DOKUZ EYLÜL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENİMİ MODELLERİ ARACILIĞIYLA ZAMAN SERİSİ TAHMİNİ

Canberk BULUT

Eylül, 2023

İZMİR

MAKİNE ÖĞRENİMİ MODELLERİ ARACILIĞIYLA ZAMAN SERİSİ TAHMİNİ

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
İstatistik, Anabilim Dalı Veri Bilimi Programı

Canberk BULUT

Eylül, 2023
İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

CANBERK BULUT tarafından **PROF.DR. BURCU HÜDAVERDİ** yönetiminde hazırlanan **"MAKİNE ÖĞRENİMİ MODELLERİ ARACILIĞIYLA ZAMAN SERİSİ TAHMİNİ "** başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Prof.Dr. Burcu HÜDAVERDİ

Danışman

Dr.Öğr.Üyesi Özge ELMASTAŞ GÜLTEKİN	Prof.Dr. Tuğba YILDIZ
Jüri Üyesi	Jüri Üyesi

Prof. Dr. Okan FISTIKOĞLU
Müdür
Fen Bilimleri Enstitüsü

TEŞEKKÜR

Tez danışmanım Prof. Dr. Burcu HÜDAVERDİ'ye, bu tezin geliştirilme ve hazırlanma sürecindeki sabırlı ve mesafe farketmeksizin yaptığı rehberlik ve değerli önerileri ile bana yol gösterici olduğu için büyük bir minnettarlık duyduğumu ifade etmek istiyorum. Kendisi, değerli deneyimlerini paylaşma konusunda büyük bir isteklilik gösterdi. Prof.Dr. Burcu HÜDAVERDİ'nin yardımları ve rehberliği sayesinde, tez çalışmamdan çok büyük keyif aldım.

Canberk BULUT

MAKİNE ÖĞRENİMİ MODELLERİ ARACILIĞIYLA ZAMAN SERİSİ TAHMİNİ

ÖZ

Finansal zaman serilerinin tahmin edilmesi, ekonomide en karmaşık problemlerden biridir. Finansal piyasaların doğrusal olmayan faktörler ve belirsizliklerle dolu karmaşıklığı, tahminleri zorlaştırır ve genellikle düzensiz hareket etmelerine neden olur. Bu nedenle klasik tahmin yöntemleri yetersiz kalabilir. Makine öğrenimi yöntemleri genellikle daha iyi sonuçlar verir ancak karmaşıklık ve oynaklıkla dolu serileri tek başına tahmin etmekte yetersiz kalabilir. Bu nedenle hibrit yaklaşımlar önemlidir. Hisse senedi piyasası tahmini için geliştirilen hibrit modeller, zaman içinde makine öğrenimi algoritmalarının kullanımını arttırmıştır. Bu çalışmada, NASDAQ, DAX, NISEI ve BIST100 hisse senedi piyasalarından dört hisse senedi üzerinde hibrit modelleme yaklaşımı uygulanmıştır. Hisse senedi piyasaları değişken yapısı ve oynaklığı ile bilinir. Bu çalışmada, oynaklığı dikkate alan çeşitli hibrit modeller önerilmiştir. Hibrit modellemelerde, ARIMA modelleri öncelikle GARCH ile birleştirilerek zaman serileri modellemesinde kullanılmış ardından SVM ve LSTM gibi zeki modeller hata serilerinin doğrusal olmayan özelliklerini yakalamak için kullanılmıştır. Ayrıca önerilen hibrit modeller, mevcut yöntemlerle karşılaştırılarak performansları değerlendirilmiştir. Değerlendirmede RMSE, MSE, MAE ve MAPE gibi hata ölçütleri kullanılmıştır. Önerilen hibrit metodoloji, zaman serisi verisinin karmaşıklığını ve oynaklığını en iyi şekilde yakalayan farklı modelleri birleştirerek tahmin performansını büyük ölçüde arttırır.

Anahtar kelimeler: Zaman Seri Tahminleme, LSTM, SVM, ARIMA, GARCH, Hibrit yaklaşımlar, hisse senedi getirileri

TIME SERIES FORECASTING THROUGH MACHINE LEARNING MODELS

ABSTRACT

Estimating financial time series is one of the most complex problems in economics. The complexity of financial markets, full of nonlinear factors and uncertainties, makes forecasting difficult and often causes them to move erratically. Therefore classical estimation methods may be insufficient. Machine learning methods often give better results, but alone may fail to predict series full of complexity and volatility. Therefore, hybrid approaches are important. Hybrid models developed for stock market prediction have increased the use of machine learning algorithms over time. In this study, hybrid modeling approach was applied on four stocks from NASDAQ, DAX, NISEI and BIST100 stock markets. Stock markets are known for their volatile nature and volatility. In this study, several hybrid models that take into account volatility are proposed. In hybrid modeling, ARIMA models were first combined with GARCH and used in time series modeling, then intelligent models such as SVM and LSTM were used to capture nonlinear properties of error series. In addition the proposed hybrid models were compared with existing methods and their performances were evaluated. Error criteria such as RMSE, MSE, MAE and MAPE were used in the evaluation. The proposed hybrid methodology greatly improves forecast performance by combining different models that best capture the complexity and volatility of time series data.

Keywords: Time Series Forecasting, LSTM, SVM, ARIMA, GARCH, Hybrid approaches, stock market

İÇİNDEKİLER

Sayfa

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU i
TEŞEKKÜR iii
ÖZ iv
ABSTRACT v
İÇİNDEKİLER vi
ŞEKİLLER LİSTESİ viii
TABLOLAR LİSTESİ x

BÖLÜM BİR - GİRİŞ 1

BÖLÜM İKİ - ZAMAN SERİSİ KAVRAMI VE YÖNTEMLERİ 5

2.1 Zaman Serisi Yapısı 5
2.2 Zaman Serisinde Tahminleme 6
2.3 Otoregresif Hareketli Ortalama (ARIMA) 7
2.4 Otoregresif Koşullu Değişen Varyans (ARCH) 9
2.5 Genelleştirilmiş Otoregresif Koşullu Değişen Varyans (GARCH) 10
2.6 I-GARCH 11
2.7 E-GARCH 12
2.8 Destek Vektör Makinesi (SVM) 13
2.9 Yinelemeli Sinir Ağı (RNN) 16
2.10 Uzun Kısa Süreli Bellek (LSTM) 17
2.10.1 Unutma Kapısı 18
2.10.2 Girdi Kapısı 20
2.10.3 Hücre Durumu 21
2.10.4 Çıktı Kapısı 22

BÖLÜM ÜÇ - HİBRİT MODEL 24

BÖLÜM DÖRT - UYGULAMA 25

4.1 Uygulama Kapsamı 25
4.2 Karşılaştırma Metrikleri 26
4.3 Veri Setleri 27
4.3.1 NASDAQ 27
4.3.2 DAX 33
4.3.3 NISEI 40
4.3.4 BIST100 46

BÖLÜM BEŞ - DEĞERLENDİRME 54

KAYNAKLAR 55

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1 SVM Marjin Yapısıs 13

Şekil 2.2 RNN Yapısı 17

Şekil 2.3 Bir LSTM'in İç Yapısı 18

Şekil 2.4 Unutma Kapısı 19

Şekil 2.5 Girdi Kapısı 20

Şekil 2.6 Hücre Durumu 21

Şekil 2.7 Çıktı Kapısı 22

Şekil 4.1 NASDAQ Veri Dağılımı 27

Şekil 4.2 NASDAQ ACF Grafiği 28

Şekil 4.3 NASDAQ PACF Grafiği 28

Şekil 4.4 NASDAQ GARCH Test 29

Şekil 4.5 NASDAQ GARCH Test 2 30

Şekil 4.6 NASDAQ SVM Model Özellikleri 31

Şekil 4.7 ARIMA-SVM NASDAQ Logaritmik Dönüştürülmüş Veri Tahmini 32

Şekil 4.8 ARIMA-E-GARCH-SVM NASDAQ Logaritmik Dönüştürülmüş Veri Tahmini 33

Şekil 4.9 DAX Veri Dağılımı 34

Şekil 4.10 DAX ACF Grafiği 34

Şekil 4.11 DAX PACF Grafiği 35

Şekil 4.12 DAX GARCH Test 36

Şekil 4.13 DAX GARCH Test 2 37

Şekil 4.14 DAX SVM Model Özellikleri 38

Şekil 4.15 ARIMA-SVM DAX Logaritmik Dönüştürülmüş Veri Tahmini 39

Şekil 4.16 ARIMA-E-GARCH-SVM DAX Logaritmik Dönüştürülmüş Veri Tahmini 39

Şekil 4.17 NISEI Veri Dağılımı 40

Şekil 4.18 NISEI ACF Grafiği 41

Şekil 4.19 NISEI PACF Grafiği 41

Şekil 4.20 NISEI GARCH Test 42

Şekil 4.21 NISEI GARCH Test 2 43

Şekil 4.22 NISEI SVM Model Özellikleri 44

Şekil 4.23 ARIMA-SVM NISEI Logaritmik Dönüştürülmüş Veri Tahmini 45

Şekil 4.24 ARIMA-E-GARCH-SVM NISEI Logaritmik Dönüştürülmüş Veri Tahmini 46

Şekil 4.25 BIST100 Veri Dağılımı 47

Şekil 4.26 BIST100 ACF Grafiği 48

Şekil 4.27 BIST100 PACF Grafiği 48

Şekil 4.28 BIST100 GARCH Test 49

Şekil 4.29 BIST100 GARCH Test 2 50

Şekil 4.30 BIST100 SVM Model Özellikleri 51

Şekil 4.31 ARIMA-SVM BIST100 Logaritmik Dönüştürülmüş Veri Tahmini 52

Şekil 4.32 ARIMA-E-GARCH-SVM BIST100 Logaritmik Dönüştürülmüş Veri Tahmini 53

TABLOLAR LİSTESİ

Sayfa

Tablo 4.1 Performans Metrikleri 26

Tablo 4.2 NASDAQ Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri 32

Tablo 4.3 DAX Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri 38

Tablo 4.4 NISEI Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri 44

Tablo 4.5 BIST100 Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri 51

BÖLÜM BİR
GİRİŞ

İnsanların temel özelliklerinden birisi gelecekle ilgili planlar yapmaları ve bilinmeyen durumları tahmin etme arzularıdır. Bu bilinmeyeni tahmin etme arzusu aslında finans alanında da öngörü yapma anlamında çok önemli bir yere sahiptir. Bilinmeyeni tahmin etmek zordur çünkü belirsizlikler ve birçok ihtimal arasından doğru olanı seçmek gerçekten zor bir karardır. Özellikle bu durum finans alanındaysa daha da kritik bir durumdur. Bu nedenle tahmin hayatımızın farklı alanlarında karşımıza çıkmaktadır. Makine öğrenimi teknolojisi ile birlikte tahminleme metodolojisi hızla gelişmeye başlamıştır. Zaman serisi tahmini, özellikle finans, ekonomi, lojistik, tedarik zinciri vb. birçok alanda kullanılmaktadır. Finans bir ülkenin ekonomik olarak kalkınması için en dinamik ve değerli rollerden birisidir (Ake ve Dehuan, 2010). Ekonomideki öngörülemeyen değişiklikler sebebiyle finansal zaman serisini tahmin etmek genellikle zordur. Piyasa oynaklığı zaman serisi tahmininde ciddi sorunlara sebep olmuştur. Bu karmaşıklığı çözecek finansal zaman serisi yöntemleri geliştirilmektedir.

Zaman serisi tahmini konusunda en çok kullanılan yöntemlerden biri Otoregresif Entegre Hareketli Ortalama (ARIMA) yöntemidir. Bu yöntemde Otoregresif (AR) ve Hareketli Ortalama (MA) modelleri birleştirilmektedir. Bu yöntem, Box ve Jenkins yöntemi olarak da bilinmektedir (Box, Jenkins, Reinsel ve Ljung, 2015; Zhang, 2003). Ayrıca finansal piyasaların, karmaşık ve oynak sistemler olması (Chiang vd., 1996) sebebiyle bir finansal zaman serisinin durağan olmasından bahsedilememektedir (Yaser ve Atiya, 1996). Bu durum finansal zaman serisi tahminlemesinin zorluğunu kanıtlamaktadır. ARIMA ve diğer doğrusal modeller genellikle tek değişkenli veriler için kısa vadeli tahminde daha etkilidir, bu doğrusal modeller hisse senedi piyasasının oynak ve değişken yapısını tahminlemede çok başarılı değillerdir (Adebiyi, Adewumi ve Ayo, 2014). 1980'lerin başlarında ekonometristler borse endeksi tahmini için uygun model bulmaya çalışıyorlardı. Ancak verilerin oynaklığı fazla ve veriler karmaşıktı (Rao, 2008). Bu model oynaklığı için Engle (1992) Otoregresif Koşullu Değişken

Varyans (ARCH) yöntemini önermiştir ve Bollerslev (1986) Genelleştirilmiş Otoregresif Koşullu Değişken Varyans (GARCH) modelini tanıtmıştır. Syrris ve Shenai (2021) çalışmalarında ARIMA ve GARCH modellerini kullanarak altın fiyat endeksini tahminlemişlerdir. Vo ve Ślepaczuk (2022), çalışmalarında ARIMA modeli ile birlikte SGARCH ve EGARCH modellerini hibrit şekilde kullanarak S&P 500 borsa endeksini tahminlemişlerdir.

Makine öğrenimi yöntemleri zaman serisi modellemesine yeni yaklaşımlar getirmiştir. Özellikle derin öğrenme algoritmaları, doğrusal olmayan modelleri ya da daha karmaşık zaman serisi verilerini tanımlayabilmektedir. Yapay sinir ağları (ANN), bir tür makine öğrenimi algoritmasıdır. Daha az varsayım ve kısıtlamaya sahip olduğu için çeşitli alanlarda yaygın olarak uygulanmaktadır. Yapay sinir ağları modellerinin bu özelliği, karmaşık ve doğrusal olmayan zaman serilerinden önemli örüntüleri yakalama yetenekleridir (Sezer, Gudelek ve Ozbayoglu, 2020). Bu algoritmalar, birçok geleneksel istatistiksel yöntemle karşılaştırılmıştır. Yapay sinir ağları, verideki doğrusal olmayan durumlarla mücadele edebilir. Zhang vd. (1998) zaman serisi tahminlemesinde ANN uygulamaları konusunda çalışmışlardır. ANN genellikle, tek gizli katmanlı ve ileriye doğru beslemeli olarak çalışan zaman serisi modellemesi ve tahminlemesi yöntemidir. Tekrarlayan sinir ağları (RNN) ise zamansal örüntüleri öğrenmek için de kullanılabilir böylece sıralı verileri modelleyebilir ve zaman serisi analizlerinde de kullanılabilir. Son zamanlarda, özel bir RNN türü olan Uzun-Kısa Süreli Bellek (LSTM) yöntemi ilgi kazanmıştır. Bu algoritma genel olarak iyi bir doğruluk ve kesinliğe sahiptir. Maknickiene ve Maknickas (2012), ileri beslemeli yapay sinir ağlarının tahminleme performansını arttırmak için LSTM yöntemini kullanmışlardır. Chen vd. (2015) çalışmasında, LSTM algoritmasının Çin borsa piyasasını tahminlemede diğer tahminleme yöntemlerine göre daha iyi sonuç verdiğini göstermişlerdir. Adebiyi, Adewumi ve Ayo (2014) ARIMA ve Yapay Sinir Ağları algoritmalarının borsa endeksi tahminlemesindeki başarı oranlarını incelemişlerdir. Li ve Bastos (2020) çalışmalarında, hisse senedi piyasası tahminine yönelik derin öğrenme üzerine çalışmışlardır. Khashei ve Bijari (2010) çalışmalarında yapay sinir ağlarını kullanmışlardır. Yine Khashei ve Bijari (2011) çalışmalarında yapay sinir

ağları ile ARIMA modelini hibrit bir yaklaşımla zaman serisi tahminlemesi için kullanmışlardır.

En çok kullanılan diğer bir yöntem ise Vapnik (1995) tarafından geliştirilen destek vektör makinesi (SVM) yöntemidir. Destek Vektör Makineleri, örüntü tanıma problemleri için kullanılmaktadır (Scholkopf, Burges ve Vapnik, 1995). Son zamanlarda SVM'ler doğrusal olmayan regresyon tahminleme problemleri için kullanılmaktadır ve genellikle iyi sonuç vermektedir (Muller, Smola ve Scholkopf, 1997). Kim (2003), finansal zaman serilerinde SVM yöntemini kullanmıştır. Finansal uygulamalarda SVM, ampirik hata içeren risk fonksiyonu ve yapısal riski minimize etme ilkesine dayanan düzenli terim yöntemini içerdikten sonra kullanılmaya başlanmıştır. Fakat doğrusal olmayan modellerin birlikte yer aldığı zaman serisi verilerini tek başına modellemede yeterli olmayabilir. Cao ve Zhao (2022), çalışmalarında ARIMA ve GARCH ile birlikte model sonucunda çıkan hataları tekrar girdi olarak SVM modeline vererek hibrit bir yaklaşım uygulamışlardır.

Bu çalışmada, tahmin doğruluğunu ve kesinliğini arttırmak için zaman serisi verisinin doğrusal ve doğrusal olmayan kısımlarını birleştiren hibrit sistemler kullanılmıştır. Hibrit yaklaşım, model performansını arttırmak için artık modelleme kullanmaktadır (Zhang, 2003; Panigrahi, Behera, 2017). Ayrıca Zhang (2003), ARIMA ile ANN yöntemini zaman serisi tahminlemesi için birlikte kullanmıştır. Kim ve Won (2018), hisse senedi fiyatını tahminlemek için hibrit bir LSTM modeli sunmuştur. Domingos vd. (2019), ARIMA yöntemi ile çok katmanlı algılayıcılar (MLP) ve destek vektör regresyonu (SVR) yöntemlerini birlikte kullanarak zaman serisi tahminlemesini hibrit model ile yapmıştır. Hibrit modellemede, doğrusal otokorelasyon ve doğrusal olmayan bileşenden oluşan zaman serileri ele alınmıştır. Öncelikle ilk bileşeni modellemek için ARIMA ya da ARIMA-GARCH modeli kullanılmıştır. Buradan elde edilen artıklar zaman serisinin doğrusal olmayan kısmını temsil etmektedir. İkinci olarak LSTM ya da SVM modeli, doğrusal olmayan zaman serisi yapısını tespit etmek için kullanılmaktadır. Hibrit modellerin birçok avantajı bulunmaktadır çünkü zaman serisi verilerinin hem doğrusal hem de doğrusal olmayan kısımları çeşitli yöntemler kullanılarak ayrı ayrı modellenir ve daha sonra bu modeller

tahmin performansını iyileştirmek için bir araya getirilir. Tahminleme için hibrit yaklaşımların çeşitli kombinasyonları, en doğru sonuca ulaşmak için kullanılmaktadır.

Tez çalışması, belirtilen bölümlerden oluşmaktadır: Bölüm 2'de zaman serileri kavramlarından bahsedilmiş olup aynı zamanda zaman serisi analizinde kullanılan yöntemler incelenmiştir. Hibrit metodoloji de bu bölümde tanıtılmıştır. Hisse senedi endeksi analizi ve yöntemlerin karşılaştırılmasını içeren uygulama, Bölüm 3'te verilmiştir. Bölüm 4'te sonuçlar tartışılmıştır.

BÖLÜM İKİ
ZAMAN SERİSİ KAVRAMI VE YÖNTEMLERİ

2.1 Zaman Serisi Yapısı

Zaman serisi verileri kronolojik olarak zamana göre sıralanarak oluşturulur. Genellikle tutarlı olan bu veriler, belirli zaman aralıkları arasında alınır. Zaman serisi verileri yaygın olarak istatistik, ekonomi ve finans alanlarında kullanılır. Zaman serisi analizi, veri içerisindeki zaman bileşenini, istatistiksel anlam çıkartarak bir örüntüye oturtmaya çalışan analitik bir yöntemdir. Genellikle zaman serileri modelleri geçmiş verilere dayanarak gelecek değerleri tahmin etmek için kullanılır.

$\{y_t : t = 0, \pm1, \pm2, \pm3,\ldots\}$ rastgele değişken dizisine stokastik süreç denir ve gözlemlenen bir zaman serisi için bir model görevi görür. Bu ortak dağılımlardaki bilgilerin çoğu, beklenen değerler, varyanslar ve kovaryanslar cinsinden tanımlanır.

μ_t, sürecin t zamanındaki beklenen değeri olmak üzere Denklem 2.1'deki gibi,

$$\mu_t = E(Y_t) \text{ için } t = 0, \pm1, \pm2, \ldots \quad (2.1)$$

tanımlanır. Genellikle μ_t her t anında farklı olabilir. t ve s zamanlarındaki gözlemler arasında ilişkiyi açıklayan otokovaryans ve otokorelasyon sırasıyla Denklem 2.2 ve Denklem 2.3'te gösterilmiştir (Cryer ve Chan, 2008):

$$\text{Cov}(y_t, y_s) = E[(y_t - \mu_t)(y_s - \mu_s)] = E(y_t y_s) - \mu_t \mu_s, t, s = 0, \pm1, \pm2, \ldots \quad (2.2)$$

$$\text{Corr}(y_t, y_s) = \frac{\text{Cov}(y_t, y_s)}{\sqrt{Var(y_t)}\sqrt{Var(y_s)}} \quad (2.3)$$

Durağan zaman serileri, ortalaması, varyansı zaman içerisinde değişmeyen, kovaryans değeri sadece $t - s$ gecikmesine bağlı olan (zayıf durağanlık) zaman

serilerine denir. Zaman içerisinde momentleri değişen ve kovaryans değeri zamana bağlı olan serilere durağan olmayan zaman serileri denir. Durağan olmayan zaman serileri finans sektöründe oldukça yaygındır. Bu zaman serilerinde daha iyi tahmin yapmak için fark alma, logaritmik dönüşüm gibi teknikler uygulanarak zaman serileri durağan hale getirilmeye çalışılsa da finansal zaman serilerindeki oynaklık sebebiyle çoğu zaman başarılamaz (Ben Baccar , 2019).

Bir zaman serisi, trend, mevsimsellik, döngüsellik ve rassal hata bileşenlerini içerebilmektedir. Trend bileşeni, verilerde düzenli bir örüntü ortaya çıkarır. Belirli ve uzun bir süre boyunca artabilir ya da azalabilir. Tutarlı, uzun vadeli bir veri hareketi yönüdür. Verinin trende sahip olması için sürekli aynı yönde ilerlemesi gerekmemektedir bu sebeple zaman zaman artıp azalma olabilir. Ancak trendin altında yatan eğilim değişmemelidir (Cryer ve Chan, 2008).

Mevsimsellik bileşeni, verilerin belirli dönemlerde aynı eğilimi göstermeleridir. Yukarı ya da aşağı yönlü olarak aynı eğilimi sergilerler. Saatlik, günlük, haftalık, aylık, 3 aylık bazda bu eğilimler sergilenebilir (Cryer ve Chan, 2008). Mevsimsellik bileşeni, doğal ve yapay faktörler veya mevsimsel değişikliklerden etkilenmektedir (Jose, 2022).

2.2 Zaman Serisinde Tahminleme

Geçmişe ait veriler geleceğe yönelik tahmin doğruluğunu arttırmak için kullanılır. Son yıllarda makine öğrenimi ile geliştirilen modeller ARIMA gibi klasik modellere ciddi birer rakip olmuşlardır. Makine öğrenimi modelleri büyük bir ölçüde geçmiş veriyi kullandıkları için aslında dışarıdan anlaşılması biraz güç görünmektedir. Bu modeller yalnızca eğitim setindeki veriyi kullanarak geçmiş ve gelecek arasındaki stokastik ilişkiyi kurmaya çalışmaktadır. Makine öğrenimi modellerinin performansları çeşitli veri setleri üzerinde kullanılarak klasik modellerle rekabet edebileceği ilgili yarışmalarda denenmiştir (Ahmed vd., 2010; Palit ve Popovic, 2006; Zhang, Eddy ve Hu, 1998). Bu denemelere göre yapay sinir ağları modeli, Box ve Jenkins gibi klasik modelleri performans olarak geride bırakmıştır. Makine öğrenimi

modelleri temelde denetimli ve denetimsiz öğrenme olmak üzere ikiye ayrılır. Denetimli öğrenmede, hem girdi hem de çıktı verisi bulunmaktadır. Model girdi ve çıktı verisi arasındaki ilişkiyi bulmayı hedefler. Ancak denetimsiz öğrenmede çıktı verisi bulunmamaktadır. Algoritma girdi verisinin arkasında, gizli olan yapıyı geçmiş verilerden öğrenmeye çalışır. Denetimli öğrenme yöntemleri zaman serisi tahminlerinde kullanılabilir (Taghiyeh vd., 2020).

2.3 Otoregresif Hareketli Ortalama (ARIMA)

Çoklu regresyon modelinde, ilgilenilen değişken, tahmin edicilerin doğrusal bir kombinasyonu kullanılarak tahmin edilir. Bir otoregresyon modelinde, otoregresif modelde ilgilenilen değişkeni, değişkenin geçmiş değerlerinin doğrusal bir kombinasyonu kullanılarak tahmin edilir. Otoregresyon terimi, değişkenin kendisine karşı bir regresyon olduğunu gösterir.

Böylece, p dereceli bir otoregresif model Denklem 2.4'teki gibi yazılabilir:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad (2.4)$$

Burada ε_t beyaz gürültüdür. Beyaz gürültü, ortalamanın $\mu = 0$ ve varyansın $\mathrm{Var}(\varepsilon_t) = \sigma^2$ olduğu durağan bir süreç olarak varsayılır. Bu, çoklu bir regresyon gibidir ancak tahmin edici olarak y_t gecikmeli değerleri vardır. Buna bir AR(p) modeli, p mertebesinde otoregresif bir model denir. Otoregresif modeller, çok çeşitli farklı zaman serisi modellerini işlemede oldukça esnektir (Hyndman ve Athanasopoulos, 2018).

Bir hareketli ortalama süreci veya hareketli ortalama modeli, mevcut değerin mevcut ve geçmiş hata terimlerine doğrusal olarak bağlı olduğunu belirtir. Yine hata terimlerinin tıpkı beyaz gürültü gibi karşılıklı olarak bağımsız ve normal dağıldığı varsayılır.

Bir hareketli ortalama modeli, MA(q) olarak gösterilir ve burada q mertebedir. Model, mevcut değeri serinin ortalamasının (μ), mevcut hata teriminin (ε) ve geçmiş hata terimlerinin (ε) doğrusal bir kombinasyonu olarak ifade eder. Geçmiş hataların

mevcut değer üzerindeki etkisinin büyüklüğü, θ ile gösterilen bir katsayı kullanılarak ölçülür. Matematiksel olarak, genel bir hareketli ortalama süreci Denklem 2.5'teki gibi ifade edilir:

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \quad (2.5)$$

Böylece MA(q) sürecinin q mertebesinin, modele dahil edilmesi gereken geçmiş hata terimlerinin sayısını nasıl etkilediği görülebilir. Mertebe ne kadar büyükse geçmiş hata terimleri şimdiki değeri o kadar çok etkiler. Bu nedenle uygun modele uyması için hareketli ortalama sürecinin mertebesini belirlemek önemlidir yani ikinci dereceden bir hareketli ortalama süreci varsa tahmin için ikinci dereceden bir hareketli ortalama modeli kullanılmalıdır.

ARIMA, daha basit olan Otoregresif Hareketli Ortalama'nın bir genellemesidir ve entegrasyon kavramını ekler. Buna göre,

AR: Otoregresif, bir gözlem ile bazı gecikmeli gözlemler arasındaki bağımlı ilişkiyi kullanan bir model.
I: Entegre edilen zaman serisini durağan hale getirmek için ham gözlemlerin farkının alınması (örneğin önceki zaman adımındaki bir gözlemden mevcut gözlemin çıkarılması).
MA: Hareketli Ortalama. Bir gözlem ile gecikmeli gözlemlere uygulanan hareketli ortalama modelinden kalan hata arasındaki bağımlılığı kullanan bir model.

Bu bileşenlerin her biri, modelde açıkça bir parametre olarak belirtilir. ARIMA(p,d,q) için standart bir gösterim kullanılır; burada parametreler, kullanılan belirli ARIMA modelini hızlı bir şekilde belirtmek için tam sayı değerlerle değiştirilir (Brownlee, 2017).

$\varphi_i \neq 0$, $\theta_i \neq 0$ ve $\sigma_\varepsilon^2 > 0$ olmak şartıyla ARIMA modeli Denklem 2.6'daki şekilde gösterilebilir:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i} \quad (2.6)$$

Bu yöntem durağan olmayan zaman serileri ile uğraşırken de "entegre" olması sebebiyle başarılıdır. "Entegre" bileşeni aslında fark alma işlemini içerir ve bu işlem durağan olmayan zaman serisini durağanlaştırır (Siami-Namini vd., 2018). Bu gösterimde,

p: Gecikme sırası olarak da adlandırılan modele dahil edilen gecikme gözlemlerinin sayısı,
d: Ham gözlemlerin fark alma sayısı,
q: Hareketli ortalamanın sırası olarak da adlandırılan hareketli ortalama boyutunu ifade eder.

Belirlenen sayıda ve türde terim içeren bir doğrusal regresyon modeli kurulur ve durağan hale getirmek yani regresyon modelini olumsuz etkileyen trend ve mevsimsel yapıları ortadan kaldırmak için veriler bir dereceye kadar fark alınarak hazırlanır.

Modelin kullanılamayacak öğesini belirten bir parametre için 0 değeri kullanılabilir. Bu şekilde ARIMA modeli, bir ARMA modelinin ve hatta basit bir AR, I veya MA modelinin işlevini yerine getirecek şekilde yapılandırılabilir.

Bir zaman serisi için ARIMA modelinin benimsenmesi, gözlemleri oluşturan temel sürecin bir ARIMA süreci olduğunu varsayar. Bu varsayımın geçerliliğini doğrulamak için gözlem verileri ve modelden gelen hata terimlerini analiz etmek önemlidir. Bu analiz, modelin varsayımlarını gözlemler ve hata terimlerinde doğrulama gerekliliğini ön plana çıkarır (Brownlee, 2017).

2.4 Otoregresif Koşullu Değişen Varyans (ARCH)

En küçük kareler regresyonunun arkasındaki temel varsayım, eş varyans yani varyansın sabitliğidir. Bu koşul sağlanmadığı durumda, tahminler yine yanlı olacak ve

minimum varyans değerleri olmayacaktır. Bu durumda hesaplanan standart hata ve güven aralıkları çok dar olacak ve ARCH, modeldeki oynaklığı modelleyerek bu durumun üstesinden gelebilmektedir ve bu şekilde en küçük kareler modelinin eksikliklerini de ortadan kaldırmış olur (Dhamija ve Bhalla, 2010).

2.5 Genelleştirilmiş Otoregresif Koşullu Değişen Varyans (GARCH)

Genelleştirilmiş Otoregresif Koşullu Değişken Varyans (GARCH), ARCH modelinin otoregresif bileşenle birlikte hareketli ortalama bileşeni içeren bir uzantısıdır.

Spesifik olarak model, ortalama bir süreçten kalan gecikme hatalarıyla birlikte gecikme varyans terimlerini (örneğin başka bir sürecin beyaz gürültü artık hatalarının modellenmesi durumundaki gözlemler) içerir. Hareketli bir ortalama bileşeninin tanıtılması, modelin hem zamana bağlı varyansta hem de zamana bağlı varyanstaki koşullu değişimi modellemesine izin verir ve varyansta koşullu artışları ve azalmaları içerir. Bu nedenle model, gecikme varyansı terimlerinin sayısını açıklayan yeni bir parametre olan "p" 'yi belirler:

• p: GARCH modeline dahil edilecek gecikme varyanslarının sayısı.
• q: GARCH modeline dahil edilecek artık gecikme hatalarının sayısı.

Bir GARCH modeli için genel olarak kabul edilen notasyon, modeli GARCH(p, q) ile belirtmektir; örneğin GARCH(1, 1) birinci dereceden bir GARCH modelidir. Bir GARCH modeli, bir GARCH(0, q)'nun bir ARCH(q) modeline eşdeğer olduğu ARCH modellerini içerir.

$p = 0$ için süreç ARCH(q) sürecine indirgenir ve $p = q = 0$ için model basitçe beyaz gürültüdür. ARCH(q) sürecinde koşullu varyans yalnızca geçmiş örnek varyanslarının doğrusal bir fonksiyonu olarak belirtilirken, GARCH(p, q) işlemi gecikmeli koşullu

varyansların da girmesine izin verir. Bu, bir tür uyarlanabilir öğrenme mekanizmasına karşılık gelir (Bollerslev, 1986). ARCH' ta olduğu gibi GARCH, gelecekteki varyansı tahmin eder ve serinin varyanstaki değişiklik dışında durağan olmasını bekler, yani serinin bir trendi veya mevsimsel bileşeni yoktur (Brownlee, 2017).

GARCH sürecini takip eden y_t serisini ele alalım. y serisinin t süresi için koşullu dağılımı Denklem 2.7'deki gibi yazılır:

$$y_t|\Psi_{t-1} \sim N(0, h_t) \tag{2.7}$$

koşullu varyans h_t , Ψ_{t-1} , t-1 anında mevcut tüm bilgileri belirtir.
$p \geq 0, q > 0, \omega > 0, \alpha_i \geq 0$ ve $\gamma_j \geq 0$ olmak şartıyla model Denklem 2.8'de,

$$h_t = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{j=1}^{p} \gamma_j h_{t-j} \tag{2.8}$$

olarak tanımlanır.

ARCH parametrelerinden en az biri sıfırdan farklı olmalıdır (q > 0). GARCH regresyon modeli,
$y_t = x_t' \beta + \varepsilon_t$,$\varepsilon_t = \sqrt{h_t} e_t$ ve $e_t \sim (0, 1)$ iken Denklem 2.9'da,

$$h_t = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{j=1}^{p} \gamma_j h_{t-j} \tag{2.9}$$

olarak tanımlanır (SAS Help Center: GARCH Models, 2018).

2.6 I-GARCH

Denklem 2.10'da verilen koşulda, ortalama, varyans ve otokovaryans sonlu ve zaman içinde sabit olduğundan GARCH süreci zayıf durağandır. Ancak bu durum, otokorelasyon varlığında zayıf durağanlık için yeterli değildir:

$$\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \gamma_j < 1 \tag{2.10}$$

Örneğin bir AR(1) - GARCH(*p*,*q*) işlemi için durağanlık koşulu Denklem 2.11'deki gibidir:

$$\frac{1}{1-\varphi_1^2} \sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \gamma_j < 1 \tag{2.11}$$

GARCH süreci durağan olduğunda, ε_t'nin koşulsuz varyansı Denklem 2.12'deki gibi

$$V(\varepsilon_t) = \frac{\omega}{\left(1 - \sum_{i=1}^{q} \alpha_i \sum_{j=1}^{p} \gamma_j\right)} \tag{2.12}$$

elde edilmektedir.

Bazen varyansın çok adımlı tahminleri, model varyansa entegre edildiğinde koşulsuz varyansa yaklaşmaz; bu durum Denklem 2.13'te gösterilmiştir.

$$\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \gamma_j = 1 \tag{2.13}$$

I-GARCH modeli için koşulsuz varyans yoktur. Ancak I-GARCH modeli zayıf durağan olmamasına rağmen güçlü durağandır (SAS Help Center: GARCH Models, 2018).

2.7 E-GARCH

E-GARCH modeli Nelson (1991) tarafından önerilmiştir. Nelson ve Cao (1992), doğrusal GARCH modelindeki negatif olmama kısıtlamalarının çok kısıtlayıcı olduğunu savunur. GARCH modeli, α_i ve γ_j parametreleri üzerinde negatif olmayan kısıtlamalar uygularken E-GARCH modelinde bu parametreler üzerinde herhangi bir kısıtlama yoktur. E-GARCH modelinde koşullu varyans, h_t gecikmeli bozuklukların asimetrik bir fonksiyonudur (ε_{t-i}):

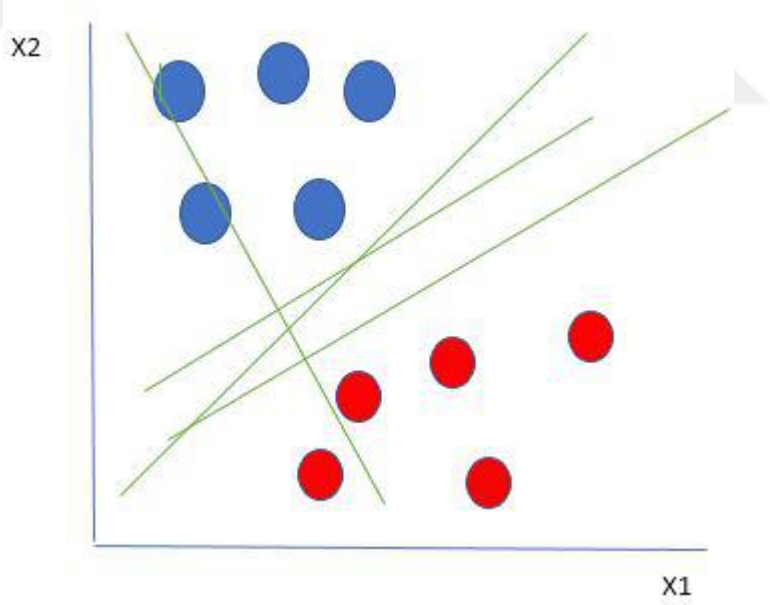
$g(z_t) = \theta z_t + \gamma[\,|z_t| - E|z_t|]$ ve $z_t = \varepsilon_t\sqrt{h_t}$ iken Denklem 2.14'deki gibi

$$\ln(h_t) = \omega + \sum_{i=1}^{q} \alpha_i g(z_{t-i}) + \sum_{j=1}^{p} \gamma_j \ln(h_{t-j}) \tag{2.14}$$

ifade edilir (SAS Help Center: GARCH Models, 2018).

2.8 Destek Vektör Makinesi (SVM)

Destek Vektör Makinesi (SVM) algoritmasının amacı, veri noktalarını belirgin bir şekilde sınıflandıran N boyutlu bir uzayda bir hiper düzlem bulmaktır. Hiper düzlemin boyutu, özelliklerin sayısına bağlıdır. Girdi özelliklerinin sayısı iki ise hiper düzlem sadece bir çizgidir. Girdi özelliklerinin sayısı üç ise hiper düzlem 2 boyutlu bir düzlem olur.



Şekil 2.1 SVM Marjin Yapısı (https://www.geeksforgeeks.org/support-vector-machine-algorithm/, 2021)

Şekil 2.1'deki gibi 2 farklı renkte noktaları sınıflandırmak için SVM algoritması en iyi hiper düzlemi seçerken aralarındaki mesafeye bakar. Bu mesafeye marjin adı verilir. Marjin değeri maksimum olan hiper düzlemi seçilir (GeeksForGeeks, 2021).

Fonksiyon yaklaşımı için SVM'nin temel fikri, x verisini doğrusal olmayan bir eşleme ile yüksek boyutlu bir özellik uzayına eşlemek ve ardından özellik uzayında doğrusal bir regresyon gerçekleştirmektir. Girdi verileri $x_i \in R$, (p toplam veri deseni sayısıdır) ve çıktı $y_i \in R$ ile birlikte n veri noktasından oluşan belirli bir eğitim kümesi ele alındığında SVM, aşağıdaki biçimde fonksiyona yaklaşır bu durum Denklem 2.15'deki gibi ifade edilebilir,

$$y(x) = w^T \varphi(x) + b \tag{2.15}$$

Burada, $\varphi(x)$, x girdi uzayından doğrusal olmayan bir şekilde eşlenen yüksek boyutlu özellik uzaylarını temsil eder. Katsayılar w ve b, düzenlileştirilmiş fonksiyonu en aza indirerek tahmin edilir (Okasha, 2014).

$$R(C) = \frac{1}{2}||w||^2 + \frac{C}{n}\sum_{i=1}^{n} L_t(d_i, y_i) \tag{2.16}$$

$$L_t(d_i, y_i) = \begin{cases} |d_i, y_i| - \varepsilon, & |d_i, y_i| \geq \varepsilon \\ 0 & \text{aksi halde} \end{cases} \tag{2.17}$$

w ve b'nin tahminini elde etmek için Denklem 2.16 ve Denklem 2.17'de verilen asal fonksiyona, pozitif aylak değişkenleri ξ ve ξ* ile Denklem 2.18'deki gibi dönüştürülür:

$$\text{Minimum } R(w, \xi*) = \frac{1}{2}||w||^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i + \xi^*{}_i) \tag{2.18}$$

kısıtlar;

$$\begin{cases} d_i - w\varphi(x_i) - b_i \leq \varepsilon + \xi_i \\ w\varphi(x_i) + b_i - y_i \leq \varepsilon + \xi^*{}_i \\ \xi_i, \xi^*{}_i \geq 0 \end{cases} \tag{2.19}$$

Birinci terim $\frac{1}{2}||w||^2$ ağırlıklar vektör normudur, d_i istenen değerdir ve C, ampirik hata ile düzenli terim arasındaki değiş tokuşu belirleyen düzenlileştirilmiş sabit olarak adlandırılır. ε, eğitim veri noktalarına yerleştirilen yaklaşıklık doğruluğuna eş değerdir. Burada aylak değişkenler, ξ ve ξ^* olarak tanımlanır. Lagrange çarpanlarını tanıtarak ve optimal kısıtlarından yararlanarak Denklem 2.19'da verilen karar fonksiyonu Denklem 2.20'deki açık forma sahiptir (Okasha, 2014):

$$y(x) = \sum_{i=1}^{n} d_i(a_i - a^*{}_i)K(x, x_i) + b \tag{2.20}$$

$(a_i) * (a^*{}_i) = 0$, $a_i \geq 0$ ve $a^*{}_i \geq 0$ eşitliklerini sağlarlar. Burada i = 1, 2, ... , n olmak üzere Denklem 2.21 'in ikili fonksiyonunu maksimize ederek elde edilirler ve aşağıdaki forma sahiptirler:

$$R(a_i - a^*{}_i) = \sum_{i=1}^{n} d_i(a_i - a^*{}_i) - \varepsilon \sum_{i=1}^{n}(a_i - a^*{}_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i - a^*{}_i)(a_j - a^*{}_j)K((x_i - x_j)) \tag{2.21}$$

Kısıtlar sırasıyla Denklem 2.22, Denklem 2.23 ve Denklem 2.24'te gösterilmiştir ;

$$\sum_{i=1}^{n}(a_i) = \sum_{i=1}^{n}(a^*{}_i) \tag{2.22}$$

$$0 \leq a_i \leq C, i = 1, 2, \dots, n \tag{2.23}$$

$$0 \leq a*_i \leq C, i = 1, 2, \dots, n \tag{2.24}$$

$K(x_i, x_j)$ çekirdek fonksiyonu olarak tanımlanır. Çekirdeğin değeri, $\varphi(x_i)$ ve $\varphi(x_j)$ öznitelik uzayındaki x_i ve x_j vektörlerinin iç çarpımına eşittir. Yani $(x_i, x_j) = \varphi(x_i)*\varphi(x_j)$ olur. Çekirdek işlevinin örnekleri Denklem 2.25, Denklem 2.26, Denklem 2.27 ve Denklem 2.28'deki gibi gösterilebilir (Okasha, 2014):

$$\text{Lineer: } K(x_i, x_j) = (x_i{}^T x_j) \tag{2.25}$$

$$\text{Sigmoid: } K(x_i, x_j) = \tanh(\gamma x_i{}^T x_j + r) \tag{2.26}$$

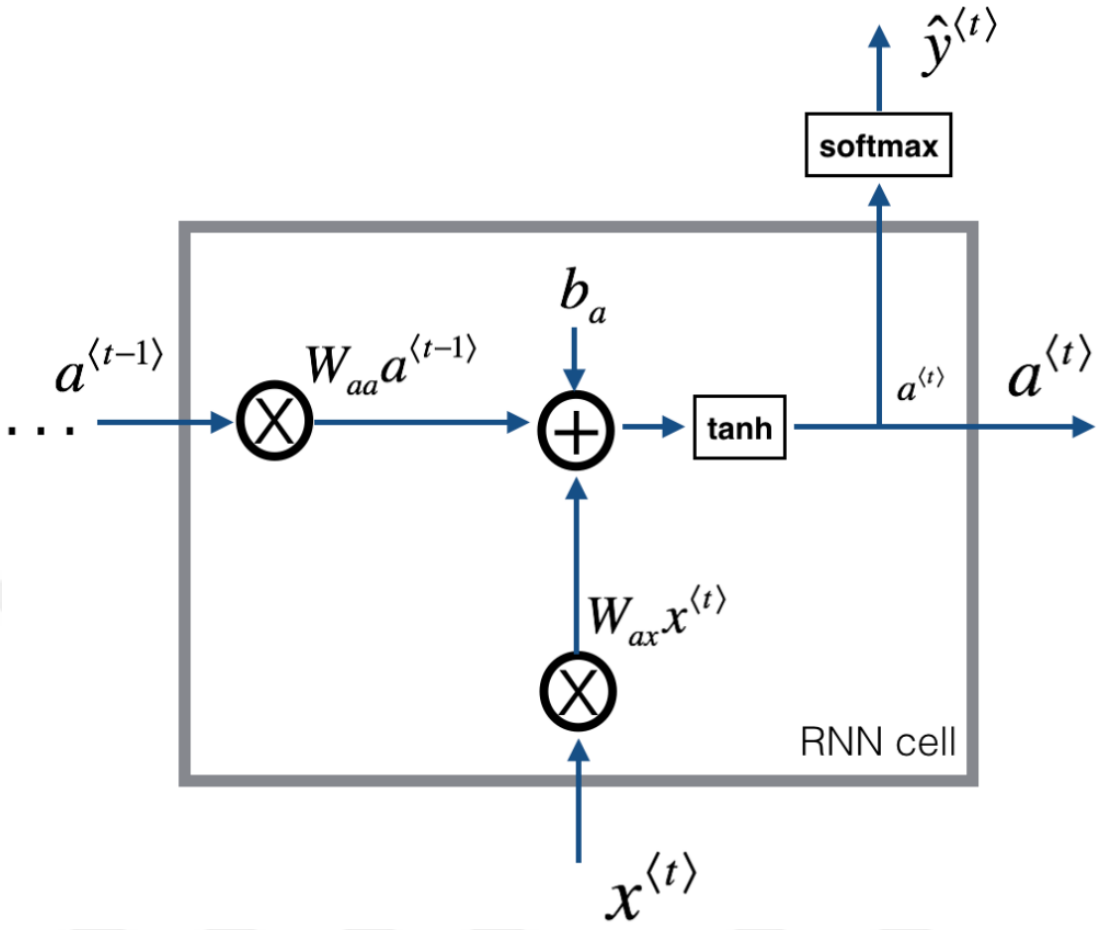
$$\text{Polinom: } K(x_i, x_j) = (\gamma x_i{}^T x_j + r)^4, \gamma > 0 \quad (2.27)$$

$$\text{Dairesel Baz Fonksiyonu(RBF): } K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0 \quad (2.28)$$

Burada γ, r ve d çekirdek parametreleridir. Çekirdek parametreleri, yüksek boyutlu özellik uzayı φ(x)'in yapısını örtük olarak tanımladığı ve böylece nihai çözümün karmaşıklığını kontrol ettiği için dikkatli bir şekilde seçilmelidir (Okasha, 2014).

2.9 Yinelemeli Sinir Ağı (RNN)

RNN yapısında sinir ağlarına gelen bilgi, katmanlardaki belirli ağırlık sabitleri ile ilişkilendirilerek sonuçta bir tahmin oluşturulur. Bu tahmin gerçek veriler ile karşılaştırıldığında bir hata payı oluşursa ağırlık sabitleri değiştirilerek yeniden tahmin oluşturulur. Bu sinir ağında asıl amaç hata payını düşürmektir. Tekrarlayan bir sinir ağı, y(t) = y(1), veri dizisini işlemek için uzmanlaşmış bir sinir ağıdır. Konuşma ve dil gibi sıralı girdiler içeren görevler için RNN'leri kullanmak genellikle daha iyidir. Bir Doğal Dil İşleme NLP (Natural Language Processing) probleminde, bir cümledeki bir sonraki kelimeyi tahmin etmek istiyorsanız ondan önceki kelimeleri bilmek önemlidir. RNN'ler, bir dizinin her öğesi için aynı görevi yerine getirdikleri ve çıktıları önceki hesaplamalara bağlı olduğu için tekrarlayan olarak adlandırılır. RNN'leri tercih etmenin bir başka sebebi de şimdiye kadar hesaplananlar hakkında bilgi toplayan bir "belleğe" sahip olmalarıdır (Nabi, 2019). Şekil 2.2'de RNN yapısı verilmektedir.

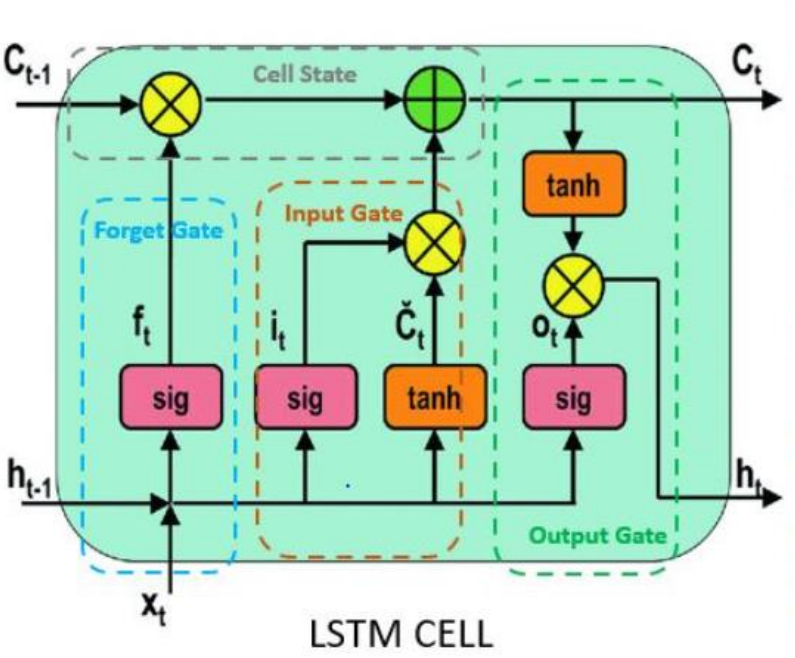


Şekil 2.2 RNN Yapısı (https://dltr.asmaamir.com/9-dizi-modelleri/2-rnnler, 2023)

Bir RNN hücresinde x_t girdiyi a_{t-1} bir önceki RNN hücresinin gizli durumunu girdi olarak alır ve a_t değerini çıktı olarak bir sonraki hücreye kullanılmak üzere verir.

2.10 Uzun Kısa Süreli Bellek (LSTM)

LSTM, veri dizisini ezberlemek amacıyla ek özelliklere sahip özel bir RNN türüdür. Her LSTM, veri akışının ve verilerin tutulup depolanmasının sağlandığı hücre veya sistem modülleri kümesinden oluşur. Hücreler geçmişten gelen verileri aktaran ve bunları bir araya getiren bir taşıma hattına benzer (Siami-Namini vd., 2018).



Şekil 2.3 Bir LSTM'in İç Yapısı (Singhal, 2020)

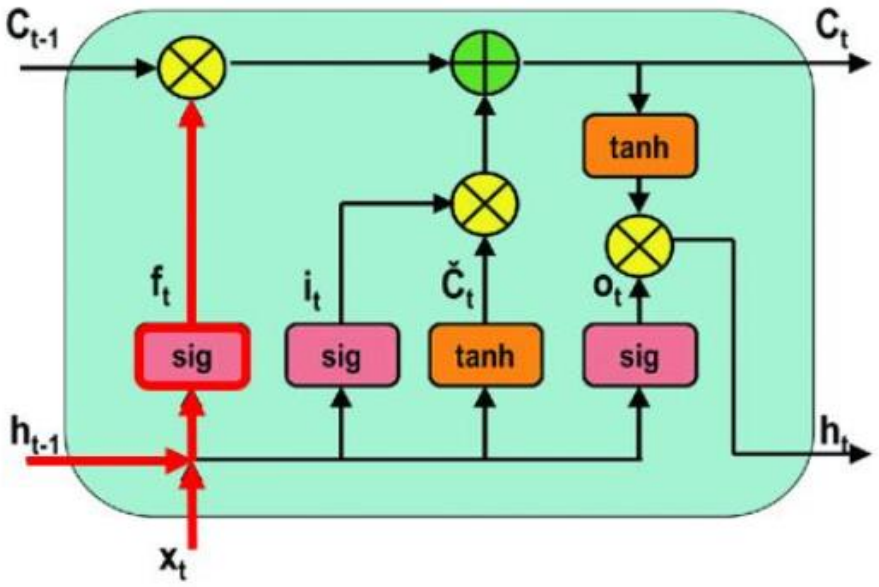
Şekil 2.3'te bir LSTM hücresi örneği verilmiştir. Her hücrede bazı kapıların kullanılmasına göre her hücredeki veriler atılabilir, filtrelenebilir veya sonraki hücrelere eklenebilir. Bu nedenle sigmoidal sinir ağı katmanına dayanan kapılar, hücrelerin verilerin geçmesine ya da atılmasına izin vermesini sağlar.

Her sigmoid katman, her veri segmentinin miktarını gösteren 0 ve 1 aralığında sayılar verir, her hücreden ne kadar veri geçeceğine karar verir. Daha doğrusu 0 değer tahmini ‘’hiçbir şeyin geçmesine izin verme ‘’ buna karşılık 1 değer tahmini ‘’her şeyin geçmesine izin ver ‘’ anlamına gelir. LSTM'de 3 tip kapı vardır; bunlar bütün LSTM modellerinde her hücrenin durumunu kontrol etmek amacıyla bulunur (Singhal, 2020).

2.10.1 Unutma Kapısı

Unutma kapısı hangi bilgilere dikkat edileceğine ve hangilerinin göz ardı edileceğine karar verir. x_t girdisinden elde edilen bilgi ve gizli durum (h_{t-1})’den gelen bilgiler sigmoid fonksiyonundan geçirilir. Sigmoid fonksiyonu 0 ile 1 arasında

değerler üretir. Bu üretilen değerlerin 0'a ve 1'e yakınlığına göre gerekli olup olmadığına karar verilir (0'a yakınsa eski çıktı değeri gerekli değildir). Bu f_t değeri daha sonra hücre tarafından çarpım için kullanılır (Singhal, 2020). Şekil 2.4'te unutma kapısı verilmiştir.



Şekil 2.4 Unutma Kapısı (Singhal, 2020)

Denklem 2.29'da unutma kapısı verilmiştir.

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f \quad (2.29)$$

Denklem 2.29'da kullanılan terim ve fonksiyonlar:

t: Zaman

f_t: t anındaki unutma kapısı

x_t: Girdi

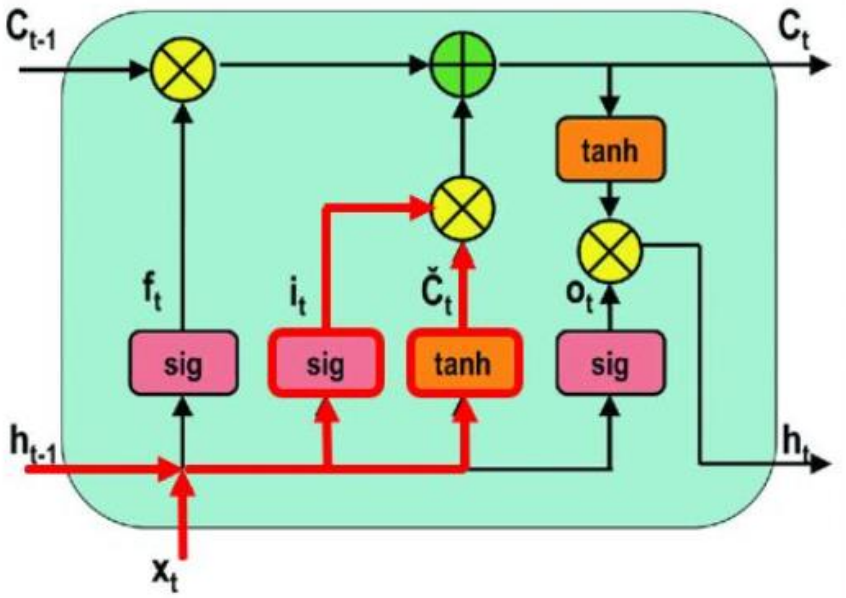
h_{t-1}: Önceki gizli durum

W_f: Unutma kapısı ile girdi kapısı arasındaki ağırlık matrisi

b_f: Bayes vektörünü ifade etmektedir.

2.10.2 Girdi Kapısı

Girdi kapısı, yeni gelen verilerden hangisinin depolanacağına karar verir. İlk olarak "girdi kapısı katmanı" adı verilen bir sigmoid katman hangi verilerin değiştirileceğini seçer. Ardından *tanh* katmanı bu duruma eklenebilecek yeni aday değerlerin bir vektörünü oluşturur (Singhal, 2020). Şekil 2.5'te girdi kapısı verilmiştir.



Şekil 2.5 Girdi Kapısı (Singhal, 2020)

Denklem 2.30 ve Denklem 2.31'de girdi kapısı (i_t) ve tanh fonksiyonu ile oluşturulan değer ($\tilde{C}_t$) verilmiştir.

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i \tag{2.30}$$

$$\tilde{C}_t = \tanh (W_c . [h_{t-1}, x_t] + b_c \tag{2.31}$$

Denklem 2.30 ve Denklem 2.31'de kullanılan terim ve fonksiyonlar:

t: Zaman

i_t: t anında girdi kapısı

W_i: Girdi ve çıktı kapısı arasındaki sigmoid operatör ağırlık matrisi

b_i: Bayes vektörü

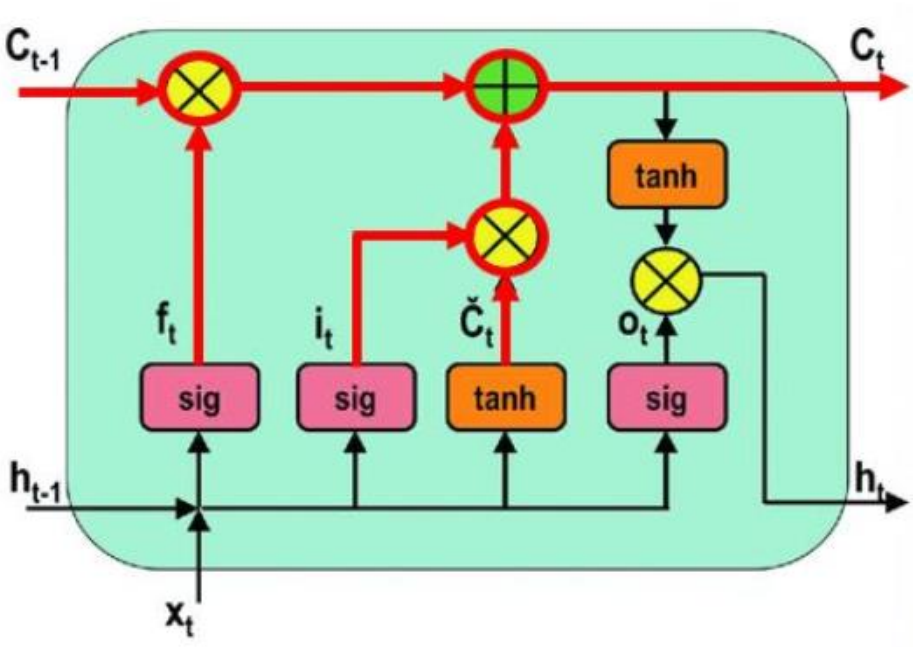
$\tilde{C}_t$: tanh tarafından oluşturulan değer

W_c: Hücre durum bilgisi ile ağ çıktısı arasındaki tanh operatör ağırlık matrisi

b_c: Bayes vektörünü ifade etmektedir.

2.10.3 Hücre Durumu

Ağ, unutma kapısı ve girdi kapısından yeterli bilgiye sahiptir. Bir sonraki adım, hücre durumundaki yeni gelen veriye ne yapılacağına karar vermek ve depolamaktır. Önceki hücre durumu (C_{t-1}), unutma vektörü (f_t) ile çarpılır. Sonuç 0 ise değerler hücre durumunda bırakılacaktır. Daha sonra ağ, girdi vektörü (i_t)'nin çıktı durumunu alır ve adım adım toplayarak yeni bir hücre durumu (C_t) değerini oluşturur (Singhal, 2020). Şekil 2.6'da hücre durumu verilmiştir.



Şekil 2.6 Hücre Durumu (Singhal, 2020)

Denklem 2.32'de hücre durumu (C_t) verilmiştir.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.32}$$

Denklem 2.32'de kullanılan fonksiyon ve terimler:

t: Zaman

C_t: Hücre durumu bilgisi

f_t: t anındaki unutma kapısı

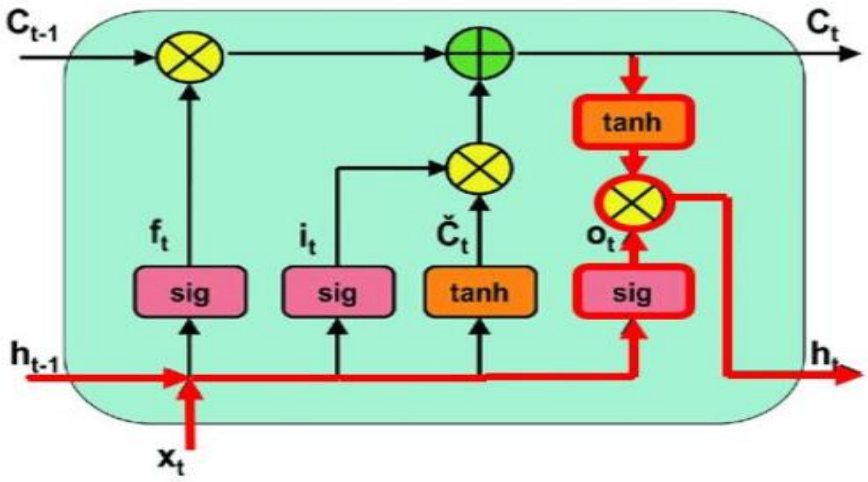
i_t: t anındaki girdi kapısı

C_{t-1}: Önceki hücre durumu

$\tilde{C}_t$: tanh tarafından oluşturulan değeri ifade etmektedir.

2.10.4 Çıktı Kapısı

Çıktı kapısı, bir sonraki gizli durumun değerini belirler. Bu gizli durum önceki girdilerin bilgilerini içerir. Şu anki durum ve önceki gizli durum sigmoid fonksiyondan geçer. Yeni hücre durumu tanh fonksiyonundan geçer. Bu iki çıktı birbiri ile çarpılır. Son değere göre ağ gizli durumunun (h_t) hangi değeri alacağına karar verir. Bu gizli durum tahmin için kullanılır (Singhal, 2020). Şekil 2.7'de çıktı kapısı verilmiştir.



Şekil 2.7 Çıktı Kapısı (Singhal, 2020)

Denklem 2.33 ve Denklem 2.34'te çıktı kapısı (o_t) ve gizli durum (h_t) verilmiştir.

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o \quad (2.33)$$

$$h_t = o_t * \tanh(C_t) \quad (2.34)$$

Denklem 2.33 ve Denklem 2.34'te kullanılan fonksiyon ve terimler:

t: Zaman

o_t: t anındaki çıktı kapısı

W_o: Çıktı kapısının ağırlık matrisi

b_o: Bayes vektörü

h_t: LSTM çıktısını ifade etmektedir.

BÖLÜM ÜÇ
HİBRİT MODEL

Bölüm 2'de belirtilen tüm modeller, doğrusal ve doğrusal olmayan alanlarda başarılıdır. Ancak hiçbiri tek başına tüm durumlar için en iyi sonucu vermez. ARIMA, ARIMA-GARCH doğrusal problemler için daha uygundur. Öte yandan LSTM ve SVM modelleri doğrusal problemler için karmaşık sonuçlar verebilir.

L_t, doğrusal bileşendir ve N_t, doğrusal olmayan bileşen olarak tanımlandığında t anındaki gözlem değeri Denklem 3.1'deki gibi,

$$y_t = L_t + N_t \tag{3.1}$$

elde edilir.

Bu iki bileşen gözlemlerden tahmin edilir. Doğrusal bileşenin tahmin edilmesi için ARIMA ve ARIMA-GARCH modelleri kullanılmıştır. Daha sonra e_t değerleri Denklem 3.2'deki gibi hesaplanmıştır,

$$e_t = y_t - \hat{L} \tag{3.2}$$

Denklem 3.2'de $\hat{L}$, t zamanı için tahmin edilen değerdir. Bundan sonra e_t değerleri kullanılarak LSTM, SVM gibi doğrusal olmayan modelleme algoritmaları ile doğrusal olmayan ilişkileri açıklamak için kullanılmıştır.

$$e_t = f(e_{t-1}, e_{t-2}, \ldots e_{t-n}) + \epsilon_t, \tag{3.3}$$

Denklem 3.3'te f, LSTM veya SVM tarafından belirlenen doğrusal olmayan bir fonksiyondur ve ϵ_t, rastgele hata. $\widehat{N}_t$'nin Denklem 3.4'te birleşik tahmin,

$$y_t = \hat{L}_t + \widehat{N}_t \tag{3.4}$$

olarak elde edilir. Kısaca, önerilen metodolojiye ulaşmak için iki adım vardır. Birinci adım: doğrusal kısım için ARIMA ve ARIMA-GARCH modeli kullanılır. İkinci adım; ARIMA veya ARIMA-GARCH modelinden elde edilen artıklar, LSTM veya SVM modellerinin geliştirmesi için kullanılır.

BÖLÜM DÖRT
UYGULAMA

4.1 Uygulama Kapsamı

Uygulamada NASDAQ (ABD), DAX (Almanya), NISEI (Hindistan), BIST100 (Türkiye) hisse senedi endekslerinin 1 Ocak 2017 ile 18 Ekim 2021 tarihleri arasındaki günlük kapanış değerleri kullanılmıştır. Dünyanın en büyük ekonomilerine sahip Amerika Birleşik Devletleri veya Almanya'daki bazı önemli ekonomik gelişmeler gelişmekte olan ekonomileri etkilemektedir. Bu veri setlerinin seçimindeki temel amaç, dünyanın önde gelen finansal piyasaları ile gelişmekte olan piyasalar arasındaki zaman serisi tahmin modellerini karşılaştırmaktır. Bu modellerin zaman serisi özellikleri açısından birbirinden farklılaşıp farklılaşmadığı araştırılmıştır. Bu uygulamada hisse senedi endeks değerlerinin tahmini için en uygun modelin seçilmesi amaçlanmıştır. Veri kümelerinde, verilerin log-getiri değerleri üzerinde çalışılmıştır. Zaman serisi tahmininde çeşitli modellerin performansları araştırılmıştır. Bu modeller; ARIMA, EGARCH, LSTM, SVM ve hibrit modeller; ARIMA-SVM, ARIMA-EGARCH-SVM, ARIMA-EGARCH-LSTM ve ARIMA-LSTM'dir. ARIMA-EGARCH modellemenin ilk adımı için dört modelin tümü benzer model davranışı göstermiştir. ARIMA(1, 1) ve EGARCH(1, 1) ile model tahminlenmiştir ardından ikinci adım olarak LSTM ve SVM uygulanmıştır. Bu modellerin tahmin performansları için her bir veri, test verileri ve eğitim verileri olarak adlandırılan iki kısıma ayrılmıştır. Eğitim verileri modelin geliştirilmesi için test verileri ise tahmin edilen modeli değerlendirmek için kullanılmıştır. Test verisi ve eğitim verisi, dört logaritmik dönüştürülmüş veri seti için sırasıyla 362 gözlem ve 845 gözlem olarak hazırlanmıştır. Modeller eğitim veri seti üzerinde eğitilmiştir ve ardından test veri seti bazında model performansları ölçülmüştür. Program olarak R kullanılmıştır. Kod detaylarına github adresinden https://github.com/cberkbulut/Zaman-Serisi/tree/main/Zaman%20Serisi%203 ulaşılabilir.

4.2 Karşılaştırma Metrikleri

Model performanslarını, Tablo 4.1'de verilen Ortalama Mutlak Hata (MAE), Ortalama Mutlak Yüzde Hatası (MAPE), Ortalama Kare Hata (MSE) ve Karekök Ortalama Kare Hata (RMSE) gibi bazı tahmin doğruluk ölçütleriyle karşılaştırılarak sonuçlara ulaşılmak istenmiştir.

Tablo 4.1 Performans Metrikleri

MSE	$\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}$
MAE	$\frac{1}{N}\sum_{i=t}^{N}\lvert x_i - \hat{x}_i \rvert$
MAPE	$\frac{100}{N}\sum_{i=t}^{N}\frac{\lvert\, x_i - \hat{x}_i \,\rvert}{\lvert\, x_i \,\rvert}$

Ortalama Hata Kareleri (MSE), tahmin edilen değerler ile gözlemlenen değerlerin ortalaması arasındaki farkların karesi alınarak hesaplanır.

Ortalama Hata Kareleri Karekökü (RMSE), tahmin edilen değerler ile gözlemlenen değerlerin ortalaması arasındaki farkların karesinin karekökleri alınarak hesaplanır. RMSE, farklı modeller arasında tahmin hatalarının doğruluğunun karşılaştırılması için kullanılır (Chen, 2020).

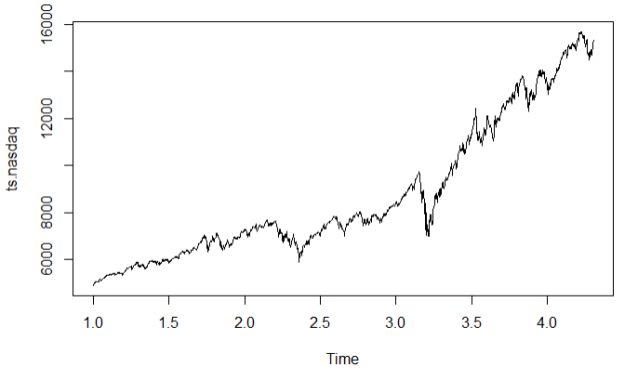
Mutlak Yüzdesel Hata (MAE), tahmin edilen değerler ile gözlemlenen değerlerin ortalaması arasındaki farkın mutlak değeri alınarak hesaplanır.

Ortalama Mutlak Yüzdesel Hata (MAPE), tahmin edilen değerler ile gözlemlenen değerlerin ortalaması arasındaki farkın, gözlem sayısına bölünüp 100 ile çarpılması ile elde edilen yüzdelik bir değerdir (Chen, 2020).

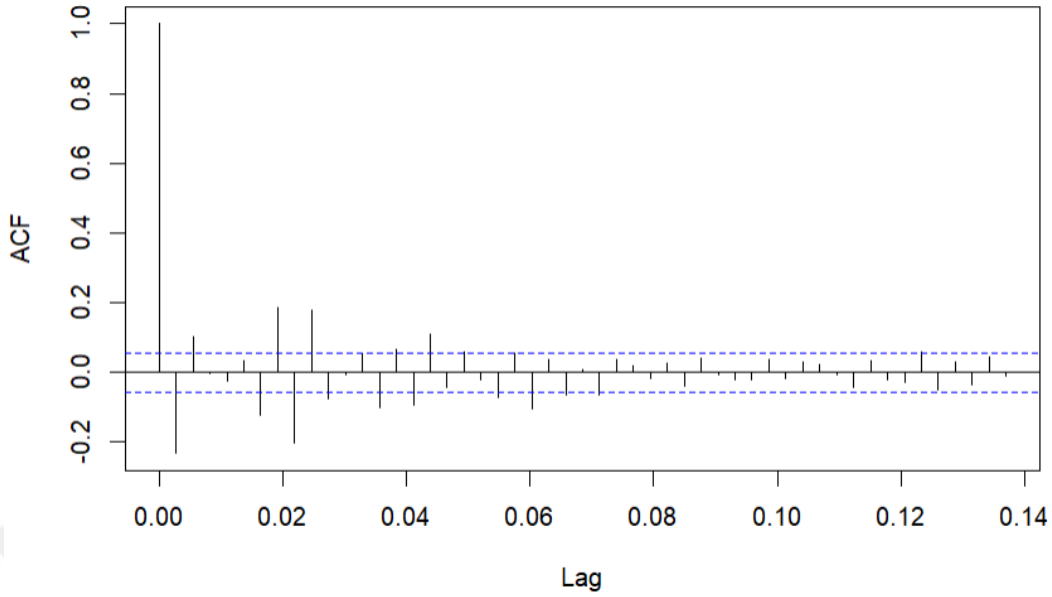
4.3 Veri Setleri

4.3.1 NASDAQ

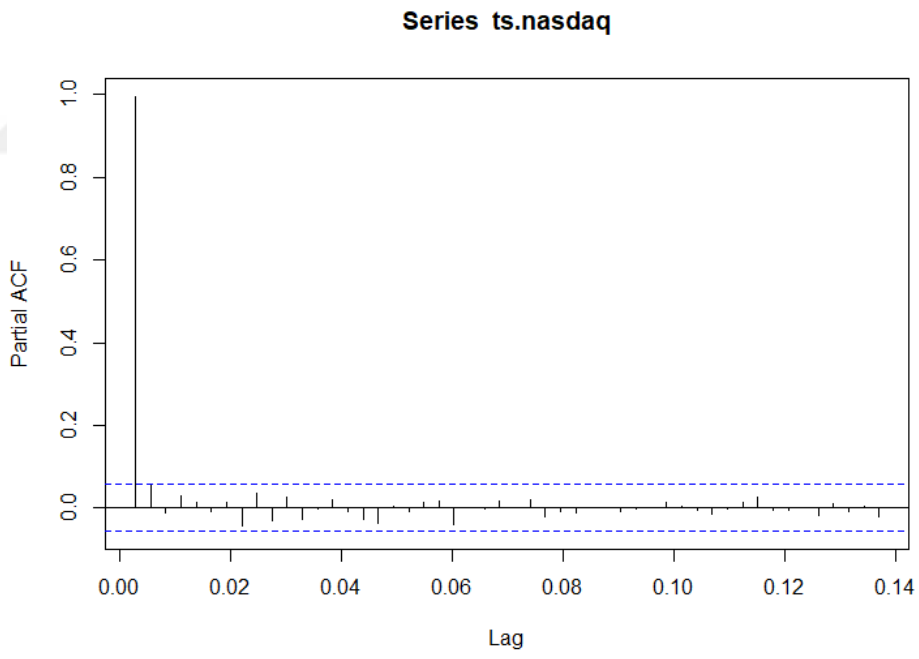
Veri seti olarak NASDAQ borsa endeksinin günlük kapanış değerleri kullanılmıştır. 1307 adet verinin %70'i eğitim seti, %30'u test seti olarak ayrılmıştır. Şekil 4.1'de verinin durağan olmadığı trend etkisi altında olduğu görülmektedir. ACF ve PACF grafiklerine de bakıldığında serinin durağan olmadığına karar verilmiştir. Seriyi durağanlaştırmak için logaritma dönüşümü ve fark alma işlemleri uygulanmıştır.



Şekil 4.1 NASDAQ Veri Dağılımı



Şekil 4.2 NASDAQ ACF Grafiği



Şekil 4.3 NASDAQ PACF Grafiği

Şekil 4.2 ve 4.3'e bakıldığında zaman serisinin durağanlaştığı ARIMA(1, 1), ARIMA(5, 1) ve ARIMA(6, 1) modellerinin denenebileceği sonucu çıkarılmıştır.

Logaritma dönüşümü ve fark alınmış seride ARIMA(1, 1) modelinin kullanılmasına karar verilmiştir.

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,1)
Mean Model      : ARFIMA(1,0,1)
Distribution    : nig

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.00078    0.000165   4.7291  2.0e-06
ar1      0.52527    0.043161  12.1701  0.0e+00
ma1     -0.60025    0.041871 -14.3355  0.0e+00
omega   -0.29314    0.037795  -7.7562  0.0e+00
alpha1  -0.13839    0.022554  -6.1361  0.0e+00
beta1    0.96705    0.004177 231.4923  0.0e+00
gamma1   0.22602    0.031681   7.1343  0.0e+00
skew    -0.39877    0.049890  -7.9930  0.0e+00
shape    1.36681    0.323810   4.2210  2.4e-05

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.00078    0.000127   6.1556 0.000000
ar1      0.52527    0.011598  45.2880 0.000000
ma1     -0.60025    0.013375 -44.8791 0.000000
omega   -0.29314    0.023289 -12.5871 0.000000
alpha1  -0.13839    0.026062  -5.3102 0.000000
beta1    0.96705    0.002565 377.0527 0.000000
gamma1   0.22602    0.033923   6.6628 0.000000
skew    -0.39877    0.048557  -8.2125 0.000000
shape    1.36681    0.361432   3.7816 0.000156

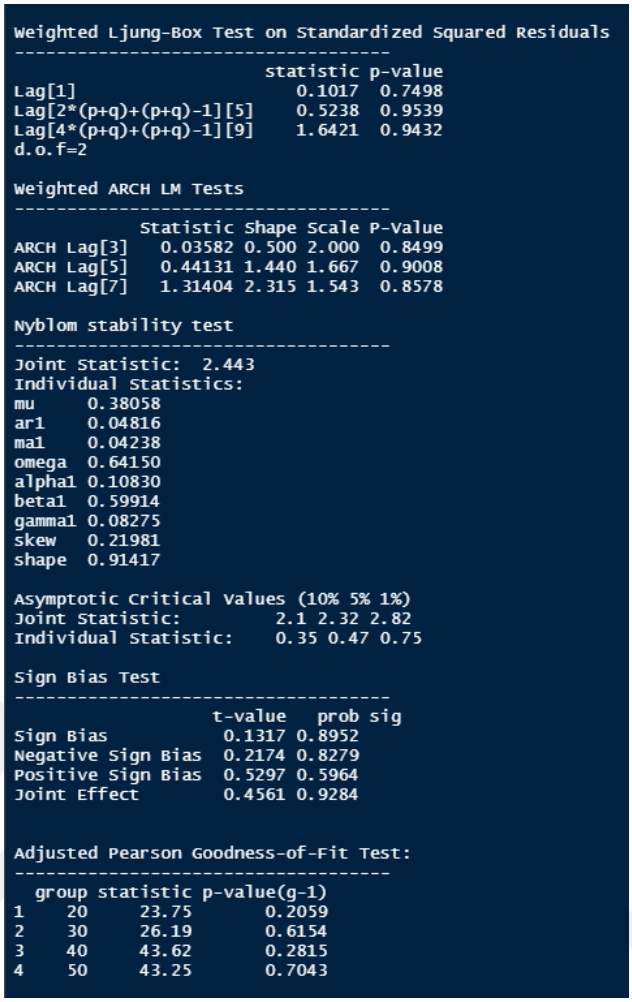
LogLikelihood : 3819.608

Information Criteria
------------------------------------

Akaike       -6.3194
Bayes        -6.2814
Shibata      -6.3195
Hannan-Quinn -6.3051

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.2097  0.6470
Lag[2*(p+q)+(p+q)-1][5]    2.0298  0.9523
Lag[4*(p+q)+(p+q)-1][9]    5.2431  0.4000
d.o.f=2
H0 : No serial correlation
```

Şekil 4.4 NASDAQ GARCH Test



```
Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1017  0.7498
Lag[2*(p+q)+(p+q)-1][5]    0.5238  0.9539
Lag[4*(p+q)+(p+q)-1][9]    1.6421  0.9432
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]   0.03582 0.500 2.000  0.8499
ARCH Lag[5]   0.44131 1.440 1.667  0.9008
ARCH Lag[7]   1.31404 2.315 1.543  0.8578

Nyblom stability test
------------------------------------
Joint Statistic:  2.443
Individual Statistics:
mu      0.38058
ar1     0.04816
ma1     0.04238
omega   0.64150
alpha1  0.10830
beta1   0.59914
gamma1  0.08275
skew    0.21981
shape   0.91417

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:          2.1 2.32 2.82
Individual Statistic:     0.35 0.47 0.75

Sign Bias Test
------------------------------------
                    t-value   prob sig
Sign Bias           0.1317 0.8952
Negative Sign Bias  0.2174 0.8279
Positive Sign Bias  0.5297 0.5964
Joint Effect        0.4561 0.9284


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     23.75       0.2059
2    30     26.19       0.6154
3    40     43.62       0.2815
4    50     43.25       0.7043
```

Şekil 4.5 NASDAQ GARCH Test 2

GARCH(1, 1) modeli sonuçlarına bakıldığında modelde parametreler $\alpha = 0{,}05$ düzeyinde anlamlıdır. ARCH Lagrange Multiplier testi hipotezi,

H_0: Hata kareleri arasında otokorelasyon yoktur. (4.1)

H_1: Hata kareleri arasında otokorelasyon vardır. (4.2)

Denklem 4.1 ve Denklem 4.2'deki gibi gösterilebilir. Ağırlıklandırılmış Ljung-Box testi hipotezi,

H_0: Hata kareleri arasında otokorelasyon yoktur. (4.3)

H_1: Hata kareleri arasında otokorelasyon vardır. (4.4)

Denklem 4.3 ve Denklem 4.4'teki gibi gösterilebilir.

Kurulan modelin yeterli olup olmadığına karar vermek için artık kareleri üzerinde ağırlıklandırılmış Ljung-Box testine ve ARCH Lagrange Multiplier (ARCH-LM)

testine bakılabilir. Bu durumda $\alpha = 0{,}05$ anlamlılık düzeyinde hata kareleri arasında otokorelasyon olmadığı kararı verilmiştir ve seçilen modelin uygun olduğu kararı verilebilir. Bu sebeple GARCH(1, 1) uygulanması uygundur.

Doğrusal yapılar modellendikten sonra doğrusal olmayan yapının modellemesi için SVM ve LSTM kullanılmıştır. LSTM modelinde epoch sayısı 25, batch_size 128, shuffle özelligi false olacak şekilde çalıştırılmıştır. Batch size modele girdi olarak verilen setin kaçarlı olacağını, epoch sayısı ise bu girdilerin kaç kez modele gireceğini belirtmektedir. SVM modelinin özellikleri Şekil 4.6'da gösterilmiştir.

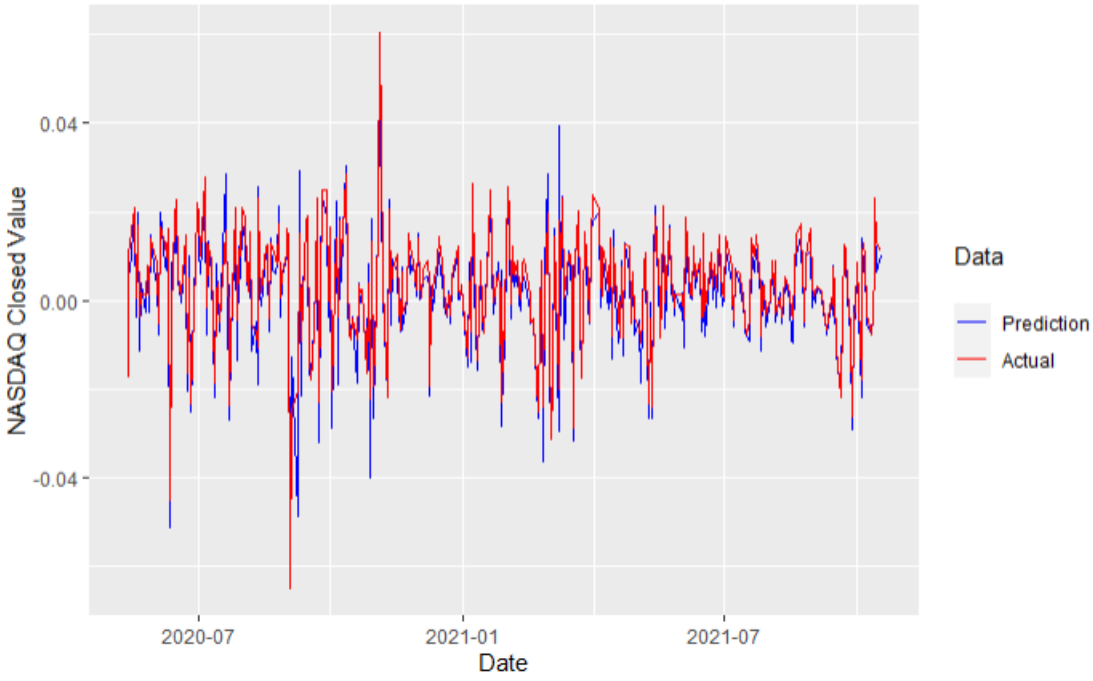
```
svm_garch_tune<-tune.svm(
     test[,2]~ e_garch,
      data = test,
      gamma = 2 ^ (-1:1),
      cost = 2 ^ (2:8),
      epsilon = seq(0, 1, 0.1),
      kernel = "radial"
```

Şekil 4.6 NASDAQ SVM Model Özellikleri

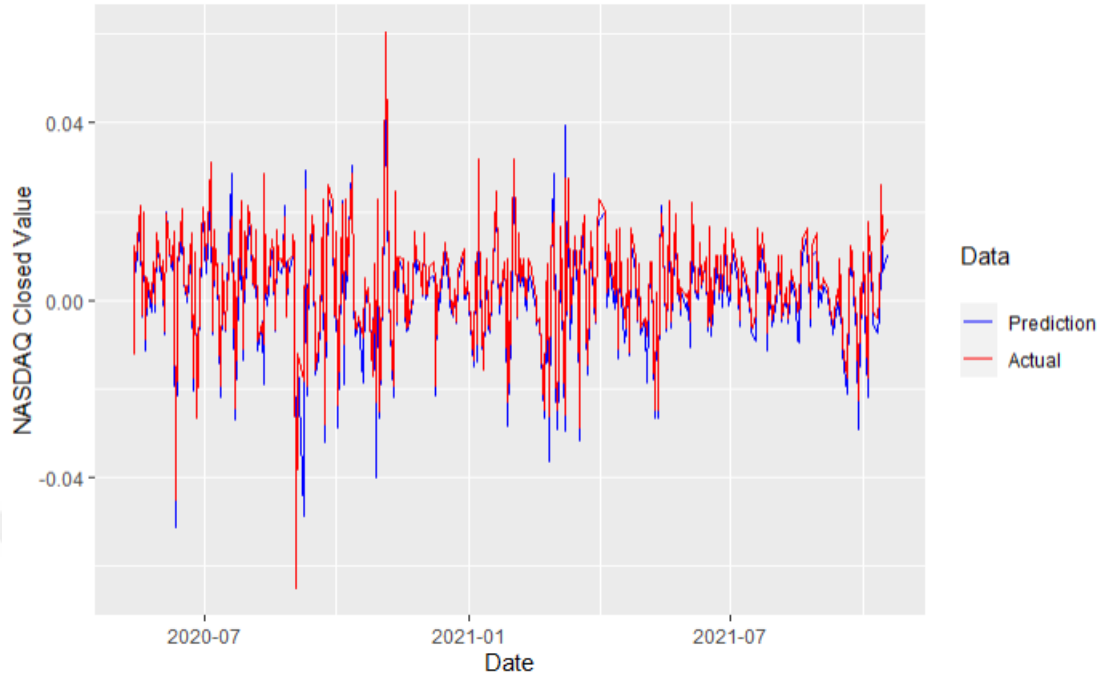
Tablo 4.2'de NASDAQ veri setine ait modellerin hata değerleri gösterilmiştir. Bu hata değerleri arasından en küçük değerleri ARIMA-E-GARCH-SVM modeli (MAE= 0,00244) ve ARIMA-SVM (MAE= 0,003322) modellerinin test verisine göre yapılan öngörü performans grafikleri Şekil 4.7 ve Şekil 4.8'de gösterilmiştir. Grafiklere bakıldığında da hata değerlerindeki yakınlığı gösterdiği ancak ARIMA-EGARCH-SVM modelinin tahminleme konusunda daha başarılı olduğu gözlenmiştir.

Tablo 4.2 NASDAQ Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri

Model	MAE	MAPE	RMSE
LSTM	0,01447	3,91177	0,01963
ARIMA-E-GARCH	0,00986	59,06250	0,01329
ARIMA	0,01001	1,52985	0,01344
SVM	0,01009	1,33651	0,01435
ARIMA_SVM	0,003322	1,33289	0,00468
ARIMA-E-GARCH_SVM	0,00244	0,59846	0,00366
ARIMA-E-GARCH_LSTM	0,00993	1,98392	0,01341
ARIMA_LSTM	0,01306	4,29118	0,01772



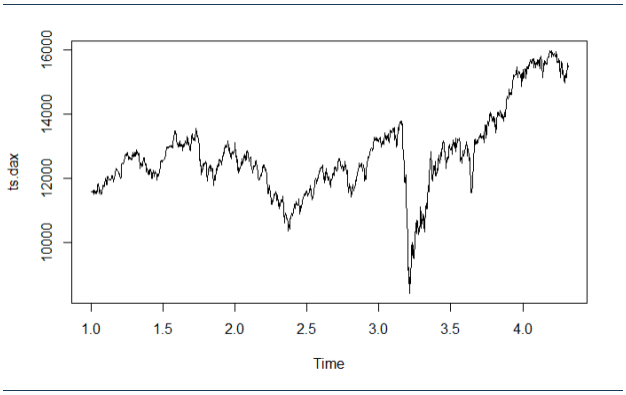
Şekil 4.7 ARIMA-SVM NASDAQ Logaritmik Dönüştürülmüş Veri Tahmini



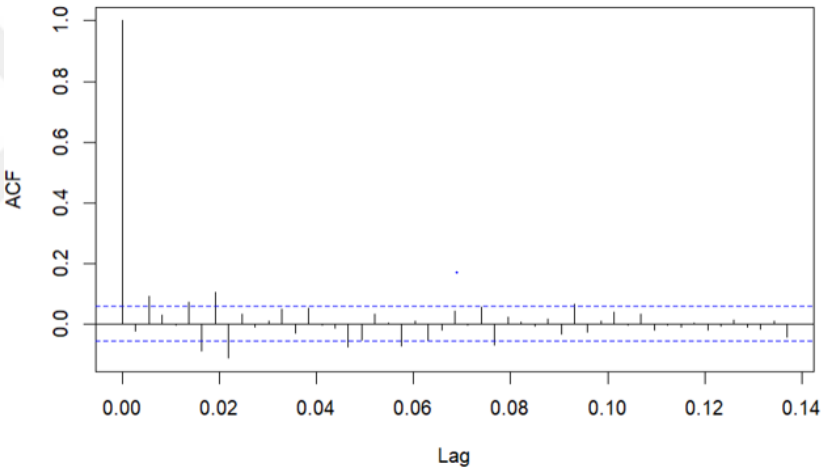
Şekil 4.8 ARIMA-E-GARCH-SVM NASDAQ Logaritmik Dönüştürülmüş Veri Tahmini

4.3.2 DAX

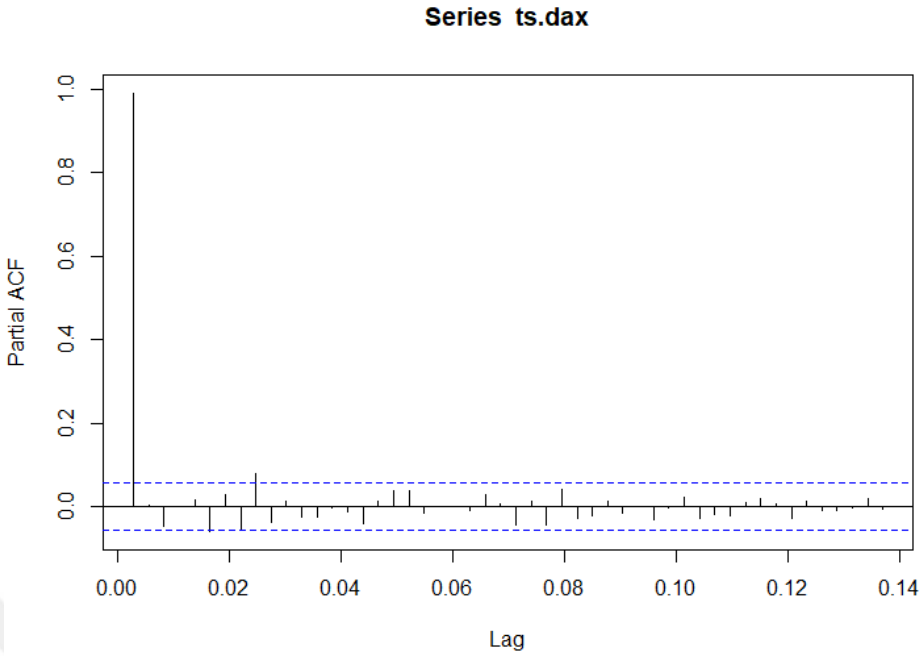
Veri seti olarak DAX borsa endeksinin günlük kapanış değerleri kullanılmıştır. 1307 adet verinin %70'i eğitim seti, %30'u test seti olarak ayrılmıştır. Şekil 4.9'da verinin durağan olmadığı trend etkisi altında olduğu görülmektedir. ACF ve PACF grafiklerine de bakıldığında serinin durağan olmadığına karar verilmiştir. Seriyi durağanlaştırmak için logaritma dönüşümü ve fark alma işlemleri uygulanmıştır.



Şekil 4.9 DAX Veri Dağılımı

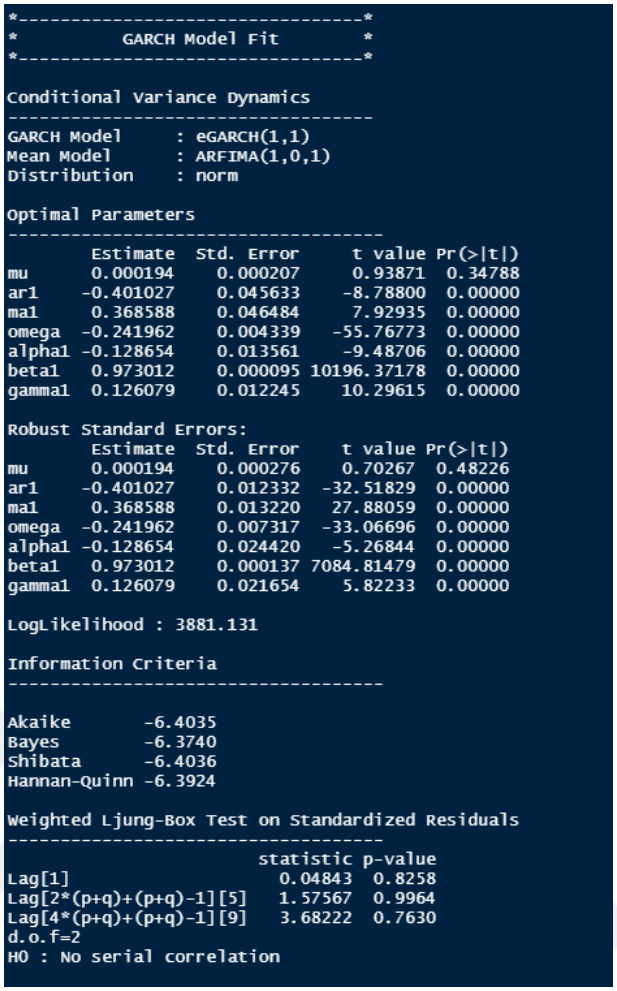


Şekil 4.10 DAX ACF Grafiği



Şekil 4.11 DAX PACF Grafiği

Şekil 4.10 ve Şekil 4.11'de belirtilen otokorelasyon ve kısmi otokorelasyon grafiklerine bakıldığında zaman serisinin Box-Jenkins metodolojisine göre durağanlaştığı ARIMA(1, 1), ARIMA(5, 1) ve ARIMA(6, 1) modelleri tanısal teşhis (diagnostic checking) ile uygunluğu sınanabilir.



```
*----------------------------------*
*          GARCH Model Fit         *
*----------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : eGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution     : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000194    0.000207     0.93871  0.34788
ar1    -0.401027    0.045633    -8.78800  0.00000
ma1     0.368588    0.046484     7.92935  0.00000
omega  -0.241962    0.004339   -55.76773  0.00000
alpha1 -0.128654    0.013561    -9.48706  0.00000
beta1   0.973012    0.000095 10196.37178  0.00000
gamma1  0.126079    0.012245    10.29615  0.00000

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000194    0.000276    0.70267  0.48226
ar1    -0.401027    0.012332  -32.51829  0.00000
ma1     0.368588    0.013220   27.88059  0.00000
omega  -0.241962    0.007317  -33.06696  0.00000
alpha1 -0.128654    0.024420   -5.26844  0.00000
beta1   0.973012    0.000137 7084.81479  0.00000
gamma1  0.126079    0.021654    5.82233  0.00000

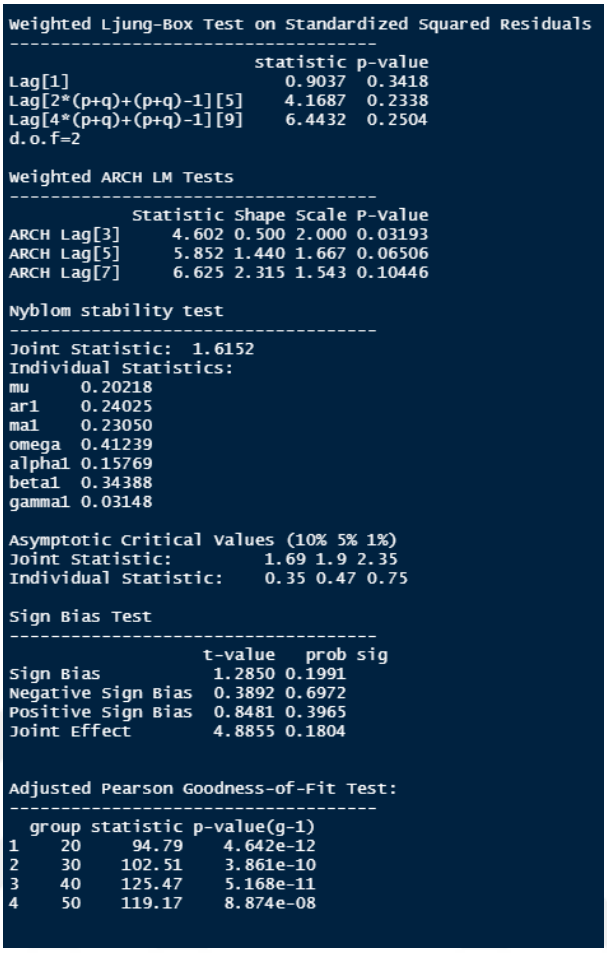
LogLikelihood : 3881.131

Information Criteria
------------------------------------

Akaike       -6.4035
Bayes        -6.3740
Shibata      -6.4036
Hannan-Quinn -6.3924

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                    0.04843  0.8258
Lag[2*(p+q)+(p+q)-1][5]   1.57567  0.9964
Lag[4*(p+q)+(p+q)-1][9]   3.68222  0.7630
d.o.f=2
H0 : No serial correlation
```

Şekil 4.12 DAX GARCH Test



Şekil 4.13 DAX GARCH Test 2

GARCH modeli sonuçlarına bakıldığında modele GARCH(1, 1) uygulanması kararı verilmiştir.

Kurulan modelin yeterli olup olmadığına karar vermek için artık kareleri üzerinde ağırlıklanıdırılmış Ljung-Box testine ve ARCH Lagrange Multiplier (ARCH-LM) testine bakılabilir. Bu durumda $\alpha = 0{,}05$ anlamlılık düzeyinde hata kareleri arasında otokorelasyon olmadığı kararı verilmiştir ve seçilen modelin uygun olduğu kararı verilebilir. Bu sebeple GARCH(1, 1) uygulanması uygundur.

Doğrusal yapılar modellendikten sonra doğrusal olmayan yapının modellemesi için SVM ve LSTM kullanılmıştır. LSTM modelinde epoch sayısı 25, batch_size 128,

shuffle özelligi false olacak şekilde çalıştırılmıştır. SVM modelinin özellikleri Şekil 4.14'te gösterilmiştir.

```
svm_garch_tune<-tune.svm(
      test[,2]~ e_garch,
       data = test,
       gamma = 2 ^ (-1:1),
       cost = 2 ^ (2:8),
       epsilon = seq(0, 1, 0.1),
       kernel = "radial"
```

Şekil 4.14 DAX SVM Model Özellikleri

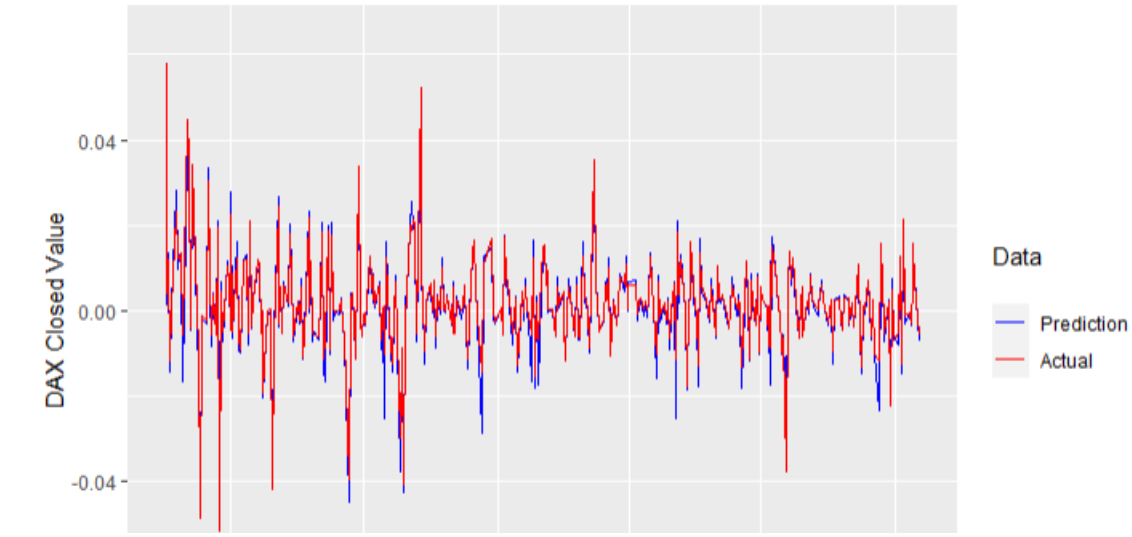
Tablo 4.3'te DAX veri setine ait modellerin hata değerleri gösterilmiştir. Bu hata değerleri arasından en küçük değerleri veren ARIMA-SVM (MAE= 0,00158) ve ARIMA-E-GARCH-SVM (MAE= 0,00166) modellerinin test verisine göre yapılan öngörü performans grafikleri Şekil 4.15 ve Şekil 4.16'da gösterilmiştir. Grafiklere bakıldığında da hata değerlerindeki yakınlığı gösterdiği ancak ARIMA-SVM modelinin tahminleme konusunda daha başarılı olduğu gözlenmiştir.

Tablo 4.3 DAX Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri

Model	MAE	MAPE	RMSE
LSTM	0,01305	8,34821	0,01802
ARIMA-E-GARCH	0,00827	76,18057	0,01204
ARIMA	0,00832	1	0,01209
SVM	0,00901	1,17050	0,01308
ARIMA_SVM	0,00158	1,23456	0,00276
ARIMA-E-GARCH_SVM	0,00166	1,15763	0,00279
ARIMA-E-GARCH_LSTM	0,00830	2,82786	0,01205
ARIMA_LSTM	0,01286	4,23613	0,01765



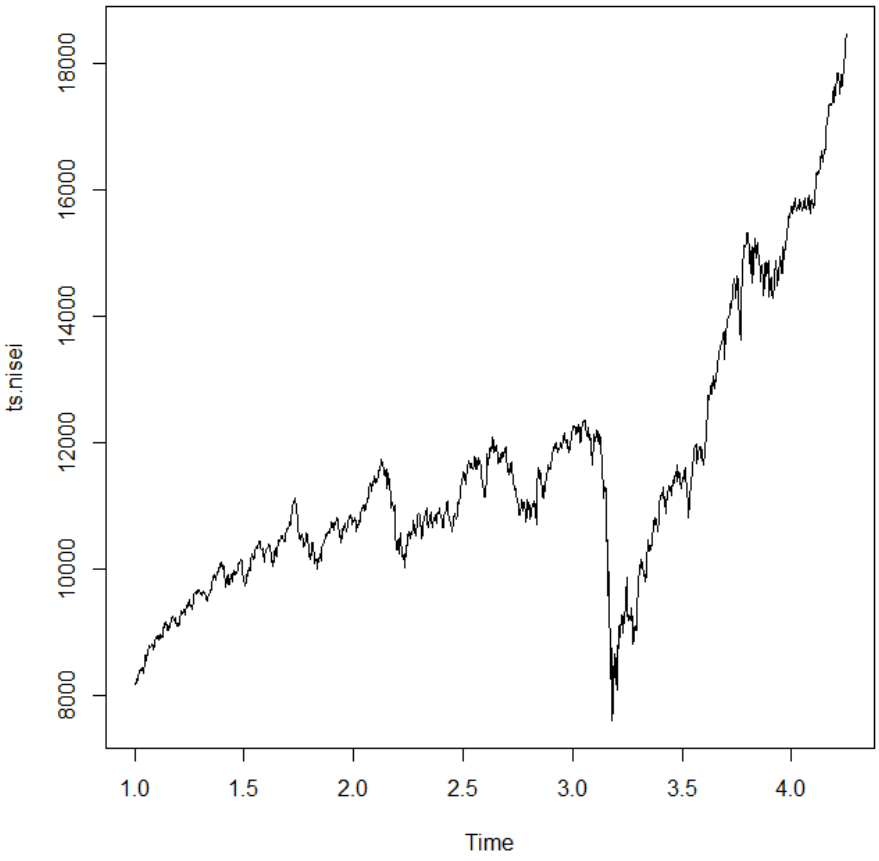
Şekil 4.15 ARIMA-SVM DAX Logaritmik Dönüştürülmüş Veri Tahmini



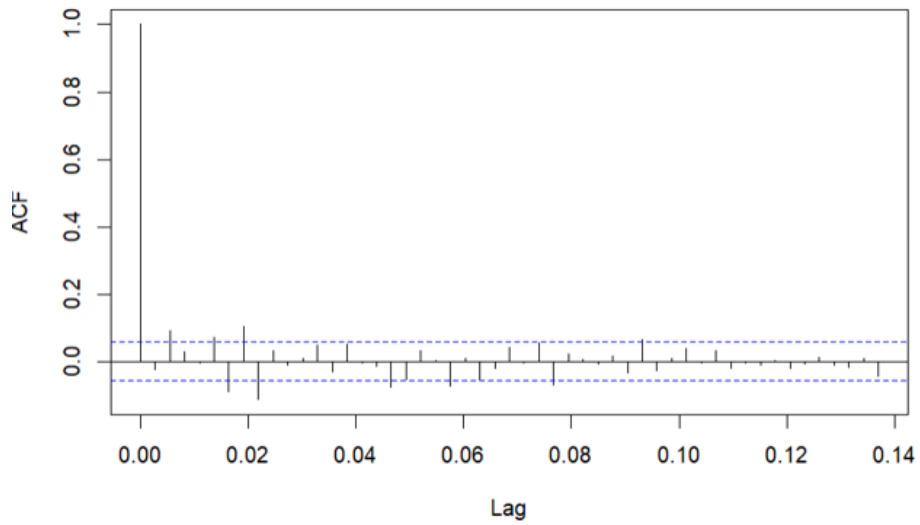
Şekil 4.16 ARIMA-E-GARCH-SVM DAX Logaritmik Dönüştürülmüş Veri Tahmini

4.3.3 NISEI

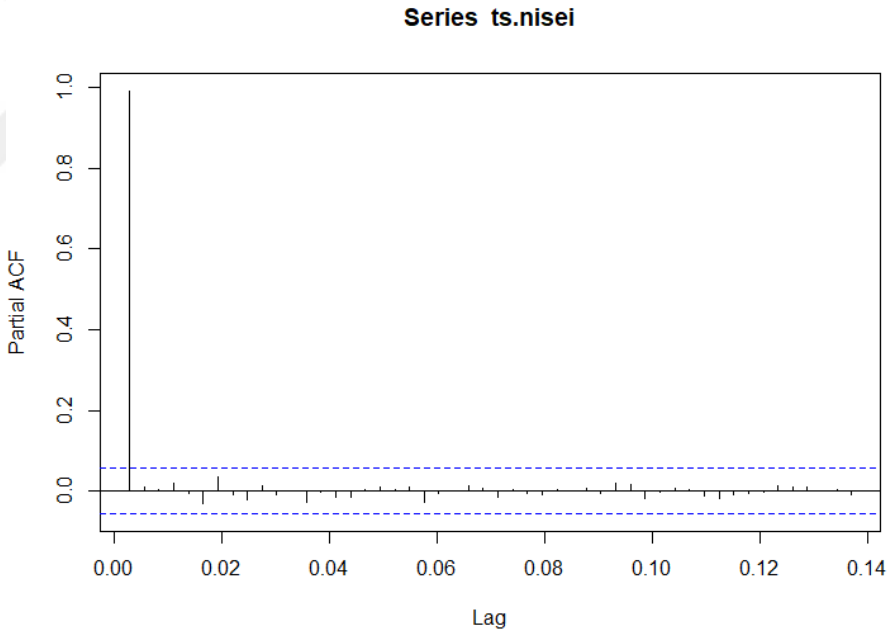
Veri seti olarak NISEI borsa endeksinin günlük kapanış değerleri kullanılmıştır. 1302 adet verinin %70'i eğitim seti, %30'u test seti olarak ayrılmıştır. Şekil 4.17'de verinin durağan olmadığı trend etkisi altında olduğu görülmektedir. ACF ve PACF grafiklerine de bakıldığında serinin durağan olmadığına karar verilmiştir. Seriyi durağanlaştırmak için logaritma dönüşümü ve fark alma işlemleri uygulanmıştır.



Şekil 4.17 NISEI Veri Dağılımı

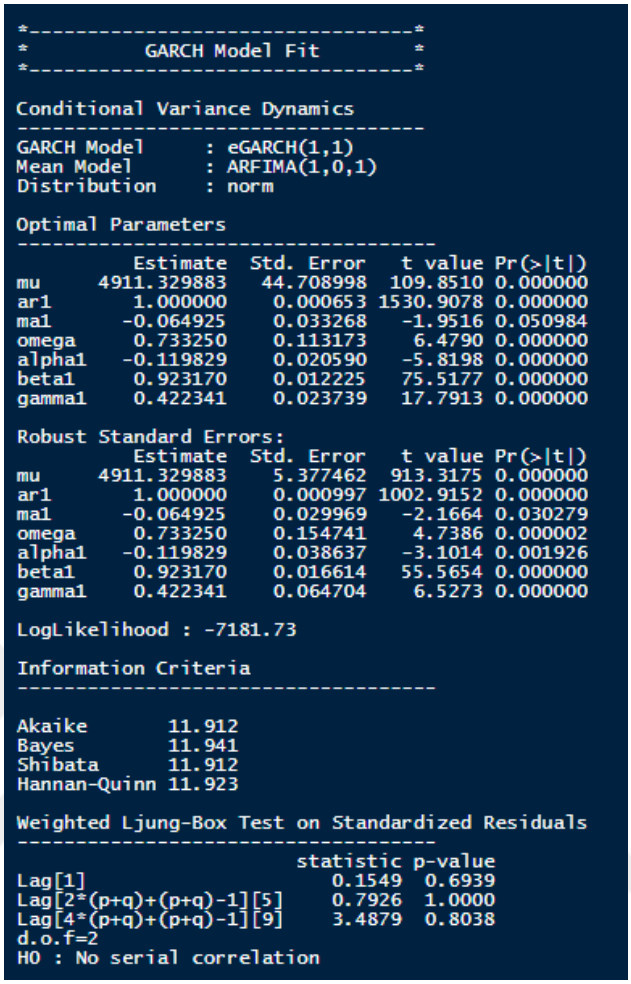


Şekil 4.18 NISEI ACF Grafiği



Şekil 4.19 NISEI PACF Grafiği

Şekil 4.18 ve Şekil 4.19'da belirtilen otokorelasyon ve kısmi otokorelasyon grafiklerine bakıldığında zaman serisinin Box-Jenkins metodolojisine göre durağanlaştığı ARIMA(1, 1), ARIMA(5, 1) ve ARIMA(6, 1) modelleri tanısal teşhis ile uygunluğu sınanabilir.



```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : eGARCH(1,1)
Mean Model       : ARFIMA(1,0,1)
Distribution     : norm

Optimal Parameters
------------------------------------
          Estimate  Std. Error    t value Pr(>|t|)
mu     4911.329883   44.708998  109.8510 0.000000
ar1       1.000000    0.000653 1530.9078 0.000000
ma1      -0.064925    0.033268   -1.9516 0.050984
omega     0.733250    0.113173    6.4790 0.000000
alpha1   -0.119829    0.020590   -5.8198 0.000000
beta1     0.923170    0.012225   75.5177 0.000000
gamma1    0.422341    0.023739   17.7913 0.000000

Robust Standard Errors:
          Estimate  Std. Error    t value Pr(>|t|)
mu     4911.329883    5.377462  913.3175 0.000000
ar1       1.000000    0.000997 1002.9152 0.000000
ma1      -0.064925    0.029969   -2.1664 0.030279
omega     0.733250    0.154741    4.7386 0.000002
alpha1   -0.119829    0.038637   -3.1014 0.001926
beta1     0.923170    0.016614   55.5654 0.000000
gamma1    0.422341    0.064704    6.5273 0.000000

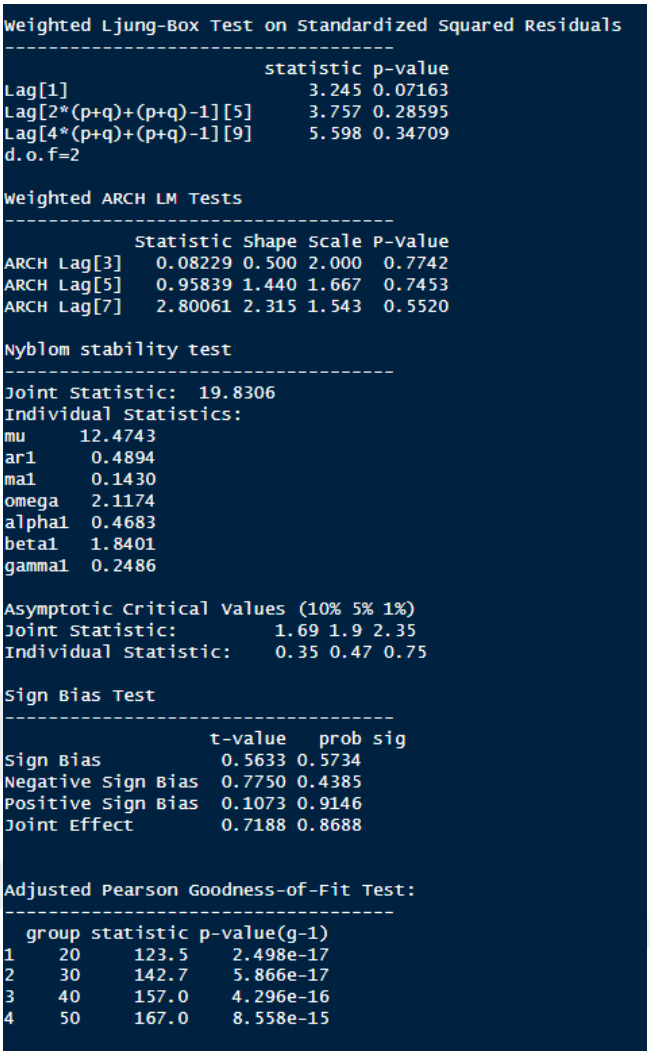
LogLikelihood : -7181.73

Information Criteria
------------------------------------

Akaike       11.912
Bayes        11.941
Shibata      11.912
Hannan-Quinn 11.923

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1549  0.6939
Lag[2*(p+q)+(p+q)-1][5]    0.7926  1.0000
Lag[4*(p+q)+(p+q)-1][9]    3.4879  0.8038
d.o.f=2
H0 : No serial correlation
```

Şekil 4.20 NISEI GARCH Test

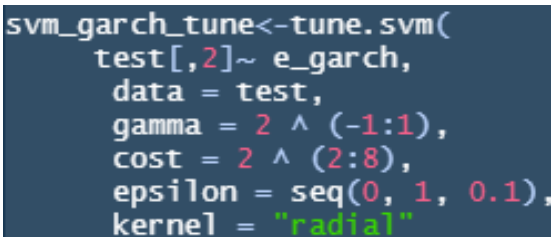


Şekil 4.21 NISEI GARCH Test 2

GARCH modeli sonuçlarına bakıldığında modele GARCH(1, 1) uygulanması kararı verilmiştir.

Kurulan modelin yeterli olup olmadığına karar vermek için artık kareleri üzerinde ağırlıklandırılmış Ljung-Box testine ve ARCH Lagrange Multiplier (ARCH-LM) testine bakılabilir. Bu durumda $\alpha = 0{,}05$ anlamlılık düzeyinde hata kareleri arasında otokorelasyon olmadığı kararı verilmiştir ve seçilen modelin uygun olduğu kararı verilebilir. Bu sebeple GARCH(1, 1) uygulanması uygundur.

Doğrusal yapılar modellendikten sonra doğrusal olmayan yapının modellemesi için SVM ve LSTM kullanılmıştır. LSTM modelinde epoch sayısı 25, batch_size 128, shuffle özelligi false olacak şekilde çalıştırılmıştır. SVM modelinin özellikleri Şekil 4.22'de gösterilmiştir.

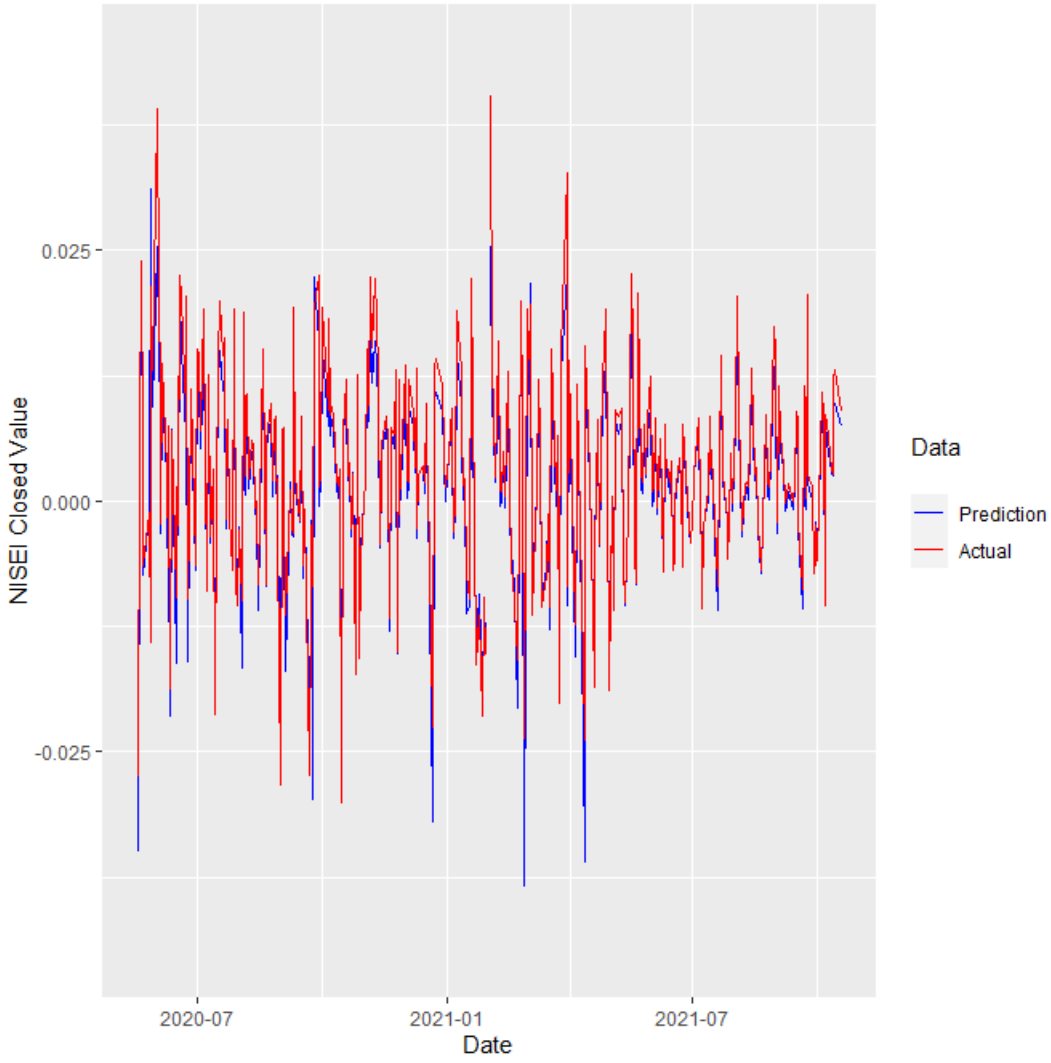


Şekil 4.22 NISEI SVM Model Özellikleri

Tablo 4.4'te NISEI veri setine ait modellerin hata değerleri gösterilmiştir. Bu hata değerleri arasından en küçük değerleri ARIMA-SVM (MAE= 0,00206) ve ARIMA-E-GARCH-SVM (MAE= 0,00222) modellerinin test verisine göre yapılan öngörü performans grafikleri Şekil 4.23 ve Şekil 4.24'te gösterilmiştir. Grafiklere bakıldığında da hata değerlerindeki yakınlığı gösterdiği ancak ARIMA-SVM modelinin tahminleme konusunda daha başarılı olduğu gözlenmiştir.

Tablo 4.4 NISEI Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri

Model	MAE	MAPE	RMSE
LSTM	0,01036	4,45507	0,01420
ARIMA-E-GARCH	0,00765	24,83050	0,01027
ARIMA	0,00776	1,23036	0,01035
SVM	0,00713	1,59433	0,01060
ARIMA_SVM	0,00206	0,48953	0,00401
ARIMA-E-GARCH_SVM	0,00222	0,52988	0,00413
ARIMA-E-GARCH_LSTM	0,00765	1,24970	0,01025
ARIMA_LSTM	0,00988	15,10841	0,01366



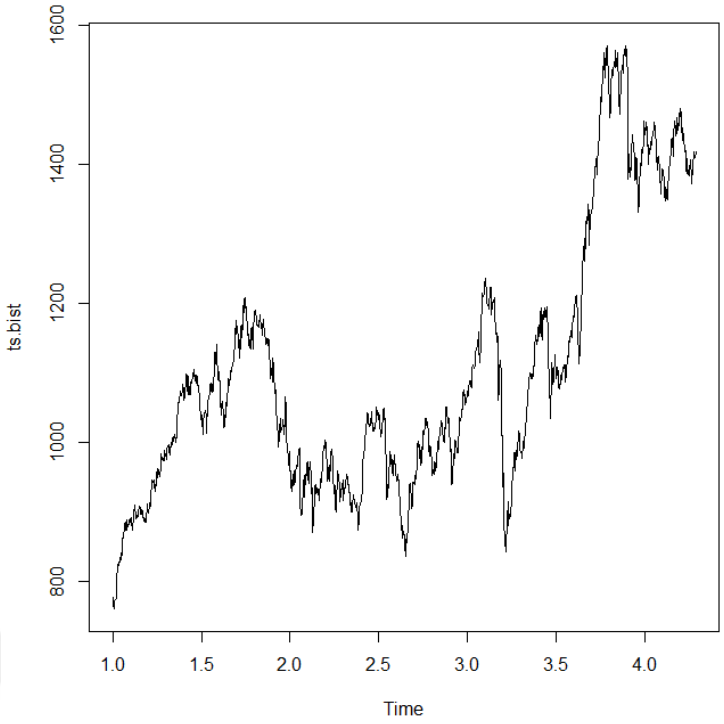
Şekil 4.23 ARIMA-SVM NISEI Logaritmik Dönüştürülmüş Veri Tahmini



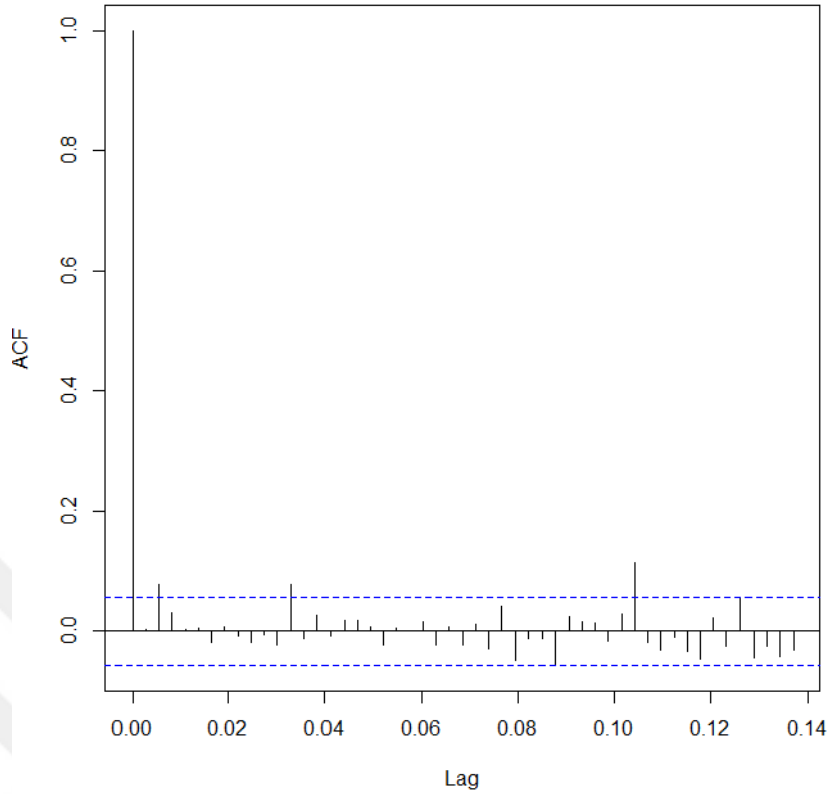
Şekil 4.24 ARIMA-E-GARCH-SVM NISEI Logaritmik Dönüştürülmüş Veri Tahmini

4.3.4 BIST100

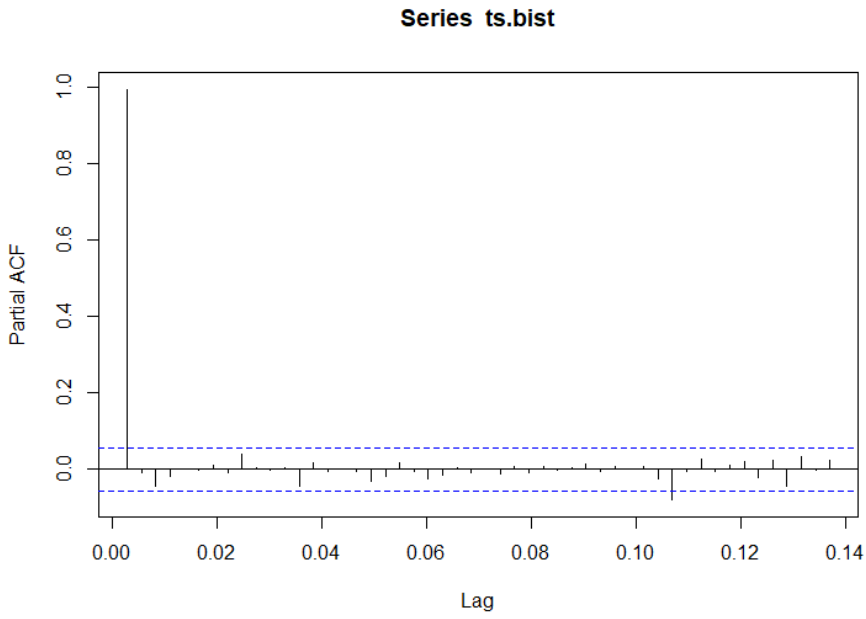
Veri seti olarak BIST100 borsa endeksinin günlük kapanış değerleri kullanılmıştır. 1305 adet verinin %70'i eğitim seti, %30'u test seti olarak ayrılmıştır. Şekil 4.25'te verinin durağan olmadığı trend etkisi altında olduğu görülmektedir. ACF ve PACF grafiklerine de bakıldığında serinin durağan olmadığına karar verilmiştir. Seriyi durağanlaştırmak için logaritma dönüşümü ve fark alma işlemleri uygulanmıştır.



Şekil 4.25 BIST100 Veri Dağılımı



Şekil 4.26 BIST100 ACF Grafiği



Şekil 4.27 BIST100 PACF Grafiği

Şekil 4.26 ve Şekil 4.27'de belirtilen otokorelasyon ve kısmi otokorelasyon grafiklerine bakıldığında zaman serisinin Box-Jenkins metodolojisine göre durağanlaştığı ARIMA(1, 1), ARIMA(5, 1) ve ARIMA(6, 1) modelleri diagnostics checking(tanısal teşhis) ile uygunluğu sınanabilir.

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,1)
Mean Model      : ARFIMA(1,0,1)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000209    0.000313     0.66628 0.505233
ar1     0.851333    0.054102    15.73583 0.000000
ma1    -0.801550    0.062553   -12.81385 0.000000
omega  -0.595401    0.005713  -104.21560 0.000000
alpha1 -0.169223    0.019436    -8.70686 0.000000
beta1   0.931126    0.000059 15867.33095 0.000000
gamma1  0.103308    0.021789     4.74120 0.000002

Robust Standard Errors:
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000209    0.000391     0.53373 0.593530
ar1     0.851333    0.037591    22.64721 0.000000
ma1    -0.801550    0.041692   -19.22547 0.000000
omega  -0.595401    0.007568   -78.67140 0.000000
alpha1 -0.169223    0.026761    -6.32359 0.000000
beta1   0.931126    0.000076 12235.34889 0.000000
gamma1  0.103308    0.026804     3.85423 0.000116

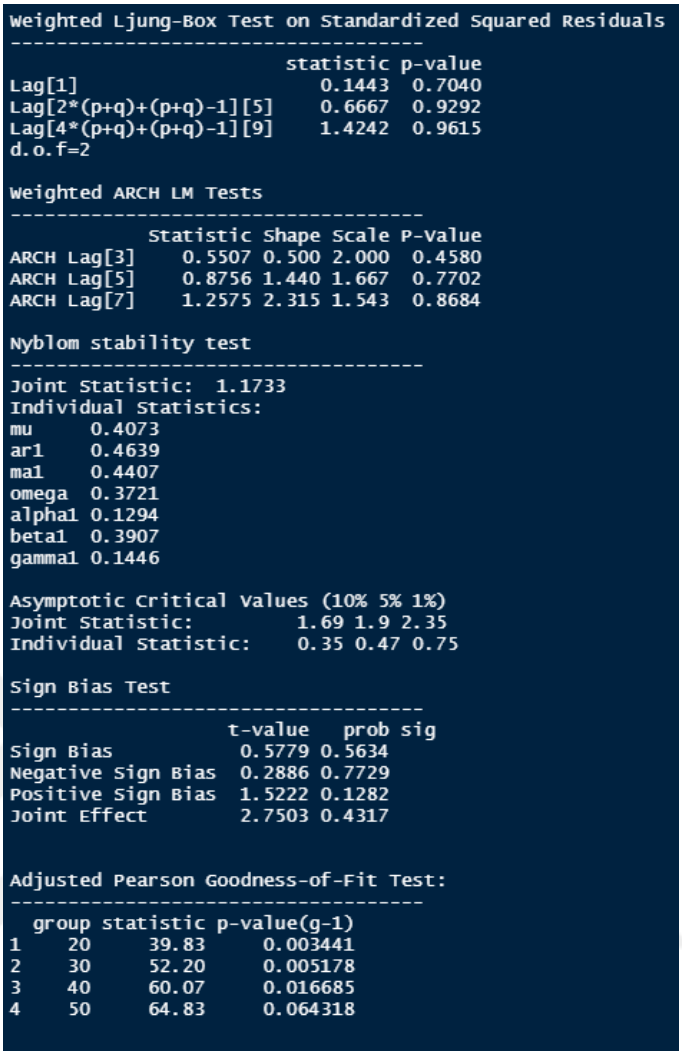
LogLikelihood : 3537.098

Information Criteria
------------------------------------

Akaike       -5.8835
Bayes        -5.8538
Shibata      -5.8836
Hannan-Quinn -5.8723

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      1.192  0.2748
Lag[2*(p+q)+(p+q)-1][5]     1.582  0.9962
Lag[4*(p+q)+(p+q)-1][9]     2.683  0.9328
d.o.f=2
H0 : No serial correlation
```

Şekil 4.28 BIST100 GARCH Test

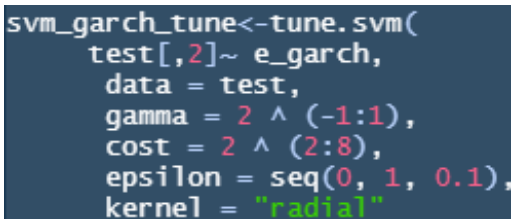


Şekil 4.29 BIST100 GARCH Test 2

GARCH modeli sonuçlarına bakıldığında modele GARCH(1, 1) uygulanması kararı verilmiştir.

Kurulan modelin yeterli olup olmadığına karar vermek için artık kareleri üzerinde ağırlıklanıdırılmış Ljung-Box testine ve ARCH Lagrange Multiplier (ARCH-LM) testine bakılabilir. Bu durumda $\alpha = 0{,}05$ anlamlılık düzeyinde hata kareleri arasında otokorelasyon olmadığı kararı verilmiştir ve seçilen modelin uygun olduğu kararı verilebilir. Bu sebeple GARCH(1, 1) uygulanması uygundur.

Doğrusal yapılar modellendikten sonra doğrusal olmayan yapının modellemesi için SVM ve LSTM kullanılmıştır. LSTM modelinde epoch sayısı 25, batch_size 128, shuffle özelligi false olacak şekilde çalıştırılmıştır. SVM modelinin özellikleri Şekil 4.31'de gösterilmiştir.

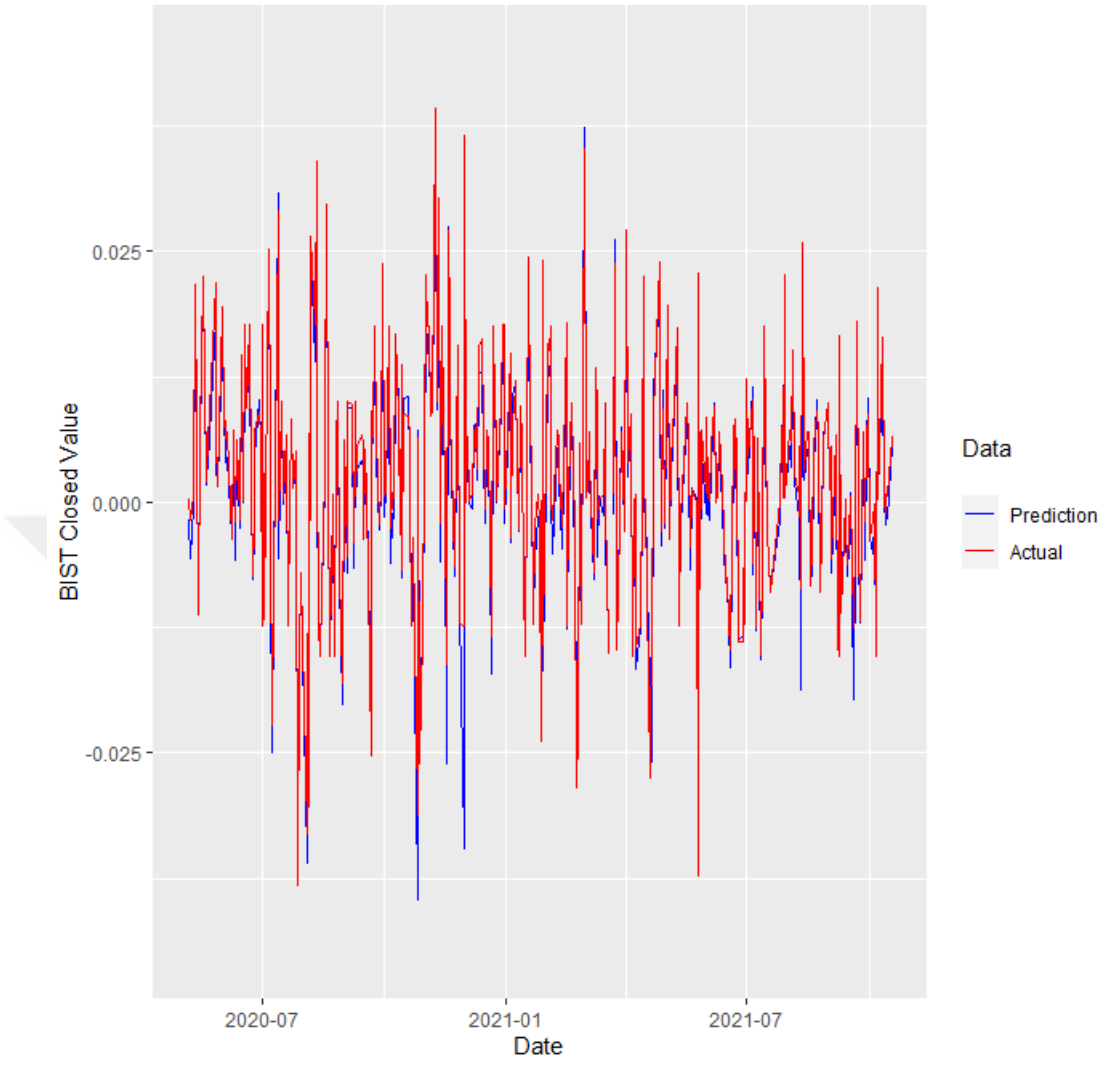


Şekil 4.30 BIST100 SVM Model Özellikleri

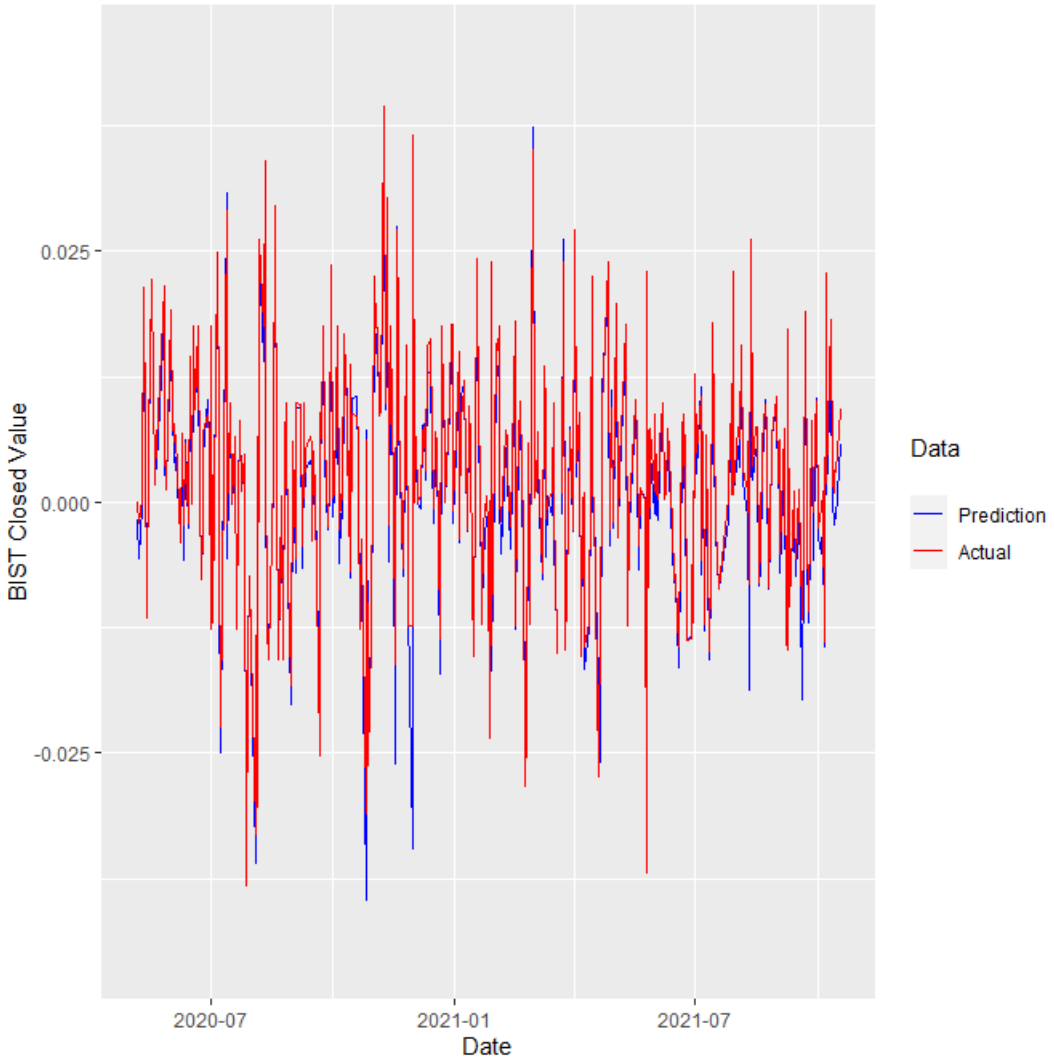
Tablo 4.5'te BIST100 veri setine ait modellerin hata değerleri gösterilmiştir. Bu hata değerleri arasından en küçük değerleri ARIMA-SVM (MAE= 0,00183) ve ARIMA-E-GARCH-SVM (MAE= 0,00191) modellerinin test verisine göre yapılan öngörü performans grafikleri Şekil 4.31 ve Şekil 4.32'de gösterilmiştir. Grafiklere bakıldığında da hata değerlerindeki yakınlığı gösterdiği ancak ARIMA-SVM modelinin tahminleme konusunda daha başarılı olduğu gözlenmiştir.

Tablo 4.5 Bist100 Logaritmik Dönüştürülmüş Veriler İçin Tahmin Doğruluğu Ölçüleri

Model	MAE	MAPE	RMSE
LSTM	0,01331	Inf	0,01866
ARIMA-E-GARCH	0,00930	99,03596	0,01339
ARIMA	0,00938	NA	0,01341
SVM	0,00930	Inf	0,01404
ARIMA_SVM	0,00183	0,82378	0,00302
ARIMA-E-GARCH_SVM	0,00191	0,93336	0,00310
ARIMA-E-GARCH_LSTM	0,00952	1,51085	0.01349
ARIMA_LSTM	0,01353	5,82403	0,01893



Şekil 4.31 ARIMA-SVM BIST100 Logaritmik Dönüştürülmüş Veri Tahmini



Şekil 4.32 ARIMA-E-GARCH-SVM BIST100 Logaritmik Dönüştürülmüş Veri Tahmini

BÖLÜM BEŞ
DEĞERLENDİRME

Bu tez kapsamında hisse senedi verileri üzerinde çeşitli zaman serisi algoritmaları ve analiz yöntemleri uygulanmıştır. İlk olarak çoğu finansal zaman serisinde de karşılaşılan durağanlık problemi, fark alma ve logaritmik dönüşüm yöntemleri ile çözülmeye çalışılmıştır. Bu veri düzenlemesinden sonra ilk olarak geleneksel yöntemlerden ARIMA uygulanmıştır. Ancak ARIMA yöntemi bu veri setlerindeki oynaklık ve karmaşıklığı açıklamaya yeterli olmamıştır. Sonraki aşamada ARIMA-GARCH, SVM ve LSTM yöntemleri de sırasıyla uygulanmıştır. Ancak bu yöntemlerden hiçbiri veriyi istenilen seviyede doğru tahmin edememiştir. Bu sebeple hibrit yöntem uygulanarak tahminleme yapılması kararı verilmiştir. İlk olarak ARIMA ve ARIMA-GARCH yöntemleri kullanılmıştır. Bu süreçte serideki oynaklığı modellemek için EGARCH(1,1) yönteminin tüm veri setleri için uygun olduğuna karar verilmiştir. Bu iki doğrusal model sonucunda oluşan hatalar, doğrusal olmayan yöntemlere girdi olarak verilmiştir. SVM ve LSTM için çeşitli model iyileştirmeleri denenmiş olup en uygun olan ayarlar ile hatalar tahminlenmiştir. Sonraki aşamada doğrusal olmayan yöntemler ile tahminlenen hatalar ile doğrusal yöntem sonucunda çıkan değerler toplanarak optimum sonuçlara ulaşılmıştır. Bu sıralama ile bütün veri setleri için tüm yöntemler ikili kombinasyon şeklinde denenmiştir. Model tahminleri ile gerçek değerler arasındaki hatalar MAE, MAPE ve RMSE metrikleri ile kıyaslanmıştır. NASDAQ veri seti için ARIMA-E-GARCH-SVM modeli, DAX veri seti için ARIMA-SVM modeli, NISEI veri seti için ARIMA-SVM modeli ve BIST100 veri seti için ARIMA-SVM modeli daha başarılı olmuştur.

KAYNAKLAR

Adebiyi, A.A., Adewumi, A.O. ve Ayo, C.K. (2014). Stock price prediction using the ARIMA model. Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014.

Adebiyi, A.A., Adewumi, A.O.; Ayo, C.K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. J. Appl. Math. 2014, 2014, 614342.

Ahmed, I., Nawaz, M., Bowra, Z., Zafar, A., Muhammad, Z., Shaukat, A., Rehman, W. ve Ahmed, N. (2010). Does service quality affect students' performance? Evidence from institutes of higher learning. African Journal of Business Management, 4, 2527-2533.

Ake, B. ve Dehuan, J. (2010). The Role of Stock Market Development in Economic Growth: Evidence from Some Euronext Countries. International Journal of Financial Research, 1.

Ben Baccar, Y. (2019). Comparative Study on Time Series Forecasting Models. (https://doi.org/10.13140/RG.2.2.32241.02408)

Bollerslev T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics (Volume 31), Issue 3, Nisan 1986, s. 307-327.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M. (2015). Time Series Analysis: Forecasting and Control, John Wiley and Sons.

Brownlee, J. (2017). How to Create an ARIMA Model for Time Series Forecasting in Python. (https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/)

Cao, X., Zhao, Z. (2022). Research on stock index forecasting based on ARIMA-GARCH and SVM mixed model.

Chen, L. (2020). Stock Price Prediction using Adaptive Time Series Forecasting and Machine Learning Algorithms [UCLA].

Chen K., Zhou Y. ve Dai F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market.

Chiang, W.C., Urban, T.L., Baildridge, G. (1996). A neural network approach to mutual fund net asset value forecasting, Omega 24 (2), 205–215.

Cryer, J. D. ve Chan, K. (2008). Time series analysis: With applications in R (2. Baskı).

De Mattos Neto, P.S., Cavalcanti, G.D., Madeiro, F. (2017). Nonlinear combination method of forecasters applied to PM time series, Pattern Recognit. Lett. 95, 65–72.

Dhamija, A. ve Bhalla, V. (2010). Financial Time Series Forecasting: Comparison of various ARCH models. Global Jounal of Finance and Management, 2, 159-172.

Domingos S., João O. ve Paulo M.N. (2019). An intelligent hybridization of ARIMA with machine learning models for time series forecasting.

Engle R.F., Kane A., Chou R. (1992). Measuring risk aversion from excess returns on a stock index, Journal of Econometrics, 52, 201-224. (http://dx.doi.org/10.1016/0304-4076(92)90070-8)

GeeksforGeeks. (2021). Support Vector Machine Algorithm. (https://www.geeksforgeeks.org/support-vector-machine-algorithm/)

Hyndman R.J., Athanasopoulos G. (2018). Forecasting: Principles and Practice (2. Baskı). (https://otexts.com/fpp2/index.html)

Jose, J. (2022). Introduction to Time Series Analysis and Its Applications.

Khashei, M., Bijari, M. (2010). An artificial neural network (p, d, q) model for time series forecasting, Expert Syst. Appl. 37 (1), 479–489.

Khashei, M., Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, Appl. Soft Comput. 11 (2), 2664–2675.

Kim, H. Y. ve 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 K.J. (2003). Financial time series forecasting using support vector machines. Neurocomputing (Volume 55), Issues 1–2, Eylül 2003, s. 307-319.

Maknickiene N. ve Maknickas A.A. (2012). Application of Neural Network for Forecasting of Exchange Rates and Forex Trading, Conference: The 7th international scientific conference "Business and Management 2012"At: Vilnius.

Li, A.W., Bastos, G.S. (2020) Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review. IEEE Access 2020, 8, 185232–185242.

Muller, K.R., Smola, J.A., Scholkopf, B. (1997). Prediction time series with support vector machines, Proceedings of International Conference on Artificial Neural Networks, Lausanne, Switzerland, (999–1004).

Nabi, J. (2019). Recurrent Neural Networks (RNNs). (https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85)

Nelson, D.B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach, Econometrica, Econometric Society, vol. 59(2), s. 347-370, Mart.

Nelson, D.B. ve Cao C.Q. (1992). Inequality Constraints in the Univariate GARCH Model, Journal of Business ve Economic Statistics Vol. 10, No. 2 (Nisan, 1992), (229-235).

Okasha, M. (2014). Using Support Vector Machines in Financial Time Series Forecasting. International Journal of Statistics and Applications, 4, 28-39.

Palit, A. K. ve Popovic, D. (2006). Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications. (https://doi.org/10.1007/1-84628-184-9)

Panigrahi, S., Behera, H. (2017). A hybrid ETS–ANN model for time series forecasting, Eng. Appl. Artif. Intell. 66, 49–59.

Rao, S. (2008). A course in Time Series Analysis. American Statistician - AMER STATIST, 56.

Singhal G. (2020). Introduction to LSTM Units in RNN. (https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn)

Taghiyeh S, Lengacher D.C. ve Handfield R.B. (2020). Forecasting model selection using intermediate classification: Application to MonarchFx Corporation.

SAS Help Center: GARCH Models. (2018). (https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/etsug/etsug_autoreg_details12.htm)

Sezer, O. B., Gudelek, M. U. ve Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.

Siami-Namini, S., Tavakoli, N. ve Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. 2018 17th IEEE International Conferen-ce on Machine Learning and Applications (ICMLA), 1394-1401.

Scholkopf, B., Burges, C., Vapnik, V. (1995). Extracting support data for a given task, in: U.M. Fayyad, R. Uthurusamy (Eds.), Proceedings of the First International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA.

Syrris I. Ve Shenai V. (2021). Forecasting Gold Prices with ARIMA and GARCH Models.

Vapnik, V.N. (1995). The Nature of Statistical Learning Theory, Springer, New York.

Vo, N., Ślepaczuk R. (2022). Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index. Working Papers 2021-25, Faculty of Economic Sciences, University of Warsaw.

Yaser, S.A.M., Atiya, A.F. (1996). Introduction to financial forecasting, Appl. In-tell. 6, 205–213.

Zhang, G., Eddy Patuwo, B. ve Y. Hu, M. (1998). Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14(1), 35-62.

Zhang, P. (2003). Zhang, G.P.: Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. Neurocomputing 50, 159-175.