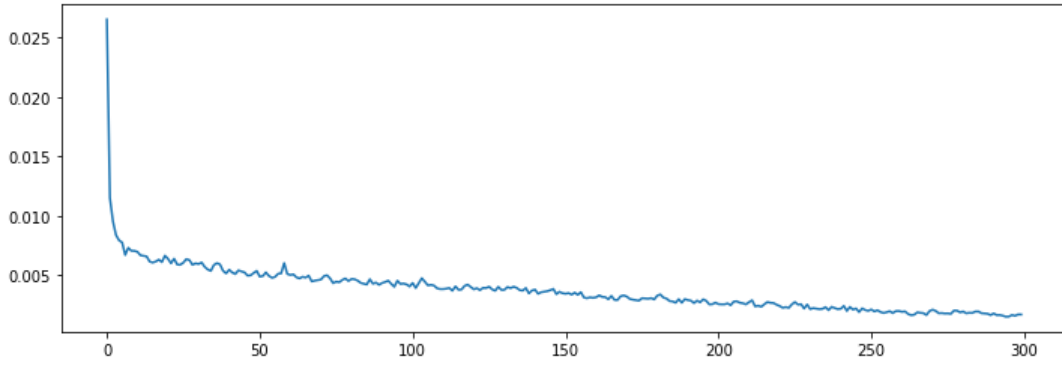
	GRU
MAE	252771,98
MAPE	0,18
RMSE	377491,71
R^2	0,80



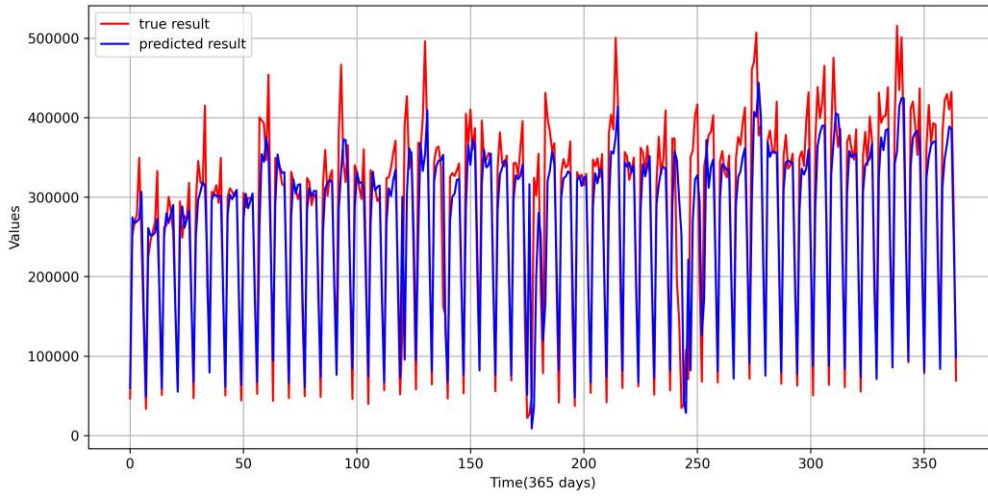
Şekil 22. GRU Kayıp Fonksiyonu

GRU modeli RNN ve LSTM'ye göre daha az kapıya sahiptir ve daha az parametresi vardır. Bu nedenle GRU performans açısından LSTM ve RNN'ye göre daha hızlı olmasına karşın uzun vade veri kümelerine uyarlanması noktasında daha kısıtlı bir algoritmadır. Tablo 10'daki ölçüm sonuçlarına göre RNN ve LSTM ile çok yakın sonuçlar verdiği gözlemlenmiştir.

Çalışmanın bu bölümünde ise derin öğrenme algoritmalarından oluşan hibrit modeller ile ilgili sonuçlara yer verilmiştir. RNN ve GRU algoritmaları ikisi bir arada hibrit model olarak kullanılmıştır. Hibrit model kavramı başarı oranlarını arttırmak için modellerin birlikte kullanılması prensibine dayanmaktadır. Modeller ayrı ayrı özelliklerde olabilirler ve ayrı öğrenme yeteneklerine sahiptirler. Bu nedenle birlikte kullanılarak YSA'da başarı oranları artırılabilir. Birden çok basit algoritma, birbirini tamamlamak ve artırmak için birlikte çalışır. Birlikte çalışarak tek başına çözemedikleri sorunları çözebilirler. Hibrit modelleri oluştururken bir modelin çıktısı diğer modele girdi olarak verilmektedir. Bu nedenle veri yapısı ve formatının buna uygun olarak belirlenmesi gerekmektedir. RNN ve GRU hibrit modeli tüm veri setlerine uygulanmıştır ve sonuçların daha başarılı olduğu görülmüştür. Aşağıdaki grafik ve tabloda RNN + GRU hibrit modelinin sonuçları yer almaktadır. Şekil 23'te görüldüğü gibi Kayıp Fonksiyonu oranı 300 epoch sayısında stabil hale gelmiştir. Bu nedenle RNN + GRU hibrit modelinin epoch sayısı 300 olarak alınmıştır. Şekil 24'te RNN+GRU hibrit modelinin tahminleme sonucu yer almaktadır. Tablo 11'de RNN + GRU modelinin performans çıktıları yer almaktadır.



Şekil 23. RNN + GRU Kayıp Fonksiyonu



Şekil 24. RNN + GRU Modeli Tahminlemesi

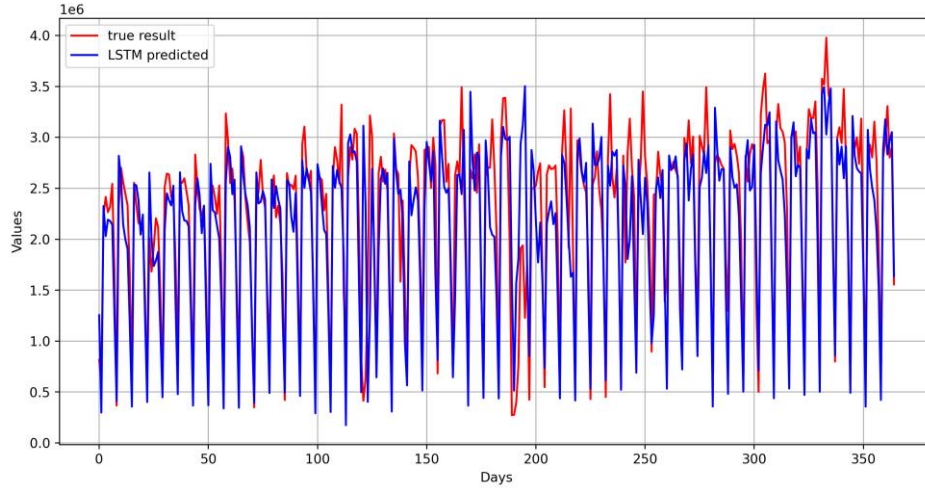
Tablo 11. RNN + GRU Modeli Performansı

	RNN + GRU
MAE	247293,97
MAPE	0,17
RMSE	379218,40
R^2	0,79

Benzer şekilde RNN + LSTM birlikte hibrit bir model oluşturulduğunda sonuçların hem makine öğrenmesi hem de derin öğrenme modellerine göre daha başarılı olduğu görülmüştür. Model, epoch sayısı 100, batch size 32 ve window size 60 parametreleri ile çalıştırılmıştır. Tablo 12 ve Şekil 25'te RNN + LSTM hibrit modelinin performans sonuçları yer almaktadır.

Tablo 12. RNN + LSTM Modeli Performansı

	RNN + LSTM
MAE	253481,53
MAPE	0,17
RMSE	374332,38
R^2	0,80



Şekil 25. RNN + LSTM Modeli Tahminlemesi

Tüm modellerin üç ayrı veri seti üzerindeki performans sonuçları Tablo 13’te karşılaştırmalı olarak belirtilmiştir.

Tablo 13. Modellerin Karşılaştırmalı Performans Tablosu

Modeller/Metrikler	MAE	MAPE	RMSE	R ²
2014-2017 Yıllarına Ait Veri Seti				
ARIMA	117562,46	0,73	131719,73	- 0,19
SARIMA	69752,57	0,29	86647,11	0,48
Prophet	98350,69	0,61	119062,67	0,02
XGBoost	95757,33	0,38	111930,82	0,14
RNN	36147,61	0,24	54819,15	0,79
LSTM	36952,69	0,24	57029,38	0,77
GRU	35892,84	0,26	55038,08	0,79
RNN + GRU	34070,98	0,20	56032,33	0,78
RNN + LSTM	33054,29	0,24	51318,22	0,81
2018-2022 Yıllarına Ait Veri Seti				
ARIMA	597292,38	0,51	749371,84	0,21
SARIMA	338533,57	0,21	470696,18	0,69
Prophet	283502,45	0,20	412533,99	0,76
XGBoost	313162,62	0,21	465993,12	0,70
RNN	258768,70	0,17	368595,47	0,81
LSTM	255900,54	0,18	368703,42	0,81
GRU	252771,98	0,18	377491,71	0,80
RNN + GRU	247293,97	0,17	379218,40	0,79
RNN + LSTM	253481,53	0,17	374332,38	0,80
2014-2022 Yıllarına Ait Veri Seti				
ARIMA	607739,82	0,50	754616,62	0,20
SARIMA	333428,83	0,21	466397,63	0,69
Prophet	770824,50	0,42	917314,04	- 0,17
XGBoost	313520,15	0,21	478258,73	0,68
RNN	287852,62	0,20	412975,82	0,76
LSTM	262566,93	0,18	378016,56	0,80
GRU	273876,99	0,21	390291,35	0,78
RNN + GRU	253907,16	0,17	380195,49	0,79
RNN + LSTM	265043,56	0,18	390127,01	0,78

SONUÇ

Bu çalışmada zaman serileri tahminlemesi konusunda 7 farklı model hazırlanmıştır. Ayrıca RNN/GRU/LSTM modelleri birleştirilerek hibrit modeller oluşturulmuştur. Literatürdeki benzer çalışmalar ile kıyaslandığında bu çalışmada konu oldukça geniş bir perspektiften incelenmiştir. Hibrit modeller dâhil 9 farklı modelin birbirleriyle olan performans karşılaştırmaları ve ayrıca literatürdeki diğer çalışmalar ile karşılaştırmalı analizi yapılmıştır. Bu yönüyle literatürdeki diğer çalışmalardan ayrılmaktadır. 2014-2017 yıllarına ait veri setinde makine öğrenmesi yöntemlerinden en başarılı sonucu SARIMA ve XGBoost modelleri göstermiştir. ARIMA modelinin performansı beklenildiği gibi olmuştur ve 0,73 MAPE değeri ile en başarısız model olmuştur. Derin öğrenme yöntemlerinden ise RNN + GRU ve RNN + LSTM hibrit modelleri en başarılı modeller olmuştur. Tekil modellerden ise LSTM ve RNN, GRU modeline göre daha başarılı olmuştur. Ancak LSTM modelinin uzun vadeli tahminlemeler için daha başarılı sonuçlar verdiği görülmüştür.

2018-2022 yılları arasındaki veri seti için ARIMA'da RMSE değeri 749371,84 ve MAPE 0,51 olarak ölçülürken SARIMA modelinde RMSE 470696,18 ve MAPE 0,21 olarak elde edilmiştir. Elimizdeki verinin mevsimsellik içermesinden dolayı SARIMA performansı ARIMA modeline göre iki katına yakın olmuştur. Bu veri setinde makine öğrenmesi yöntemlerinden en iyi performansı 0,20 MAPE değeri ile Prophet göstermiştir. XGBoost ile SARIMA modelleri benzer performans değerlerine sahip olmuştur. Derin öğrenme modellerinden ise RNN + GRU ve RNN + LSTM hibrit modelleri 0,17 MAPE değeri ile en başarılı modeller olurken bunları sırasıyla RNN, LSTM ve GRU modelleri takip etmiştir. 2014-2022 yılları arasındaki veri seti için benzer şekilde RNN + GRU ve RNN + LSTM hibrit modelleri en başarılı modeller olmuştur. Veri setlerine göre modellerin performansında kayda değer bir farklılık gözlemlenmemiş, tüm veri setleri için model performansları benzer şekilde ölçülmüştür. Literatürdeki çalışmalar ile kıyaslandığında mevcut çalışmadaki sonuçların daha başarılı olduğu görülmektedir.

Tüm veri setlerinde en iyi performansı RNN + GRU ve RNN + LSTM hibrit modelleri göstermiştir. Tüm veri setlerinde en başarısız model beklenildiği gibi ARIMA modeli olmuştur. Makine öğrenmesi modellerinden Prophet ile XGBoost benzer performans göstermiştir. Sezonel veriler üzerinde ise SARIMA modeli, ARIMA ve diğer makine öğrenmesi modellerine göre daha başarılı performans göstermiştir. Ayrıca Prophet hariç tüm modeller üç veri setinde de başarı oranları olarak aynı oranda değişim göstermiştir.

Bu çalışmanın sonuçları değerlendirildiğinde geçmiş tüketimler baz alınarak gelecekteki kontör tüketimleri tahmin edilebilir. Bu çalışmanın çıktıları kullanarak şirketlerin geleceği tahmin edebilmesi ve planlaması açısından bir projeksiyon ortaya konulabilir. Şirketler gelecek stratejik ve finansal planlarını ve kontör satışından gelecek olan tahmini gelir miktarlarını bu tahminlemeye göre yapabilirler. Yine böyle bir tahminleme, şirketlerin satış kampanyalarına ve pazarlama çalışmalarına yön verebilir. Tüm bunlara ek olarak eldeki verilerin yapay zekâ destekli sistemler tarafından işlenmesi ve değerli bir veriye dönüştürülmesi prestij açısından da şirketler için önemli bir uygulama olabilir.

Çalışmanın daha ileriye taşınması için farklı hibrit modeller oluşturularak modellerin performans iyileştirmesi yapılabilir. Ayrıca derin öğrenme modellerinden olan LSTM için tek katmanlı bir model tercih edilmiştir. Derin öğrenme yöntemleri için gizli katman sayısı artırılarak farklı modeller çalışılabilir ve performans karşılaştırması yapılabilir.

ÖNEMLİ NOT

Bu çalışma, İstanbul Üniversitesi-Cerrahpaşa Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans programında Hakan AKÇAY tarafından Doç. Dr. Derya YILTAŞ KAPLAN danışmanlığında hazırlanan "Zaman Serileri Tahminleme Algoritmalarının Karşılaştırmalı Uygulaması" isimli yüksek lisans tezinin içeriğinden oluşturulmuştur.

KAYNAKLAR

Abar, H. (2020). XGBoost ve Mars Yöntemleri İle Altın Fiyatlarının Kestirimi. *Ekev Akademi Dergisi*, 24(83), 427-445.

Al-Asadi, M. J. (2019). *Deep Learning For Time Series Forecasting of Currency Exchange Rates*. Yüksek Lisans Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü.

- Alpay, Ö. (2020). LSTM Mimarisi Kullanarak USDTRY Fiyat Tahmini. *Avrupa Bilim ve Teknoloji Dergisi* (Special Issue), 452-456.
- Amazon. (2023, 22 Ocak). *What is deep-learning?* Erişim adresi <https://aws.amazon.com/what-is/deep-learning/> adresinden alındı
- Baş, C. N. (2019). *Time Series Analysis With Deep Learning Approaches For Industry 4.0*. Thesis (MSc), Galatasaray University, Graduate School Of Science and Engineering.
- Can, M. (2009). *İşletmelerde Zaman Serileri Analizi İle Tahmin*. Doktora Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü.
- Çetin, Ö. (2022). *Güneş Enerji Santrallerinde Derin Öğrenme İle Aylık Elektrik Üretim Tahmini*. Yüksek Lisans Tezi, Mehmet Akif Ersoy Üniversitesi, Fen Bilimleri Enstitüsü.
- Dinçoğlu, P. (2022). *Perakende Sektöründe Veri Madenciliği Teknikleri İle Satış Tahmini*. Yüksek Lisans Tezi, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- Ejder, U. (2022). *DataMining Based On Regularized Convolutional Neural Network For Time Series Financial Prediction Algorithm*. Doktora Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü.
- Erbudak, A. E. (2022). *Veri Madenciliği ve Makine Öğrenimi İle Döviz Kuru Tahmini Uygulaması*. Yüksek Lisans Tezi, Altınbaş Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- Facebook. (2023, 23 Şubat). *Prophet*. Erişim adresi https://facebook.github.io/prophet/docs/quick_start.html adresinden alındı
- Güdelek, M. U. (2019). *Zaman Serisi Analiz ve Tahmini : Derin Öğrenme Yaklaşımı*. Bilgisayar Mühendisliği. Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü.
- Günay, M. (2018). *Makine Öğrenmesiyle Müşteri Kayıplarının Tahmini*. Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü.
- Gündüz, H. (2019). *Derin Öğrenme Yöntemleri İle Zaman Serisi Tahmini*. Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2019). Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine*, 114-119.
- Indulkar, Y. (2021). Time Series Analysis of Cryptocurrencies Using Deep Learning & Fbprophet. *Pune: 2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*.
- İnce, U. (2021). *Mobil Uygulamaların Gelecek Dönemde Kullanımının Zaman Serileri Kullanılarak Tahmin Edilmesi*. Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü.
- Kara, A. (2019). Global Solar Irradiance Time Series Prediction Using Long Short-Term Memory Network. *Gazi Üniversitesi Fen Bilimleri Dergisi*, 882-892.
- Kaya, U., Akba, F., Medeni, İ. T., & Medeni, T. D. (2020, Temmuz). Bitcoin Fiyat Değişimlerinin Makine Öğrenmesi, Zaman Serileri Analizi ve Derin Öğrenme Yöntemleriyle Değerlendirilmesi. *Bilişim Teknolojileri Dergisi*, 3(13), 341-355.
- Kayış, Ö. (2021). *Fiber İnternet Müşteri Şikayet Tahminlemesi*. Yüksek Lisans Tezi, Beykent Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- Kırbaş, İ., Sözen, A., Tuncel, D. A., & Kazancıoğlu, F. Ş. (2020). Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches. *Chaos, Solitons and Fractals*, 138.
- Lim, B., & Zohren, S. (2021). Time-series forecasting with deep-learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), 202-209.
- Microsoft. (2023, 10 Mart). *What-is-deep-learning?* Erişim adresi Microsoft Azure: <https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-is-deep-learning/> adresinden alındı
- Mohammedali, S. S. (2021). *Time Series Classification Using Deep Learning Networks*. Thesis (MSc), Karabük University, Institute of Graduate Programs.

- Nusrat, F. (2022). *Arima ve Derin Öğrenme Modelleri Kullanılarak Türkiyedeki Covid19 Vaka ve Vefat Sayılarının Tahmini*. Yüksek Lisans Tezi, Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü.
- Oracle. (2023, 15 Nisan). *What-is-deep-learning*. Erişim adresi <https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-deep-learning/> adresinden alındı
- Özdemir, M. C. (2021). *Arima ve XGBoost Modelleri İle Elektrik Talep Tahmini*. Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- Peixeiro, M. (2022). *Time Series Forecasting in Python by Marco Peixeiro*. New York: Manning Publications Co.
- Shastri, S., Singh, K., Kumar, S., Kour, P., & Mansotra, V. (2020). Time series forecasting of Covid-19 using deep learning models India-USA comparative case study. *Chaos, Solitons and Fractals*, 140(110227).
- Şeker, A. (2017). *Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme*. Fen Bilimleri Enstitüsü. İstanbul: Rapor, Yıldız Teknik Üniversitesi.
- Tanişman, S., Karcioğlu, A. A., Uğur, A., & Bulut, H. (2021, Aralık). LSTM Sinir Ağı ve ARIMA Zaman Serisi Modelleri Kullanılarak Bitcoin Fiyatının Tahminlenmesi ve Yöntemlerin Karşılaştırılması. *Avrupa Bilim ve Teknoloji Dergisi*, 32, 514-520.
- Üntez, A. (2022). *Gümüş Ons Paritesi Üzerine Derin Öğrenme İle Finansal Tahmin Modellemesi Geliştirilmesi*. Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- Wasee, M. S., Kundu, P. S., Mahzabeen, I., Tamim, T., & Alam, R. G. (2022). Time-Series Forecasting of Ethereum Price Using Long Short-Term Memory (LSTM) Networks. *Kuala Lumpur: Proc. of the 8th International Conference on Engineering and Emerging Technologies (ICEET)*.
- Yang, Y., Gao, W., & Guo, C. (2017). Aero-engine Lubricating Oil Metal Content Prediction Using Non-stationary Time Series ARIMA Model. *2017 10th International Symposium on Computational Intelligence and Design*.
- Yaşar, H. (2019). *Zaman Serileri Analizi ve Derin Öğrenme Modelleri Kullanarak Amerikan Doları/Türk Lirası Döviz Kuru İçin Hibrid Tahmin Modeli*. Yüksek Lisans Tezi, Doğu Üniversitesi, Fen Bilimleri Enstitüsü.