T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İŞLETME ANA BİLİM DALI

FİNANSAL ZAMAN SERİSİ TAHMİNİNDE AÇIKLANABİLİR YAPAY ZEKA İLE MAKİNE ÖĞRENİMİ TAHMİN PERFORMANSININ GELİŞTİRİLMESİ

Doktora Tezi

Taha Buğra ÇELİK

Danışman
Doç. Dr. Elif BULUT

II. Danışman
Dr. Öğr. Üyesi Özgür İCAN

SAMSUN
2023

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İŞLETME ANA BİLİM DALI

FİNANSAL ZAMAN SERİSİ TAHMİNİNDE AÇIKLANABİLİR YAPAY ZEKA İLE MAKİNE ÖĞRENİMİ TAHMİN PERFORMANSININ GELİŞTİRİLMESİ

Doktora Tezi

Taha Buğra ÇELİK

Danışman

Doç. Dr. Elif BULUT

II. Danışman

Dr. Öğr. Üyesi Özgür İCAN

SAMSUN
2023

TEZ KABUL VE ONAYI

Taha Buğra ÇELİK tarafından, **Doç. Dr. Elif BULUT** danışmanlığında hazırlanan **"FİNANSAL ZAMAN SERİSİ TAHMİNİNDE AÇIKLANABİLİR YAPAY ZEKA İLE MAKİNE ÖĞRENİMİ TAHMİN PERFORMANSININ GELİŞTİRİLMESİ"** başlıklı bu çalışma, jürimiz tarafından 15.10.2023 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Doktora Tezi olarak kabul edilmiştir.

	Unvanı Adı Soyadı **Üniversitesi** **Ana Bilim/Ana Sanat Dalı**	**Sonuç**
Başkan	Prof. Dr. Vedide Rezan USLU Ondokuz Mayıs Üniversitesi İstatistik Teorisi Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Prof. Dr. Erol EĞRİOĞLU Giresun Üniversitesi Uygulamalı İstatistik Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Prof. Dr. Kazım Barış ATICI Hacettepe Üniversitesi Sayısal Yöntemler Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Doç. Dr. Elif BULUT Ondokuz Mayıs Üniversitesi Sayısal Yöntemler Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Doç. Dr. Durmuş YILDIRIM Ondokuz Mayıs Üniversitesi Muhasebe ve Finansman Ana Bilim Dalı	☒ Kabul ☐ Ret

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

Prof. Dr. Ahmet TABAK
Enstitü Müdürü

BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Doktora tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet ☐ (Gerekli ise ekler kısmına ekleyiniz)

Hayır ☒

14 /11 / 2023
Taha Buğra ÇELİK

TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

Tez Başlığı : FİNANSAL ZAMAN SERİSİ TAHMİNİNDE AÇIKLANABİLİR YAPAY ZEKA İLE MAKİNE ÖĞRENİMİ TAHMİN PERFORMANSININ GELİŞTİRİLMESİ

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 12/10/2023 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 13

Tek kaynak oranı : % 5 çıkmıştır.

12 /10 / 2023
Doç. Dr. Elif BULUT

ÖZET

FİNANSAL ZAMAN SERİSİ TAHMİNİNDE AÇIKLANABİLİR YAPAY ZEKA İLE MAKİNE ÖĞRENİMİ TAHMİN PERFORMANSININ GELİŞTİRİLMESİ

Taha Buğra ÇELİK
Ondokuz Mayıs Üniversitesi
Lisansüstü Eğitim Enstitüsü
İŞLETME ANA BİLİM DALI
Doktora, Kasım/2023
Danışman: Doç. Dr. Elif BULUT

Son zamanlarda, makine öğrenimi tekniklerini kullanarak borsa fiyat yönünü başarılı bir şekilde tahmin etmede önemli miktarda çaba sarf edilmektedir. Makine öğrenmesi teknikleri ve modelleri, yapısı gereği literatürde kara kutu olarak adlandırılırlar, yani modelin yapmış olduğu tahmin ve çıkarımların nasıl yapıldığına dair çıkarsama yapılamaz. Bu duruma dair çözüm önerileri "açıklanabilir yapay zeka" çatısı altında geliştirilmektedir. Bu çalışmada daha başarılı makine öğrenmesi teknikleri geliştirmek yerine mevcut modellerin tahminlerinin güvenilirliğini değerlendirmek için kullanılabilecek ve böylece karar vericinin genel tahmin performansının düşmesinden sorumlu olan kötü kararlardan kaçınmasına olanak tanıyan bir açıklanabilir yapay zeka yaklaşımı önerilmektedir. Çünkü tahmin modelinin herhangi bir tahminden ne kadar emin olduğunun bir ölçüsü olsaydı, nispeten daha yüksek güvenilirliğe sahip tahminler karar vermek için kullanılabilirken, daha düşük kaliteli kararlardan kaçınılabilmesine olanak sağlanabilir. Bu çalışmada, makine öğrenimi, açıklanabilir yapay zeka ve deneysel mod ayrışımına dayalı borsa yönü tahmini için yeni bir iki aşamalı tahmin modeli önerilmiştir. Çalışmanın bulgularına göre tahmin modelinin literatürdeki mevcut tahmin modellerinden daha başarılı sonuçlar ürettiği gösterilmektedir.

Anahtar Sözcükler: Makine Öğrenimi, Finansal Zaman Serisi Tahmini, Derin Öğrenme, Açıklanabilir Yapay Zeka, Deneysel Mod Ayrıştırma.

ABSTRACT

IMPROVING MACHINE LEARNING PREDICTION PERFORMANCE WITH EXPLAINABLE ARTIFICIAL INTELLIGENCE IN FINANCIAL TIME SERIES PREDICTION

Taha Buğra ÇELİK
Ondokuz Mayıs University
Institute of Graduate Studies
Department of Business Administration
Ph.D., November/2023
Supervisor: Assoc. Prof. Dr. Elif BULUT

Recently, significant efforts have been made to successfully predict the direction of stock prices using machine learning techniques. Machine learning techniques and models are inherently referred to as black boxes in the literature, meaning that no inference can be made about how the model makes its predictions and inferences. Solution proposals for this issue are being developed under the umbrella of "explainable artificial intelligence". In this study, we suggest using an explainable AI approach that can be used to evaluate the reliability of the predictions of existing models instead of developing more successful machine learning techniques, thereby allowing decision-makers to avoid bad decisions that are responsible for the overall prediction performance decrease. Because if there was a measure of how certain the prediction model is about any given prediction, relatively more reliable predictions could be used for decision-making, and lower-quality decisions could be avoided. In this study, a new two-stage prediction model based on machine learning, explainable artificial intelligence and empirical mode decomposition is proposed for predicting the direction of the stock market. According to the findings of the study, the prediction model produces more successful results than existing prediction models in the literature.

Keywords: Machine Learning, Financial Time Series Prediction, Deep Learning, Explainable Artificial Intelligence, Empirical Mode Decomposition.

ÖN SÖZ VE TEŞEKKÜR

Öncelikle bu çalışmanın konusu ve akademik gelişimim süresince beni en doğru şekilde yönlendirebilmek için gayret ve özverilerini esirgemeyen, bilgi ve tecrübelerinden her daim yararlandığım kıymetli hocam Dr. Öğr. Üyesi Özgür İCAN'a, tez çalışmalarımda ve akademik hayatımda her zaman yanımda olan saygıdeğer hocam Doç. Dr. Elif BULUT'a ve çalışmalarım esnasında yeterince zaman ayıramadığım sevgili eşim Merve CELİK'e teşekkürlerimi sunarım.

Taha Buğra ÇELİK

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

1. GİRİŞ

Yatırım teorileri ve stratejileri ile ilgili olarak çok sayıda teori ve yaklaşım önerilmiş ve günümüze kadar olan süreçte bu alanda en büyük katkılardan birini Markowitz (1952), portföy seçimi üzerine yayınlamış olduğu çalışma ile öne sürmüştür. Ancak modern portföy teorisinin temelde yalnızca risk ve getiri arasındaki dengeyi kuramsallaştırmanın haricinde yatırım kararlarının belki de en önemli ayağından birisi olan gelecekte oluşabilecek fiyat düzeyleri ile ilgili ön görüler doğrultusunda belirli bir yatırım portföyü oluşturulmasına dair herhangi bir önermede bulunmamaktadır.

Finansal zaman serisi tahmini için sayısız yöntem ve istatistiksel teknik kullanılarak fiyat tahminleri yapmak suretiyle yatırım stratejileri önerilmiştir. Ancak bu tür çalışmalar, uygulanan yatırım veya portföy oluşturma teorisine ilişkin genel bir teorik çerçeve çizmekten ziyade doğrudan finansal zaman serisi tahmini uygulaması üzerinden yatırım kararlarının başarısını test etmeye dönük olmaktadır. Finansal piyasalar, özellikle hisse senedi piyasaları, yatırımcıların ve kısa vadeli fiyat hareketlerinden getiri elde etmeyi amaçlayanların doğru kararlar vererek sermaye kazancı elde etmelerini sağlar. Ancak hisse senedi piyasalarındaki fiyat hareketleri doğrusal değildir ve sürekli bir biçimde doğru kararlar vermek zordur.

Makine öğrenimi (machine learning (ML)) ve derin öğrenmede son yıllarda yaşanan hızlı gelişmeler sonucunda borsa tahmini alanında büyük ilerleme kaydedilmiştir. Bu açıdan son yıllarda bu alanda yapılan güncel çalışmaları genel bir çerçevede incelemekte yarar vardır.

1.1. İlgili Çalışmalar

Birçok yeni yöntem ve teknik bir arada geliştirilip uygulandığından, hisse senedi piyasası tahmin literatüründe araştırmaları sınıflandırmak çok zordur. Bu nedenle, kesin bir sınıflandırma yapmak yerine, ilgili araştırmalar, tahmin başarısını doğrudan artırmak için önerdikleri yöntem veya tekniklere göre gruplandırılabilir.

Literatürde tahmin başarısını artırmak için modelin kullandığı öznitelik uzayını belirleme/optimize etmeye odaklanan çalışmalar yaygındır. Zhou vd. (2020), destek vektör makinesi (SVM) ile borsa yönünü tahmin etmek için geçmiş işlem verileri, teknik göstergeler, haberler ve Baidu endeksi dahil olmak üzere birden fazla veri kaynağı kullanmıştır. Öznitelik odaklı çalışmalarda, ML veya derin öğrenme

yöntemleri ile melezlenerek yeni tahmin modelleri sıklıkla kullanılmaktadır. Bu çalışmalar, tahmin modelinin başarısını artırmak için özniteliklerin iyileştirilmesine odaklanmaktadır.

Topolojik veri analizinde kullanılan bir yöntem olan *kalıcı benzeşiklik* (Persistent homology) borsa tahmin modelinde faydalı girdiler elde etmek için kullanılmaktadır (Ismail vd., 2020). Öznitelik seçimi için genetik algoritmalar (GA) da başvurulan yöntemler arasındadır. Yun vd. (2021), öznitelik seçimi için genetik algoritmalar (GA) kullanan melez bir GA-XGBoost tahmin modeli önermiştir.

Girdilere odaklanarak modelin tahmin başarısını artırma çabası olan bu yaklaşımlara literatürde öznitelik mühendisliği denilmektedir (Shen ve Shafiq, 2020). Hisse senedi fiyatlarından gizli öznitelikleri öğrenmek için Zhang vd. (2021), derin inanç ağlarını (deep belief networks (DBN)) önermektedirler. Thakkar ve Chaudhari (2020), tarihsel borsa verilerinden ve geri yayılım sinir ağlarından (backpropagation neural network (BPNN)) öznitelik ağırlık matrisi türetmek için terim frekansı-ters belge frekansı (term frequency–inverse document frequency (TF–IDF)) kullanmışlardır. Hao ve Gao (2020), borsa endeksinin fiyat eğilimini tahmin etmek için çoklu zaman ölçekli öznitelik öğrenmeyi önermektedirler.

Farklı tarihsel fiyat ve metinsel veri kaynaklarından tamamlayıcı öznitelikleri öğrenmek için, Liu vd. (2020), tekrarlayan bir evrişimsel sinir çekirdeği (recurrent convolutional neural kernel) kullanmışlardır. Yang vd. (2020), öznitelik çıkarımı için evrişimli sinir ağını (convolutional neural network (CNN)) ve tahmin için uzun kısa süreli bellek (long short-term memory (LSTM)) ağını birleştirmektedirler.

Özniteliklere odaklanan bir diğer yaklaşımda ise, öznitelik kümesinin karmaşıklığını azaltarak tahmin modellerinin hesaplama verimliliğini artırmak için değişken otomatik kodlayıcılar (variational auto-encoders (VAE)) ve temel bileşenler analizi (principal components analysis (PCA)) gibi boyut azaltma yöntemlerine başvurulmaktadır (Ghorbani ve Chong, 2020; Gunduz, 2021).

Ayrıca teknik analiz (TA) göstergelerinin grup cezalandırmalı lojistik regresyon (group penalized logistic regressions), dikkat temelli BiLSTM (attention-based BiLSTM), yapay sinir ağı (artificial neural network (ANN)), destek vektör makinesi (support vector machine (SVM)), rastgele orman (random forest (RF)) ve naive Bayes (NB) gibi çeşitli derin öğrenme ve makine öğrenmesi yöntemlerinde girdi

olarak kullanılması literatürde yaygındır (Lee vd., 2022; Nabipour vd., 2020; Patel vd., 2015; Y. Yang vd., 2022).

Boyut azaltmanın yanı sıra, bir diğer önemli yaklaşım, karmaşık zaman serilerinin daha yönetilebilir alt bileşenlere ayrıştırılmasıdır. En yaygın olarak kullanılan ayrıştırma yaklaşımlarından biri deneysel mod ayrıştırmadır (empirical mode decomposition (EMD)) (Huang vd., 1998). Hisse senedi fiyatını ayrıştırmak için Xu ve Tan (2021), tarafından EMD uygulanmış ve alt bileşenler geçici dikkat LSTM (temporal attention LSTM) ile tahmin edilmiştir. Benzer şekilde hisse senedi piyasasının yönünü tahmin etmek amacıyla EMD çeşitli tahmin modelleri ile birlikte kullanılabilmektedir (Jin vd., 2020; F. Zhou vd., 2019). Ayrıca, EMD'ye alternatif olarak geliştirilmiş, varyasyonel mod ayrıştırma (variational mode decomposition (VMD)) adı verilen daha gelişmiş bir ayrıştırma yöntemi vardır (Dragomiretskiy ve Zosso, 2014). Hisse senedi piyasası tahmini için derin öğrenme ve makine öğrenimi modelleriyle birlikte VMD'nin kullanılmasının başarılı sonuçlar sağladığı kanıtlanmıştır (Bisoi vd., 2019; Niu vd., 2020; Yujun vd., 2021).

Tahmin modelleriyle birlikte kullanılan tekil spektrum analizi (singular spectrum analysis (SSA)), deneysel dalgacık dönüşümü (empirical wavelet transform (EWT)), toplu EMD (ensemble EMD (EEMD)) gibi başka ayrıştırma yöntemleri de bulunmaktadır (H. Liu ve Long, 2020; Xiao vd., 2019; Yujun vd., 2020).

Yukarıda bahsedilen çalışmalarda, tek bir tahmin modelinin tahmin performansını artırmak için ek yöntem ve tekniklerin melezlendiği yaygın olarak gözlemlenmiştir. Literatürde, çeşitli ortamlarda çoklu tahmin modellerinin kullanılmasına yönelik bariz bir eğilim vardır. Bu türden yaklaşımlar, daha iyi bir bileşik tahmin modeli üretmek için çoklu tahmin modellerini birleştiren bir meta yaklaşım olan topluluk öğrenmesi (ensemble learning) olarak adlandırılır.

Yüksek tahmin başarısı elde etmek için çoklu modellere başvurmak yaygın olarak kullanılan bir yaklaşımdır ve literatürde çoklu sınıflandırıcılar (multi-classifiers), çoklu sınıflandırıcılar kombinasyonu (multi-classifiers combination) ve uzman sistemler bileşeni (mixture of experts) olarak bilinir (daha fazla ayrıntı için bkz. (Mohandes vd., 2018)).

Topluluk yöntemleri, torbalama (bagging), istifleme (stacking) ve güçlendirici (boosting) olarak adlandırılan üç ana kategoriye ayrılmaktadır. Torbalama yöntemi

rastgele örnekleme kullanılarak oluşturulan farklı eğitim alt kümeleriyle eğitilmiş çeşitli tahmin modelleri grubunu içermektedir. Topluluk üyeleri tarafından yapılan tahminler daha sonra nihai bir tahmin değeri üretmek için bir kombinasyon şemasına (oylama, ortalama alma veya herhangi bir başka kural kümesi gibi) verilir. İstifleme yaklaşımında, daha yüksek tahmin doğruluğu elde etmek için bir grup tahmin modelinin çıktıları başka bir tahmin modelinde girdi olarak birleştirir (Dash vd., 2019). Son olarak, güçlendirici topluluk modellerinde, eğitilmiş bir modelin başka bir veri kümesinde yapmış olduğu yanlış sınıflandırma örneklerine odaklanmak için eğitim verileri yinelemeli olarak değiştirilir.

Literatürde borsa tahmin modellerinin başarısını artıracak pek çok yöntem ve teknik bulunmakta ve sürekli olarak yenileri geliştirilmektedir. Bahsedilen modellerin güncel literatürde tahmin başarısının zaten önemli ölçüde yüksek olduğu sonucuna varılabilir. Kullanılan yöntem ne olursa olsun, makine öğrenmesi yöntemlerinin örüntü tanıma yetenekleri sayesinde Yun vd. (2021), gibi bu çalışmaların bazılarında elde edilen tahmin doğruluk (isabet) oranlarını aşmak giderek daha zor hatta imkansız hale gelmektedir.

Yeni veya mevcut tekniklerin farklı kombinasyonları ile birbirine yakın tahmin başarılarının raporlanması, daha fazla ilerleme kaydetmenin giderek daha zor olduğunu göstermektedir. Bu nedenle, hesaplama karmaşıklığını artıran yeni tekniklerle uğraşmak yerine, mevcut tekniklerin tahminlerinin güvenilirliğinin nasıl artırılacağına dair alternatif bir yön ortaya çıkmaktadır. Başka bir deyişle, bir tahmin modelinin başarısız tahminlerinin belirli bir yüzdesinden kaçınmasını sağlayacak bir yaklaşımla, yalnızca güvenilir tahminlerle devam ederek dolaylı olarak modelin genel başarısını önemli ölçüde daha yüksek bir seviyeye yükseltecektir. Bu yaklaşımın tek dezavantajı, öznitelik kümesinin yalnızca belirli kombinasyonlarının gerçek tahminlerle sonuçlanmasıdır, çünkü tahminlerin bir kısmı güvenilmez olarak nitelendirilecek ve dolayısıyla göz ardı edilecektir. Bunun sonucunda, çok yüksek doğruluk oranlarıyla tahminler yapmak, tüm tahmin ufkunun belirli dönemleri için tahmin yapmaktan kaçınmanın maliyetini beraberinde getirecektir.

Güncel literatürde bu tür çabaların açıklanabilir yapay zeka (explainable artificial intelligence (XAI)) adı altında incelendiği görülmektedir. Son zamanlarda XAI kapsamında iki farklı yaklaşım öne çıkmaktadır. Bu yaklaşımlardan biri, yerel yorumlanabilir modelden bağımsız açıklamalardır (local interpretable model-

agnostic explanations (LIME)) (Ribeiro vd., 2016) ve her bir tahminin örnek bazında açıklanarak herhangi bir modelin yorumlanmasına olanak tanımaktadır. Diğer bir yaygın yöntem ise SHapley Additive Explanations (SHAP) (Lundberg ve Lee, 2017) yöntemidir.

Bu çalışmada, ilk olarak, günlük borsa kapanış fiyat yönünü tahmin etmek için iki aşamalı bir istifleme topluluk tahmin modeli geliştirilmiştir. Öznitelik mühendisliği yerine EMD tekniğini kullanarak ayrıştırma yaklaşımı tercih edilmiştir. Deneysel mod ayrıştırmanın tercih nedeni ilerleyen bölümlerde şekilde açıklanmıştır.

EMD'nin tahmin modelinin işleyişini açıkça kolaylaştırdığı bu çalışmada gösterilmektedir. İçsel mod fonksiyonları (intrinsic mode functions (IMF)) olarak da bilinen bu ayrıştırılmış seriler, orijinal zaman serisinin basitçe alt bileşenleridir. İlk aşamada her bir IMF, iki ayrı ANN modeli[1] ile tahmin edilmiştir. İlk ANN modeli, her bir IMF'nin sürekli değerini tahmin etmek için kullanılmıştır. Başka bir deyişle, nicel değerlerini (regresyon tahmini) tahmin etmekte olup bu nedenle kısaca "ANNR" olarak adlandırılmaktadır. Öte yandan, ikincisi ANN modeli zaman serisinin yönünün (yukarı ve aşağı) sınıflandırılması için kullanılmış ve bu ANN modeli bir sınıflandırıcı model olarak tasarlandığından "ANN Sınıflandırıcı (classifier)" anlamına gelen ANNC olarak adlandırılmıştır. Böylece tüm IMF'ler için bu iki YSA modeli ayrı ayrı eğitilmiş ve model nesneleri tahmin yapmak için kaydedilmiştir. Bu iki farklı YSA modelinin kullanılmasının nedeni, yapılan ön araştırmalara göre belirli IMF'leri tahmin etmek için ANNC'nin, diğerleri için ise ANNR'nin üstünlüğünü ortaya koymasıdır. Bu nedenle, bu iki modelin tahminleri, ilk aşamada göreceli güçlerinden yararlanmak için birleştirilmiş ve ikinci aşamada birleştirilmiş tahminleri üçüncü bir tahmin modeline beslenmiştir.

İkinci aşamadaki üçüncü model, rastgele orman (RF) ve aşırı gradyan artırma (XGBoost) gibi farklı algoritmaların karşılaştırmalı performanslarına göre seçilmiş ve RF daha iyi sonuçlar verdiği için tercih edilmiştir. Ayrıca, bu tahmin modeli, orijinal zaman serisinin yönünü tahmin etmek için bir sınıflandırıcı (yukarı ve aşağı yön tahmini) olarak kullanılmıştır. Özetlemek gerekirse, genel mimari, sayısız kombinasyon arasında olası istifleme topluluğu modeli konfigürasyonlarından

[1] Burada ifade edilen ANN modeli ile ileri beslemeli yapay sinir ağı (feedforward artificial neural network (FNN)) kastedilmektedir.

biridir. Son olarak, tercih edilen tekniklere dayalı olarak önerilen topluluk modeli EMD-ANNRC-RF olarak adlandırılmıştır.

1.2. Hedef ve Amaçlar

Bu çalışmada literatüre katkı olarak ortaya koyduğumuz nokta, açıklanabilir yapay zeka yaklaşımlarından biri olan LIME algoritmasının bir topluluk sınıflandırıcıya entegre edilmesi durumunda ne gibi katkılar sunabileceğinin araştırılmasıdır. Literatür incelendiğinde, finansal zaman serileri tahmininde, modelin yaptığı tahminlerin güvenilirliğini "model açıklanabilirliği" yaklaşımı ile araştırarak tahmin başarısını arttıran herhangi bir araştırmaya rastlanmamıştır.

Bu kadar yoğun çalışılan bir konunun (borsa tahmini) bu boyutlarıyla ele alındığı bir çalışmanın literatüre doğrudan katkı sağlayacağı düşünülmektedir. Literatürde yeni sayılabilecek XAI yaklaşımlarının yön tahmin modelleri ile bütünleştirilmesi test edilmeli ve ileriki çalışmalarda daha da geliştirilmelidir. Bu nedenle bu çalışmada makine öğrenimi algoritmaları ile bütünleştirilebilen LIME algoritmasından nasıl yararlanılacağı ortaya koyulmuştur. Bu bağlamda bu bütünleştirmenin detaylarını açıklamak faydalı olacaktır.

XAI literatüründe, kullanılan tahmin modelinden bağımsız olarak, LIME adı verilen ve her bir tahmini örnek bazında açıklamaya çalışan bir yaklaşım bulunmaktadır. LIME, herhangi bir sınıflandırıcının tahminlerini yorumlanabilir ve modele güven sağlayacak şekilde açıklar. Tahmin özelinde yerel olarak yorumlanabilir bir model öğrenerek açıklamalar sunmaktadır (Ribeiro vd., 2016).

LIME, her tahmin için sınıf olasılıkları sağlayarak (bu çalışmadaki tahmin görevi yukarı ve aşağı tahmin olduğundan) tahmin modeline yeni bir özellik ekler. Bu çalışmanın esas noktası, tahmin modelinin ikinci aşamasında LIME algoritmasının RF modeli ile bütünleştirilmesidir. Yukarıda bahsedildiği üzere ANNR ve ANNC çıktıları RF modeline girdi olarak verilerek altı büyük borsa endeksinin yukarı ve aşağı yönlü hareket yönü tahmin edilmiştir.

Bu çalışmanın temel motivasyonu bu aşamada ortaya çıkmaktadır. Ön araştırmalar esnasında, RF modelinin ANNR ve ANNC'nin belirli çıktı değerleri için üstün doğrulukta tahminler yaptığı keşfedilmiştir. Genel olarak, RF modeline herhangi bir girdi örneği verildiğinde, 0 veya 1 tahminlerinden (çıktı) birini yapar.

Ancak, LIME algoritması uygulandıktan sonra, 0 ve 1 sınıf etiketlerinin her birine bir olasılık atanır. Böylece RF modeli tarafından yapılan her bir sınıf tahmininin olasılıkları LIME ile hesaplanabilmektedir. Örneğin, tahmin modeline herhangi bir girdi kümesi verildiğinde 0 sınıfı için 0,10 ve 1 sınıfı için 0,90 olasılığının elde edildiğini varsayalım. Sınıflar için hesaplanan bu olasılıklar, test kümesindeki her tahmin için elde edilmektedir.

Burada, sınıf olasılıkları, tahminler için güven düzeyi olarak kullanılmaktadır. Sınıf olasılıklarından biri karar verici için yeterince yüksekse, o zaman bu tahmine güvenir ve sonuca göre karar verir. Bir önceki örneğe göre, karar verici için 0,80 güvenilecek kadar yüksekse, karar verici 1 sınıf etiketi lehine karar verir. Burada 0.80 karar verici için güven düzeyidir. Öte yandan, karar verici güvenirlik düzeyini 0,91 olarak belirlerse 0,90<0,91 olduğu için herhangi bir karar vermekten çekinecektir. Başka bir deyişle, güven koşulu sağlanmamış olacaktır. Güven seviyesi 0,50 olarak ayarlanırsa, LIME, tahmin modeli ile bütünleştirilmemiş gibi sadece RF algoritması tarafından yapılan tahminleri elde ederiz.

Ancak, güven düzeyi arttıkça, bazı tahminler için güven koşulu sağlanamayacağından bazı tahminlerden kaçınılacak, ancak güvenilir tahminler için daha yüksek bir isabet oranı elde edilmesi beklenecektir. Diğer bir deyişle güvenirlik düzeyi düşük olan tahminlerden kaçınılacağı için test kümesinin tamamı için tahmin yapılamayacaktır. Bu hipotezi test etmek adına, 0,50 ile 1,00 arasındaki tüm güven seviyeleri için isabet oranındaki artış ve tahmin sayısındaki (doğruluk ve güvenilir tahmin sayısı değiş tokuşu) azalış ortaya konulmuştur.

Literatür incelendiğinde, LIME algoritmasının burada önerdiğimiz haliyle bir sınıflandırıcı modelinin zaman serisi tahmini uygulanması ile ilgili çalışmaya rastlanmamıştır.

2. TAHMİN MODELLERİ

Bu çalışmada çeşitli finansal yatırım araçları ile ilgili yatırım kararları önerebilecek istatistiksel ve makine öğrenmesi uygulamaları araştırılmaktadır. Bu anlamda, yatırım amaçlı finansal varlık alım ve satımlarında uygun bölgelerin belirlenmesi kritik öneme sahiptir.

Finansal zaman serilerindeki karmaşık ve doğrusal olmayan yapı nedeniyle, söz konusu uygun bölgelerin tespit edilmesi oldukça güç ve karmaşık bir işlemdir. Bu görev için geçtiğimiz yirmi yıl içerisinde çok sayıda makine öğrenmesi ve istatistiksel teknik önerilmiştir. Ancak literatürdeki bu çaba incelendiğinde büyük bir çoğunluğunda, yatırım kararlarında etkin bir araç geliştirmekten ziyade, finansal zaman serilerinin geliştirilen tekniklerin test edildiği bir parkur olarak kullanıldığı gözlemlenmektedir. Bunun ötesinde etkin yatırım stratejileri geliştirebilen görece az sayıda çalışma bulunmaktadır. Dolayısıyla etkin bir stratejiden söz edilebilmesi için finansal yatırım araçlarının uygun alım ve satım bölgelerinin tespiti önem arz etmektedir.

Uygun yatırım araçlarının seçiminde geçmişten günümüze başvurulan iki yöntem kullanılmakta olup bunlar temel ve teknik analiz olarak bilinmektedirler. Detaylarına birinci bölümde yer verilen bu yöntemlerin ortak amacı ise bahsi geçen uygun alım ve satım bölgelerinin tespit edilmesidir. Her iki yöntemin başarılı olduğu noktalar bulunmakla birlikte bu yöntemler ile kesin çıkarımlar yapılabilmesi mümkün değildir. Buna rağmen birer başvuru aracı olarak kullanılıyor olmalarının nedeni, çeşitli durum ve şartlarda büyük oranda işe yarıyor olmalarından ileri gelmektedir.

Önerilen çalışma ile çeşitli temel ve teknik analiz yöntemleri istatistiksel sınıflandırma teknikleri ile birleştirilerek hangi durum ve şartlarda, hangi oranlarda (olasılıklarda), hangi çıkarımların (alım veya satım) yapılabileceğini tespit etmek amaçlanmaktadır.

Bu amaçla, çeşitli finansal yatırım araçlarının veya menkul kıymetlerin uygun alım ve satım bölgelerinin tespiti için olasılıksal olarak sınıflandırılması planlanmaktadır. Bu nedenle, bu bölümde istatistiksel ve makine öğrenmesi alanlarında sınıflandırma yaklaşım ve teknikleri üzerinde durulacaktır.

2.1. İstatistiksel Sınıflandırma

Sınıflandırma, matematik ve bilimin bütün alanlarında karşılaşılan bir kavramdır. Sınıflandırma, bilimin farklı alanlarında özünde aynı olmakla birlikte çeşitli şekillerde tanımlanabilmektedir. Tanımlardaki bu farklılığın nedeni olarak farklı bilimsel alanların bilgiye ulaşmada birbirinden farklı araç, yöntem ve yaklaşımları kullanıyor olması düşünülebilir. Ancak söz konusu tanımlar incelendiğinde ortak fikir, birden fazla varlığın belirli ölçütler doğrultusunda birbirleriyle olan benzerlik ve farklılıklarını ortaya koyarak bu benzerlik ve farklılıklara göre gruplandırılmasıdır.

Bouveyron vd. (2019), sınıflandırmayı tanımlarken yukarıda yer alan tanıma ilave olarak, grupların doğası hakkında hali hazırda bir bilgi mevcut olduğunda bu gruplandırma işlemini sınıflandırma olarak tanımlamaktadır. Diğer taraftan gruplar hakkında herhangi bir ön bilgiye sahip olunmayabilir. Böyle veriler üzerinde yapılabilecek gruplama işlemlerine ise istatistik ve makine öğrenmesi literatüründe kümele denilmektedir.

İstatistiksel sınıflandırma, bilgisayar teknolojilerinin gelişmesi ile birlikte daha da önem kazanmış olmakla birlikte başlangıcı 18. yüzyıllarda matematiksel istatistiğin temellerini atan Laplace'a kadar gitmektedir (Katz, 1998). 1936 yılında Fisher tarafından ortaya koyulan diskriminant analizi istatistiksel sınıflandırma alanındaki en önemli gelişmelerden biridir (Das Gupta, 1980). Ancak, diskriminant analizinin sadece normal dağılım gösteren veriler için kullanılabiliyor olmasından dolayı 1960'lı yıllarda alternatif yaklaşımlar arayışına girilmiştir.

Temelde Bayes teoremini esas alan bu yaklaşımlar sınıflandırılma için olasılık teorisinden yararlanmaktadır. 1970'lerde, en yakın komşu yöntemi gibi bir veri noktasının sınıflandırılması için, o noktaya en yakın olan eğitim verilerinin sınıflarının kullanıldığı ayırıcı (discriminative) yöntemler geliştirilmiştir (Fix ve Hodges, 1989).

Son 30 yıl içerisinde yapay sinir ağları ve derin öğrenme gibi makine öğrenimi teknikleri, istatistiksel sınıflandırma alanında önemli bir rol oynamaktadır. Bu teknikler, büyük veri kümelerinde sınıflandırma yapmak için kullanılmakla beraber birçok uygulama alanında yer bulmaktadır.

Sınıflandırma ve kümeleme işlemlerinde uygulanan hesaplama ve işlem teknikleri arasındaki ayrıma işaret etmek adına özellikle makine öğrenmesi literatüründe sınıflandırma teknikleri kontrollü öğrenme (supervized learning),

kümele teknikleri ise kontrolsüz öğrenme (unsupervized learning) yöntemleri olarak bilinmektedir. Kısacası istatistiksel sınıflandırma, önceden belirlenmiş sınıflar veya kategoriler içindeki yeni gözlemlerin veya verilerin belirlenmesi sürecidir. Bu süreç, tıbbi teşhis, görüntü ve ses tanıma veya doğal dil işleme gibi çeşitli alanlarda yaygın olarak karşılaşılan bir problemdir.

Sınıflandırma için öncelikli olarak farklı kaynaklardan veriler toplanır ve analiz için uygun bir formata dönüştürülür. Sınıflar arasındaki farklılıkları ayırt etmede yardımcı olan ölçülebilir öznitelikler, toplanan verilerden çıkarılır. Özniteliklere örnek olarak görüntü tanıma probleminde renk, doku veya şekil olabilir. Çıkarılan öznitelikler, istenmeyen gürültüleri kaldırmak, verileri normalleştirmek ve analize uygun hale getirmek için genelde ön işleme tabi tutulurlar. Sınıflandırılacak farklı sınıfları temsil edebilmesi için bir eğitim kümesi seçilmesi gerekmektedir. Veri tipine, sınıf sayısına ve sınıflandırmada kullanılan özelliklere bağlı olarak belirli bir matematiksel sınıflandırıcı model seçilir. Belirli başlı sınıflandırıcı modellere örnek olarak lojistik regresyon, karar ağaçları ve destek vektör makineler gösterilebilir. Seçilen bu model, öznitelikler ve sınıflar arasındaki desenleri ve ilişkileri öğrenmek için seçilen eğitim kümesi üzerinde eğitilir. Eğitilen bu model, ayrı bir test kümesinde doğruluğunun ve performansının ölçülmesi için değerlendirilir. Model eğitilip sınandıktan sonra, yeni verileri önceden belirlenmiş sınıflara atamak için gerçek uygulamalarda kullanılabilir hale gelmiş olmaktadır.

Mirkin (1996)'nin matematiksel sınıflandırma konusuna ilişkin kapsamlı monografisinde ifade edildiği üzere sınıflandırma, yapısal olarak hiyerarşik ve hiyerarşik olmayan şeklinde olmak üzere iki ana gruba ayrılmaktadır. Hiyerarşik sınıflandırmada sınıflar arasında hiyerarşik bir sıralama bulunmakta ve sınıflandırma yukarıdan aşağıya doğru sıralanabileceği gibi aşağıdan yukarıya doğru da sıralanabilmektedir. Hiyerarşik olmayan sınıflandırmada ise, hiyerarşik bir sıra takip etmek yerine, nesneler arasındaki benzerliği veya yakınsaklığı açıklayabilecek bir ölçü elde etmek suretiyle bu nesneleri bir araya getirerek veya birbirinden ayırarak yeni sınıfların oluşması sağlanır. Sınıflandırma yapmaktaki temel ve en önemli amaç, belirli bir olay veya olguya ait yapının anlaşılabilmesine ve diğer olay ve olgularla olan ilişkisinin açığa çıkarılmasına olanak sağlanmasıdır.

Sınıflandırma işleminin yukarıda bahsedilen yapısal farklılıklarının yanı sıra benimsenen yaklaşım ve hesaplama teknikleri bakımından da yine iki farklı yöntem ile karşılaşılmaktadır. Öncelikle sınıflandırma işlemini matematiksel olarak ifade

edecek olursak, $X_i,\ i \in \{1, \dots, I\}$ oluşturulan sınıflara ait ayırt edici öznitelikler[2] olmak üzere, söz konusu sınıfları $k \in \{1,2, \dots, K\}$ için C_k ile gösterelim. Öyle bir f fonksiyonu olsun ki, $C_k = f(x_1, x_2, \dots, x_I)$ elde edilsin. Sınıflandırma işlemlerinde benimsenen yaklaşım ve hesaplama tekniklerindeki bu fark, burada kullanılan f fonksiyonunun işleyiş biçimi ve çerçevesindeki farklılıklardan kaynaklanmaktadır. Bu bağlamda f fonksiyonunun yapısal ve işleyiş özelliklerine dair istatistik ve makine öğrenimi literatüründe üretken (generative) yöntemler ve ayırıcı yöntemler olmak üzere iki ayrı yaklaşım söz konusudur. Şayet burada $f(x_1, x_2, \dots, x_I),\ C_k$ üzerinde tanımlı bir olasılık yoğunluk fonksiyonu ise bu durumda üretken sınıflandırma yönteminden söz edilecektir. Diğer taraftan $f(x_1, x_2, \dots, x_I),\ C_k$ üzerinde bir olasılık dağılımı tanımlamadan doğrudan sınıf etiketlerini (class labels) tayin ediyorsa bu durumda ayırıcı sınıflandırma yaklaşımı söz konusu olmaktadır. Bu aşamada, ayırıcı ve üretken yöntemlerin detaylarına değinmek yararlı olacaktır.

Ayırıcı ve üretken modeller, makine öğrenmesinde sınıflandırma problemlerini çözmek için kullanılan iki farklı yaklaşımdır. Ayırıcı modeller, doğrudan sınıflandırma işlemi için bir karar sınırı çizerler. Bu sınır, sınıflar arasındaki farklılıkları en iyi şekilde ayıran bir hiperdüzlem ya da benzeri bir yapıdır. Verilen bir öznitelik vektörüne dayanarak, ayırıcı model sınıf etiketi tahminini yapar. Bu modeller, sınıf etiketleri tahmin etmek için çok hızlı ve doğru olabilirler, ancak sınıflandırma işlemi sırasında veri kümesinin dağılımına ilişkin herhangi bir bilgi sağlamazlar. Üretken modeller ise veri kümesinin arka planındaki olasılık dağılımını modellerler. Bu modeller, veri kümesinin çeşitli parametrelerini öğrenir ve daha sonra bu parametreleri kullanarak yeni veriler oluştururlar veya bir verinin hangi sınıfa ait olduğunu tahmin ederler. Örneğin, bir üretken model, bir görüntüyü sınıflandırmak için kullanılabilir. Bu model, her sınıf için ayrı bir olasılık dağılımı öğrenir ve daha sonra bir veri noktasının hangi sınıfa ait olma olasılığına karar vermek için bu dağılımları kullanır.

2.1.1. Ayırıcı Yöntemler

Ayırıcı yöntemlerde verinin nasıl üretildiğinin herhangi bir önemi olmadan, öznitelik kümesini göz önünde bulundurarak doğrudan sınıf tayini yapılmaktadır. Ayırıcı yöntemlere örnek olarak verilebilecek tekniklerden en bilinenleri Logistic

[2] İng. “feature”.

regresyon, destek vektör makineleri, yapay sinir ağları, K-En yakın komşu, karar ağacı ve rastlantısal orman teknikleri gösterilebilir.

2.1.2. Üretken Yöntemler

Üretken yöntemler ile yapılan sınıflandırmalardaki temel yaklaşım, sistemdeki bütün değişkenlerin üzerine bir olasılık yoğunluk fonksiyonu tanımlayıp, sonrasında bu fonksiyonu kullanarak yeni gözlemleri sınıflandırmaktır. Bu nedenle üretken yöntem yaklaşımını esas alan teknikler deterministik model yerine olasılıksal model olarak nitelendirilirler. Üretken yöntemlerde sistemin girdileri (öznitelikler) ile çıktıları (sınıflar) ortak bir olasılık dağılım fonksiyonu ile ifade edilmeye çalışılır. Dolayısıyla öznitelik değişkenlerine ait yeni gözlemler modele girdiğinde, bu girdilerin sınıflara olan aidiyetleri olasılıklar ile ifade edilmiş olmaktadır. Üretken yöntemlere örnek teknikler olarak doğrusal ayırma analizi, naive bayes, hidden Markov modelleri, Bayesyen ağlar ve generative adversarial ağlar gösterilebilir.

2.1.3. Ayırıcı ve Üretken Yöntemler Arasındaki Farklılıklar

Ayırıcı ve üretken modeller arasındaki temel fark, üretken modellerin veri kümesinin arka planındaki olasılık dağılımını modellemesidir, ancak ayırıcı modeller sınıflar arasındaki karar sınırını belirlerler. Her iki model türü de makine öğrenmesi alanında önemli roller oynar ve uygulanabilecek pek çok alanda kullanılırlar.

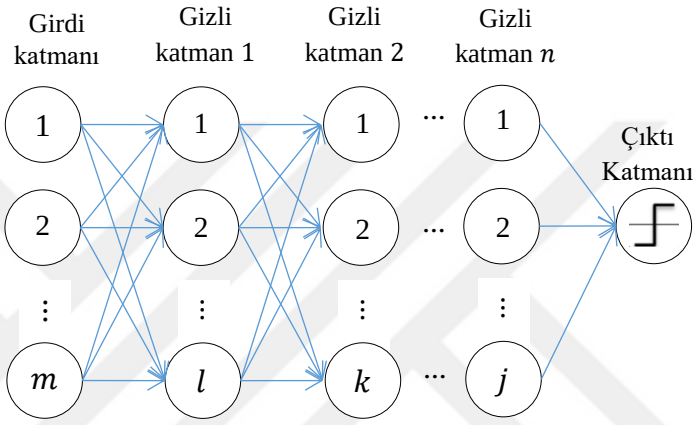
Üretken yöntemlerde, olasılıksal kavramlar ile açıklanabilen veri yapısı ve belirsizlik gibi kavramlar modelde iyi bir şekilde kullanılabiliyorken, ayırıcı yöntemler bunlardan yoksun kalmaktadır. Ayırıcı yöntemlerde bunun yerine hata fonksiyonları ve kernellerden yararlanılmaktadır. Dahası, ayırıcı yöntemlerin tek odak noktası sınıflandırma için değişken uzayını en uygun biçimde ayırmak olduğundan, veri uzayı hakkındaki ön bilgiler ve esnek modelleme araçlarından yararlanılamamaktadır. Bu nedenlerle ayırıcı yöntemler, değişkenler arasındaki ilişkilerin üretken yöntemlerdeki kadar açık veya görselleştirilebilir olmadığı kara kutular gibi düşünülebilir (Jebara, 2012).

2.2. Yapay Sinir Ağları

Yapay sinir ağlarının (YSA) başlangıcı, McCulloch ve Pitts (1943), tarafından geliştirilen ve sinir ağları için eşik mantığı olarak adlandırılan hesaplamalı modele dayanmaktadır. YSA, bir girdi ve bir çıktı kümesi arasındaki eşlemeyi öğrenen denetimli bir makine öğrenimi algoritmasıdır.

Temel YSA mimarisi üç katmandan oluşur ve her katman düğümlerden (nodes) meydana gelmektedir. Birinci katman girdi katmanıdır ve her düğüm bir girdiyi ifade eder. Gizli katman olarak adlandırılan ikinci katman birden fazla katman içerebilir. Son katman ise, çıktı katmanı olarak adlandırılır ve her girdi örneği için modelin çıktısını üretir.

Tasarıma bağlı olarak, düğümler birbirine bağlıdır, ancak tüm düğümlerin bağlı olması gerekmemektedir. Model, girdiler ve çıktılar arasındaki ilişkiyi yakalamak için bir veri örneğiyle eğitilir. Şekil 2.1.'de temsili bir YSA modeli gösterilmektedir.



Şekil 2.1. YSA modellerinin genel mimarisi.

YSA'ların bugünkü anlamıyla eğitilmelerini sağlayan çok katmanlı sinir ağlarının temelini oluşturan algoritma geri yayılım algoritmasıdır. Geri yayılım algoritması, yapay sinir ağlarında ağın ürettiği çıktı ile istenen çıktı arasındaki farkı (hatayı) en küçükleyerek ağın parametrelerini en uygun hale getirmek için delta kuralının genelleştirilmiş halinin uygulanmasıdır.

Burada ağın ürettiği çıktı ile istenen çıktı arasındaki hata miktarını en aza indirmek için gradyan arama tekniği kullanılmaktadır. Geri yayılım algoritması ileri ve geri yayılım olmak üzere işleyişini iki aşamada tamamlamaktadır. Öncelikle girdiler, ağın girdi katmanından başlayıp, gizli katmalarından ilerleyerek çıkış noktasına doğru yayılmaya başlar. Ağın gizli katmanlarında her bir girdi için ağırlıklar bulunmaktadır. Her bir girdi, kendisine ait ağırlıkla çarpılıp toplandıktan sonra gizli katmanda yer alan etkinleştirme fonksiyonuna[3] ulaşır. Birden fazla gizli

[3] Etkinleştirme fonksiyonu, yapay sinir ağlarında bir nöronun giriş değerlerine ve o andaki bağlantı ağırlıklarına bağlı olarak çıkış değerini hesaplayan fonksiyondur. (*İng. activation function*). (*TÜBA - Türkçe Bilim Terimleri Sözlüğü*, t.y.). Etkinleştirme fonksiyonu olarak genelde sigmoid, ReLU (Rectified Linear Unit) gibi fonksiyonlar kullanılmaktadır.

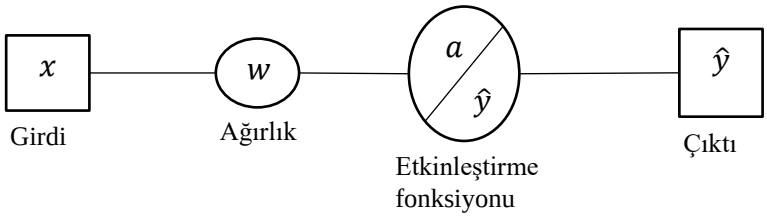
katman olması halinde bu işlem ileriye doğru çıkış katmanına ulaşana kadar her bir gizli katman için tekrarlanır. Bahsi geçen hesaplamalar yapılıp nihai olarak çıktı katmanına ulaşıldıktan sonra bu değerler çıktı katmana ait etkinleştirme fonksiyonuna verilir ve çıktı katmanının etkinleştirme fonksiyonu bir çıktı üretir. Elde edilen bu çıktıya ağın çıktısı veya tahmini denilmektedir.

Ağın çıktısı ile esas olarak ağın üretmesini istediğimiz veya beklediğimiz çıktı arasındaki farka ise hata veya sapma denilmektedir. Bu noktada ağın ilk aşaması tamamlanmış olup ikinci aşaması olan geri yayılım aşaması başlamaktadır. Geri yayılım aşamasında, elde edilen bu hata ağ boyunca geriye doğru yayılarak ilerler. Hata geriye doğru yayılırken, ağda yer alan ağırlıklar, gradyan inişi algoritması ile bu hata miktarını en aza indirecek şekilde yeninden ayarlanır. Dolayısıyla hata bir f fonksiyonu olarak tanımlancak olursa $\hat{f}(x) = \min_x f(x)$ olarak ifade edilmektedir. gradyan inişi algoritmasında fonksiyonun en küçük olduğu noktayı tespit edebilmek için öncelikle rastgele bir x_1 değeri için $df(x_1)/dx_1 > 0$ olması durumunda fonksiyonun artış halinde olduğu belirlenir ve ardından daha küçük bir x değeri için yeniden hesaplama yapılır. Benzer şekilde bu defa da $df(x_2)/dx_2 < 0$ olması durumunda ise fonksiyon azalmakta ve daha yüksek bir x değeri verilmesi gerekmektedir. Bu durumda $x_2 = x_1 - \alpha(df(x)/x)$ olmaktadır. Ağın eğitim sürecinde ağırlık değerlerinde yapılan değişiklikler α ile gösterilmektedir. Böylece

$$\Delta x = -\alpha \frac{df(x)}{x}$$

elde edilir. Bu işlem, katmanlar boyunca geri yayarak türevin zincir kuralını uygulayarak yapılır ve böylece hesaplanan gradyanlar, ağın parametrelerinin değerlerini yeniden ayarlamak için kullanılır. İleri ve geri yayılım işlemeleriyle, sinir ağından beklenen performans elde edilene kadar parametreler tekrar tekrar güncellenerek ağın performansı iyileştirilir.

Nihai olarak burada hatayı en küçüklenerek sinir ağının tahminlerinin doğrulunun arttırılması amaçlanmaktadır. Örneğin kolaylık olması adına bir gizli katmanı ve bir girdi ile bir çıktısı olan sinir ağının işleyişini inceleyebiliriz.



Şekil 2.2. Tek girdi, tek gizli katman ve bir çıktısı olan YSA

Etkinleştirme fonksiyonu $\hat{y}$'nin sigmoit fonksiyonu olduğunu varsayacak olursak,

$$\hat{y} = \sigma(a) = \frac{1}{1 + e^{a}}$$

Burada $a(w) = xw$ olmak üzere, ağın üretmesi beklenen esas çıktıyı t ile gösterecek olursak, bu durumda hata fonksiyonu şu şekilde ifade edilir;

$$E = \frac{1}{2}(t - \hat{y})^2$$

Hata fonksiyonunda yer alan $1/2$ terimi, türev işleminde kolaylık olması için bulunmaktadır. O halde,

$$\frac{\partial e}{\partial w} = \frac{\partial e}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a} \cdot \frac{\partial a}{\partial w}$$

ayrıca,

$$\frac{\partial \sigma(a)}{\partial w} = \frac{\partial \sigma(a)}{\partial a} \cdot \frac{\partial a}{\partial w} = \sigma(a)\big(1 - \sigma(a)\big)x$$

olmak üzere, nihai olarak

$$\frac{\partial e}{\partial w} = -\sigma'(a)(t - y)x$$

$$\Delta w = \alpha\sigma'(a)(t - y)x$$

bulunur. YSA ile hem regresyon tahmini (sürekli sayısal değerlerin tahmini) hem de sınıflandırma (sınıf etiketinin tahmini) yapılabilmektedir. Bu çalışmada yapay sinir ağının bu iki özelliği de kullanılmıştır.

3. AÇIKLANABİLİR YAPAY ZEKA

Bilgisayarların belirli görevleri yerine getiriken insanlarda olduğu gibi deneme yanılma yöntemi ile tecrübe ederek öğrenebilecekleri fikri her zaman merak konusu olmuştur. Her ne kadar insanların ve diğer canlıların "öğrenme" süreçleri tam anlamıyla anlaşılamamış olsa da belirli öğrenme görevi türleri için etkili olan algoritmalar icat edilmiştir ve teorik bir öğrenme anlayışı ortaya çıkmaya başlamıştır (Mitchell, 1997).

Makine öğrenmesi algoritmaları çok büyük verilerle çalışarak bir görevi yerine getirmek adına belirli bir modelin, veriler arasındaki sebep sonuç ilişkisini eşlemlemesini[4] sağlar. Dolayısıyla makine öğrenmesi algoritmalarını ve modellerini bir matematiksel fonksiyon olarak anlaşılabilmektedir. Makine öğrenmesi, bir problemin çözümü için veriye dayalı bir yaklaşım benimser. Bu yaklaşımda, bir matematiksel model veya algoritma, veriye dayalı bir örüntüyü yakalamak veya bir tahmin yapmak için kullanılır.

Makine öğrenmesi algoritmaları, genellikle veri kümelerindeki ilişkileri öğrenmek ve bu ilişkileri temsil edebilecek matematiksel fonksiyonları bulmak için kullanılır. Öğrenme süreci, bir modelin parametrelerini ayarlayıp, veriye uyum sağlayarak gerçekleştirilir. Tıbbi kayıtlardan hareketle hangi hastalıklarda hangi tedavilerin uygulanacağı veya hangi teşhisin koyulacağını öğrenme görevi örnek olarak verilebilir. Verilerden hareketle makine öğrenmesi algoritmaları ile öğrenen modeller daha sonrasında öğrenmiş oldukları bu kuralları kullanarak bir görevi gerçekleştirmek için uygulamaya hazır hale getirilmektedir. Diğer bir deyişle, algoritmanın öğrendiği kuralları kullanarak yeni veriler için tahminler yapılabilmektedir.

Makine öğrenmesi modellerinin doğru çalıştığından veya yapmış oldukları tahmin ve çıkarsamaların doğruluğundan emin olmak için kararları nasıl aldıklarına dair bir anlayışa sahip olmak sağlıklı kararlar alabilmek adına önem arz etmektedir. Ancak bu modellerin iç işleyişi ve karar verme süreçleri genelde anlaşılmaz veya açıkça açıklanamaz olduğundan dolayı literatürde kara kutu olarak adlandırılırlar.

Tıbbi kayıtlardan verilen örnekte olduğu gibi kritik uygulamalarda, karar verme sürecinin anlaşılması ve modellerin doğru çalıştığından emin olunması önem

[4] Eşlem (*İng. map; mapping*): Bir kümedeki değer ya da niceliklerin bir başka kümedeki değer ve niceliklerle tekil bir şekilde bağlanması. Matematikte eşanlamlısı "fonksiyon". (*TÜBA - Türkçe Bilim Terimleri Sözlüğü*, t.y.)

arz etmektedir. Bu nedenle, araştırmacılar, makine öğrenmesi modellerinin iç işleyişini daha iyi anlamak için çalışmalar yapmaktadırlar. Bu çalışmalar açıklanabilir yapay zeka çatısı altında yürütülmektedir.

Son yıllarda yapay zeka ve makine öğrenimi alanındaki çalışmaların ilerlemesiyle birlikte açıklanabilir yapay zeka kavramı ortaya çıkmıştır. 1950'lerde başlayan yapay zeka araştırmalarıyla birlikte insan ve diğer canlıların davranış ve karar alma süreçlerinin matematiksel bir çerçevede nasıl ifade edilebileceğine dair araştırmalar başlamıştır. Ardından 1960'larda ilk makine öğrenimi modelleri geliştirilmiş olmakla birlikte bu modellerin matematik ve istatistik teorisi açsından arka planda yetersiz teoreme sahip olmasından dolayı işleyişleri ve nasıl karar aldıkları üzerinde yeterli anlayışa sahip olunamamıştır. Geliştirilen makine öğrenimi modelleri bu özelliklerinden dolayı, yani üretmiş oldukları sonuçların nasıl üretildiği tam olarak anlaşılamadığı için literatürde kara kutu olarak adlandırılmışlardır. Özellikle 90'lı yıllarda bilgisayar teknolojilerinin de gelişmesi ile birlikte makine öğrenimi modellerinin matematiksel çerçevesi ve işlem adımları daha karmaşık hale gelmiştir ve böylece elde ettikleri sonuçların nasıl üretildiği konusu daha da zor anlaşılır hale gelmiştir. Açıklanabilir yapay zeka çatısı altında son yıllarda bu duruma bir çözüm getirebilmek adına çalışmalar yoğunlaşmaktadır. Böylece makine öğrenimi ve derin öğrenme modellerinin almış olduğu kararların nasıl üretildiğine dair şeffaf bir anlayış sağlanabilmesi hedeflenmektedir. En güncel ve ünlü yaklaşımlara örnek olarak LIME (Local Interpretable Model-Agnostic Explanations) ve SHAP (SHapley Additive exPlanations) yer almaktadır (Ribeiro vd., 2016; Lundberg ve Lee, 2017).

Açıklanabilir yapay zeka (e**X**plainable **A**rtificial **I**ntelligence (XAI),) makine öğrenimi modellerinin işleyişi hakkında anlaşılır bir şekilde açıklama yapılabilmesine olanak tanıyan bir yaklaşımdır. XAI, özellikle derin öğrenme gibi karmaşık modellerin kullanıldığı alanlarda, sonuçların nasıl elde edildiği hakkında şeffaf bir anlayış sağlamak için kullanılır. XAI, yapay zeka modellerinin kararlarının ve sonuçlarının anlaşılır ve açıklanabilir olması için bir dizi teknik kullanır. Bu teknikler, makine öğrenimi modellerinin içindeki veri işleme adımlarını, özellikle de modelin hangi özellikleri veya faktörleri dikkate aldığını ve hangi sonuçlara ulaştığını anlamak için kullanılır.

XAI, birçok uygulama alanında kullanılır. Örneğin, sağlık sektöründe, bir hastalığın tanısını koymak veya bir tedavi planı oluşturmak için kullanılan makine

öğrenimi modellerinin nasıl karar verdiğini anlamak önemlidir. Özellikle sınıflandırma problemlerinde oldukça yararlı sonuçlar üretmektedir. Belirli bir makine öğrenmesi modeli ile herhangi bir hastalık tanısına dair sınıflandırma yapıldığında, modelin yapılan bu sınıflandırmayı nasıl yaptığını ve bu sınıflandırmanın hangi oranda güvenilir olduğunu gösterebilmektedir. XAI teknikleri, aynı zamanda finans, e-ticaret, güvenlik ve otomotiv sektörlerinde de kullanılmaktadır. XAI'nin kullanımı, makine öğrenimi modellerinin kararlarının insanlar tarafından anlaşılması ve doğrulanması için gereklidir. Ayrıca, modelin işleyişinin şeffaf olması, hatalı veya yanlış sonuçların tespit edilmesini ve düzeltilmesini kolaylaştırır.

3.1. Yerel Yorumlanabilir Modelden Bağımsız Açıklamalar (LIME)

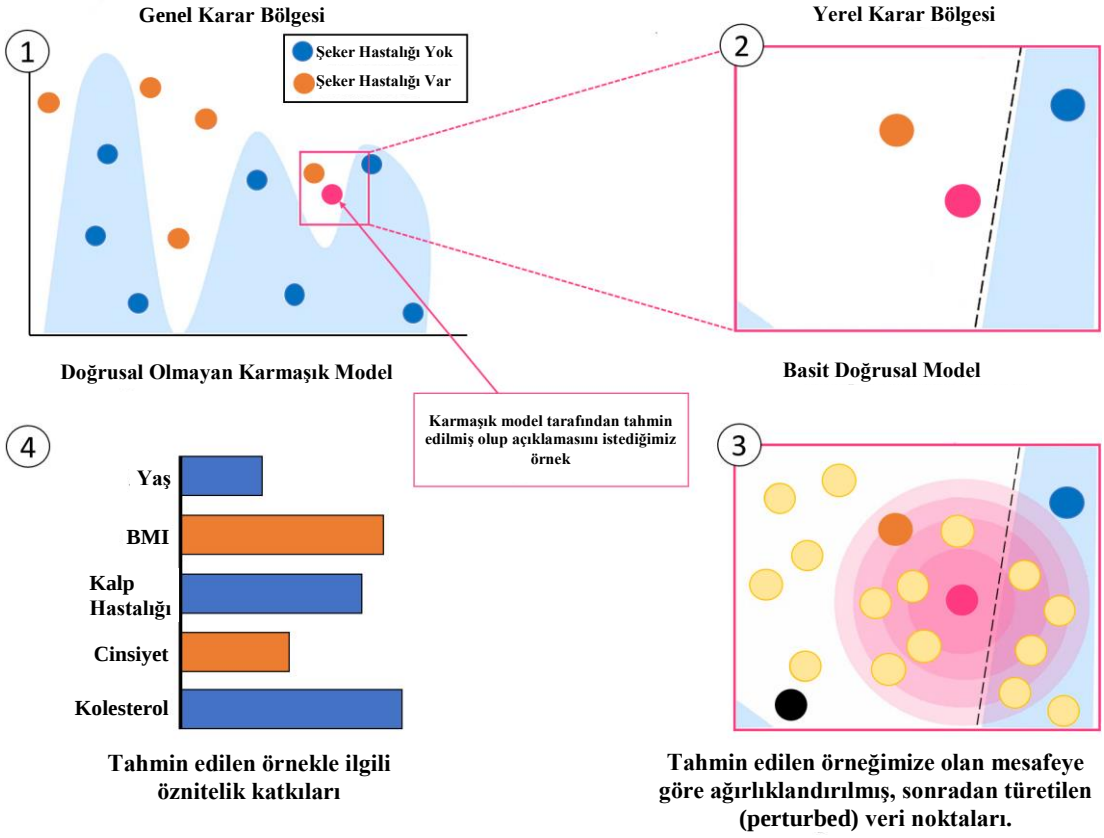
Herhangi bir sınıflandırıcının veya regresyon tahmin edicisinin belirli bir dereceye kadar güvenilirliğe izin vererek, her bir tahmin için açıklamalar sağlamak adına Yerel Yorumlanabilir Modelden Bağımsız Açıklamalar algoritması Ribeiro vd. (2016), tarafından önerilmiştir. Kısacası LIME, belirli bir tahmin için güven düzeylerinin elde edilmesini sağlar. LIME, modelden bağımsız olarak bir modelin *"yerel"* olarak güvenilir olup olmadığına[5] karar verir ve bir modelin bir tahmin etrafında özniteliklerin nasıl *"açıklanacağını"* ortaya koyar. LIME algoritmasının bu özelliği, yerel güven olarak bilinir (Rothman, 2020). Aynı zamanda bu açıklamanın *"Yorumlanabilir"* veya insanlar tarafından anlamlandırılabilir olması gerekmektedir. Ayrıca LIME herhangi bir makine öğrenimi modeline uygulanabilir, dolayısıyla *"Modelden Bağımsızdır"*.

LIME'ın işyeyişi Şekil 3'te basitçe gösterilmektedir. LIME'ın uygulanabilmesi için öncelikle herhangi bir Makine öğrenimi modelinin eğitilmiş ve tahmin yapmaya hazır hale gelmiş olması gerekmektedir. Açıklanması istenen herhangi bir örnek (pembe nokta) söz konusu olduğunda, öncelikle sadece pembe noktanın yer aldığı yerel düzleme odaklanılır. Diğer bir deyişle, modelin karar düzleminin geri kalanında nasıl davrandığı ile ilgilenilmez. Bu aşamada önemli bir nokta açığa çıkmaktadır. Açıklamanın yapılacağı yerel noktaya odaklanıldığında, örneklerin hangi sınıfa ait olacağını belirleyen özniteliklerin bazıları yerel bölgede daha belirleyici olabiliken, bazıları daha az öneme sahip olabilmektedir (Nieto

[5] Burada yer alan "yerel" olarak güvenilirlik kavramıyla "*yerel sadakat (local fidelity)*" ifade edilmek istenmektedir. Esas olarak LIME'ın yapmış olduğu açıklamanın, sınıflandırıcının (tahminlerini açıklamaya çalıştığımız makine öğrenmesi modeli) davranışını en iyi şekilde yansıtması arzu edilmektedir.

Juscafresa, 2022). Dolayısıyla makine öğrenimi modeli için genel karar düzleminde daha çok (daha az) öneme sahip olan öznitelikler, yerel bölgede daha az (daha çok) öneme sahip olabilir.

Şekil 3'de sağ alt köşede yer alan üç numaralı görselde, artık sadece açıklanacak örneğin bulunduğu yerel bölge ile ilgilenildiğinden, sadece bu bölgede yer alan örneklerin açıklanabilmesi adına vekil (surrogate) bir model tayin edilir. Şekilde vekil model olarak doğrusal bir model belirlenmiştir (siyah noktalı çizgi). Anlaşılacağı üzere bu vekil modelin de eğitilmesi gerekmektedir. Vekil modelin eğitilebilmesi için pembe noktanın etrafında farklı örnekler (sarı noktalar) tedirgenerek[6], bu örnekler pembe noktaya olan uzaklıklarına göre ağırlıklandırılır. Nihai olarak pembe noktanın tahminlenmesinde hangi özniteliğin daha büyük öneme sahip olduğuna dair açıklama elde edilmiş olur (bkz. Şekil 3'te dört numaralı görsel).



Şekil 3. Bir hastanın şeker hastası olup olmadığını belirlemeye çalışan bir sınıflandırıcının herhangi bir tahminini LIME'ın açıklamaya yönelik işlem adımları (Nieto Juscafresa, 2022).

[6] Tedirgeme kuramı (perturbation theory): Bir türevsel denklemin çözümünü ondan biraz değişik bir denklem çözümünden yürüyerek matematik bir dizi biçiminde elde etme yöntemi (TÜBA - Türkçe Bilim Terimleri Sözlüğü, n.d.). Bir diğer tanıma göre tedirgeme kuramı, cevabı kesin olarak bilinen daha basit bir problemin çözümünden yola çıkarak, bir probleme yaklaşık bir çözüm bulmaya yönelik yöntemleri içermektedir (Holmes, 2013).

LIME tarafından üretilen açıklama aşağıdaki genel formülün matematiksel optimizasyonu ile elde edilir (Molnar, 2022);

$$explanation(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

burada x, açıklanmak istenen ana model f ($f: \mathbb{R}^d \rightarrow \mathbb{R}$) tarafından tahmin edilmiş örnektir. x örneği için açıklama modeli, hem model karmaşıklığını[7] ($\Omega(g)$) düşük tutan hem de açıklamanın ana model f'nin (örneğin xgboost modeli) tahminine ne kadar yakın olduğunu ölçen kayıp fonksiyonu $\mathcal{L}$ (örneğin ortalama hata kareleri)'yi en aza indiren g (örneğin doğrusal regresyon modeli) modelidir.

Daha önce ifade edildiği üzere, vekil model g'nin eğitilebilmesi için, yerel bölgede x'e olan uzaklık dikkate alınarak π_x tarafından ağırlıklandırılmış yeni örnekler türetilmektedir. LIME, bu yeni örnekleri türetebilmek için eğitim kümesi ve normal dağılım göz önünde bulundurularak x örneğinin öznitelik verlerinde küçük değişikliler yapılır. Diğer bir deyişle, bu yeni örnekler her bir özniteliğin ortalaması ve standart sapması göz önünde bulundurularak, normal dağılımdan örnekleme yapılarak elde edilir. Rastgele veri türeterek elde edilen bu yeni veri seti (Z) kullanılarak, Açıklama fonksiyonu, $explanation(x)$, optimize edilmektedir. Açıklama fonksiyonunda $\mathcal{L}(f, g, \pi_x)$ ve $\Omega(g)$ olmak üzere LIME için optimize edilmesi gereken iki kayıp fonksiyonu bulunmaktadır. Bunlardan ilki

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathrm{Z}} \pi_x(z)\big(f(z) - g(z)\big)^2$$

olmak üzere $\mathcal{L}$, $\pi_x(z)$ tarafından tanımlanan bir yakınlık ölçüsü içinde, g'nin ölçümlerinin f'den ne kadar uzaklaştığının ölçüsünü gösterir. Burada $\pi_x(z) = exp(-D(x,z)^2/\sigma^2)$ olmak üzere, π_x belirli bir mesafe fonksiyonu D (genellikle öklit mesafesi) üzerinde tanımlanan üssel kerneldir ve x'in yerel komşuluğunu veya yakınlık ölçüsünü ifade eder. $\sigma^2 \in [0,1]$ ise kernel genişliğini göstermektedir. Özetle, $\mathcal{L}$ temel olarak f modeli ile g modelinin tahmini arasındaki mesafelerin karelerinin toplamına eşit olmaktadır. Buna ilaveten, yakınlık ölçüsü π_x, türetilmiş veri noktasının x örneğine ne kadar yakın olduğunu ölçümlemek için formüle

[7] Model karmaşıklığı, öğrenilmeye çalışılan fonksiyonun karmaşıklığını ifade eder (P. Schneider ve Xhafa, 2022, p. 8). Makine öğrenimi modelinin yapısı ve sahip olduğu parametre sayısı, ne kadar karmaşık olduğunu belirler. Bu açıdan düşünüldüğünde model karmaşıklığını düşük tutmak için izlenebilecek yollardan birine örnek olarak daha az sayıda öznitelik kullanımı gösterilebilir. Örneğin doğrusal regresyon modelini ele alalım. Doğrusal regresyon modelinin karmaşıklığı, öznitelik sayısı arttıkça artar, çünkü bu durumda daha fazla ağırlık parametresinin tahmin edilmesi gerekecektir. Böylece öznitelik uzayı arttıkça doğrusal regresyon modelinin karmaşıklığı da artacaktır. Benzer şekilde rastgele orman modelinde kullanılan ağaç sayısı arttıkça, modelin karmaşıklığı da artacaktır.

eklenmiştir. Bunun nedeni, Şekil 3'te üç numaraları görselde olduğu gibi, siyah veri noktasının yanlış sınıflandırılmasının, yerel bölgenin dışında kaldığı için konu ile ilgisinin bulunmamasıdır.

LIME uygulamada, G için doğrusal modeller sınıfını kullanmaktadır. Öyle ki,

$$g(z) = w_g \cdot z$$

Böylece $\mathcal{L}$ modeli ile, ağırlıklandırılmış bir doğrusal regresyon modeli minimize edilmeye çalışılmaktadır. İkinci hata terimi olan $\Omega(g)$ ise, g modelinin karmaşıklığını ifade etmektedir. Model karmaşıklığının hesaplanması, modelden modele değişiklik göstermektedir. Örneğin doğrusal regresyon modeli için model karmaşıklığı $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_n x_n + \varepsilon$ ile elde edilirken rastgele orman için karar ağaçlarının belirli bir oylama mekanizmasıyla birleşiminden elde edilmektedir.

4. SİNYAL İŞLEME

Sinyal işleme, zaman veya uzayla değişen fiziksel niceliklerin analizini, manipülasyonunu ve yorumlanmasını içeren bir alandır. Sinyal işlemenin geçmişi Mısırlılar ve Yunanlılar gibi eski medeniyetlere kadar dayanmaktadır. Mısırlılar ve Yunanlılar sinyalleri anlama ve manipüle etme konusunda müzikal harmonileri ve sesin özelliklerini inceleyerek seslerin özelliklerini anlamaya çalışmışlardır. İlerleyen dönemlerde, 17. yüzyıla gelindiğinde ise Isaac Newton ve Gottfried Wilhelm Leibniz gibi matematikçiler, sürekli sinyalleri analiz etmek için matematiksel bir çerçeve ortaya koyan kalkülüsü geliştirmişlerdir.

Telgrafın 1830'larda icat edilmesi ile birlikte sinyal işlemede önemli bir aşamaya gelinmiştir. Telegraf teknolojisi, bilgiyi elektrik sinyallerine kodlayarak kablolar vasıtasıyla uzun mesafelere iletmeyi sağlamaktaydı (Telegraph | Invention, History, ve Facts | Britannica, 2023). 19. yüzyılda telefonun geliştirilmesiyle telekomünikasyon teorisi başlamış oldu. Bu durum, sinyallerin verimli iletimi ve alımı için analizler içeren bir çalışma alanını da beraberinde getirdi. Buraya kadar olan dönemde sinyal işlemenin mekanik bir işlem olduğu söylenebilir.

Günümüzde sinyal işlemeden bahsederken esas olarak 20. yüzyılda ortaya çıkan dijital sinyal işleme alanı anlaşılmaktadır. Bilgisayarların ve dijital teknolojilerin ortaya çıkması, sinyal işlemenin devrim niteliğinde bir değişim geçirmesine olanak tanımıştır. Bunlara örnek olarak 1980'lerden sonra ortaya çıkan *dalgacık* analizi (wavelet analysis), en küçük kareler ve tekrarlamalı en küçük kareler algoritmaları gösterilebilir (Schneider ve Farge, 2006; Cadzow, 1994).

Signal işleme görüntü, ses ve video işleme, radar ve sonar sistemleri gibi oldukça geniş uygulama alanları bulmaktadır. Son yıllarda özellikle veri bilimi ve makine öğreniminin ortaya çıkmasıyla birlikte sinyal işleme alanının kapsamını daha da genişletmiş ve örüntü tanıma ve sınıflandırma modelleriyle birlikte uygulama alanları bulmuştur.

4.1. Dijital Sinyal İşleme

Sinyal işleme, yukarıda da ifade edidiği üzere analog veya dijital ortamda işlem gören sinyallerin açığa çıkarılarak analiz edilmesi işlemidir. Analog sinyallerin çeşitli yöntemlerle dijital yapıya dönüştürülerek bilgisayar ortamındaki uygulama süreçlerine ise dijital sinyal işleme denilmektedir. Böylece bilgisayarların hesaplama kapasitesinin üstünlüğünden yararlanılarak sinyal işleme süreçlerinin daha verimli

bir biçimde yapılabilmesine olanak sağlanmaktadır. Özellikle bilgisayar ve internet teknolojilerinde geçmişten günümüze yaşanan gelişmeler ses ve görüntü gibi sayısız verinin büyük miktarlarda depolanabilmesini sağlamıştır. Dolayısıyla dijital ortamda depolanan bu verilerin verimli bir şekilde işlenerek analiz edilebilmesi için sinyal işleme alanında yapılan araştırmalar önem kazanmaktadır.

Geleneksel sinyal işleme teknikleri genellikle sinyallerin doğrusal ve durağan/sabit özelliklerde olduğunu varsayar ancak bu özellikler gerçek dünya sinyalleri için her zaman geçerli olmamaktadır. Bu nedenle konuşma tanıma, finans, meteorolojik araştırmalar ve biyomedikal mühendisliği gibi çeşitli alanlarda yaygın olarak karşılaşılan doğrusal ve durağan olmayan sinyallerin analizinde yetersiz kalabilmektedir. Klasik modellerden kaynaklanan bu sınırlamayı gidermek adına EMD gibi doğrusal olmayan ve durağan olmayan sinyalleri etkili bir şekilde işleyebilen uyarlanabilir yaklaşımlar önerilmiştir. Belirli bir sinyali modlara ayrıştırarak sinyali yerel bileşenlerine ayıran EMD, sinyalleri zaman içinde değişen frekans içeriklerine uygun hale getirir.

4.2. Deneysel Mod Ayrıştırma

EMD, Huang vd. (1998) tarafından geliştirilen, doğrusal ve durağan olmayan verileri analiz etmek için önerilen bir yöntemdir. Karmaşık bir veri kümesi, sınırlı sayıda IMF'ye ayrıştırılabilir ve bu ayrıştırma, doğrusal ve durağan olmayan serilere uygulanabilmesi için verilerin yerel karakteristik zaman ölçeğine dayanır (Huang vd., 1998). Deneysel mod ayrıştırması, bir sinyalin farklı bileşenlere ayrılmasıdır ve genellikle veri analizinde, veri bir bütün olarak ele alındığında elde edilemeyen bazı bilgiler çıkarılmak isteniyorsa başvurulmaktadır. Bu nedenle, verilerin ayrıştırılması, verilerin doğasında bulunan özelliklere ilişkin iç dinamikler elde etmek için yeni elde edilen bileşenlerin analiz edilmesini sağlar. Her bir mod, orijinal sinyalin belirli bir bölümünü temsil eder. Belirli bir $X(t)$ sinyali için EMD algoritması şu şekilde çalışmaktadır (Huang vd. , 1998):

1. Yerel maksimum $X_{üst}$ ve yerel minimum X_{alt} tarafından ayrı ayrı tanımlanan $X(t)$ sınırları belirlenir.
2. Ekstremumlar tanımlandıktan sonra, tüm yerel maksimumlar, üst sınır olarak $X_{üst}(t)$ kübik spline çizgisi ile bağlanır.
3. Alt sınır $X_{alt}(t)$'ı üretmek adına yerel minimumlar için bir önceki izlek tekrarlanır.
4. Üst ve alt sınırın ortalaması m_t ise,

$$m_t = \left(X_{\text{üst}}(t) + X_{alt}(t)\right)/2$$

5. Öyleyse ilk bileşen, $h_1(t)$,

$$h_1(t) = X(t) - m_t$$

6. Elde edilen ilk bileşenden sonra, $X(t)$, $h_1(t)$ ile değiştirilir ve durdurma kıstası sağlanana kadar tüm izlek tekrarlanır, burada durdurma kıstası:

$$\sum_t \frac{\left(h_i(t) - X(t)\right)^2}{X(t)^2} < \epsilon$$

Burada ϵ eşik değerdir ve genelde yaklaşık olarak 0,2'ye eşittir. Bu ayrışma yöntemi belirli bir matematiksel temele dayanmaz, daha ziyade verilerin kendisi ayrıştırmayı belirler ve doğrusal ve durağan olmayan süreçlere uygulanabilir.

5. ÖNERİLEN MODEL: İKİ AŞAMALI TOPLU EMD-ANNRC-RF MODELİ

Hisse senedi piyasası tahmin literatüründe iki temel yaklaşım bulunmaktadır. İlk yaklaşım, regresyon tahmini olarak da bilinen zaman serilerinin gerçek fiyat seviyelerini tahmin etmektir. Bu yaklaşımda, tahmin edilen değerleri, hata kareler ortalamasının karekökü (root mean squared error (RMSE)), ortalama hata karesi (mean squared error (MSE)), ortalama mutlak hata (mean absolute error (MAE)) ve ortalama mutlak yüzde hatası (mean absolute percentage error (MAPE)) gibi bilinen değerlendirme ölçütlerine göre gözlenen değerlerle karşılaştırmak yaygın bir uygulamadır. Son zamanlarda, tahmin performansını değerlendirmek için tahmin yönü doğruluğu (isabet oranı olarak da bilinir) veya diğer bir deyişle hareket yönünü doğru tahmin etmek başarılı borsa tahminleri için hayati önem taşıdığından daha yaygın bir kriter haline gelmektedir. İsabet oranı (İsabet rate) şu şekilde hesaplanmaktadır,

$$\text{İ}sabet = \frac{TP + TN}{TP + TN + FP + FN}$$

Burada TP doğru pozitif tahminler, TN doğru negatif tahminler, FP yanlış pozitif tahminler ve FN yanlış negatif tahminlerdir. İsabet oranıyla birlikte yaygın olarak kullanılan diğer performans değerlendirme ölçütleri, *kesinlik (Kesinlik), Duyarlılık (Duyarlılık), F1-score* ve eğri altındaki alan (*area under the receiver operating characteristic curve* (ROC-AUC)) olarak sıralanmaktadır. Bu değerlendirme ölçütleri şu şekilde elde edilir:

$$Kesinlik = \frac{TP}{TP + FP}$$

$$Duyarlılık = \frac{TP}{TP + FN}$$

$$F1 - score = 2\frac{(Precision)(Recall)}{Precision + Recall}$$

$$ROC\ AUC = \int_0^1 \frac{TP}{TP + FN} d\left(\frac{TP}{TP + FP}\right)$$

Borsa tahminindeki ikinci yaklaşım, zaman serisi sınıflandırması olarak da bilinen zaman serisi yön tahminine dayanmaktadır. Borsa yönü sınıflandırması yapmak için, hedef veriler genellikle sırasıyla yukarı ve aşağı hareketleri temsil eden 1 ve 0 olarak etiketlenir. Bu nedenle, tahmin prosedürü bir sınıflandırma problemine dönüşür ve birler ve sıfırlardan oluşan hedef kümesine göre girdiler ışığında bir

sınıflandırıcı model eğitilir. Bu iki ana tahmin yaklaşımının kendine göre avantajları ve dezavantajları vardır. Regresyon tahmininde, tahmin sonuçlarının sınıflandırmaya kıyasla "daha fazla bilgi" sunması için gerçek veri değerleri (sürekli değerler) tahmin edilir. Öte yandan, zaman serisi sınıflandırmasının sonuçlarının yorumlanması, regresyon tahminine göre daha kolaydır. Ayrıca, hedef veri kümesi sınıflandırma tahmininin sürekli hedef kümelerinin aksine Boolean olduğundan, hedef veri ve öznitelik kümesi ilişkili olduğu sürece tahmin modeli için sınıflandırma tahmininin işlenmesi daha kolaydır. Bu çalışmada, borsa fiyat verilerini tahmin etmek için, her iki yaklaşımın avantajlarından yararlanılarak yeni bir iki aşamalı istifli topluluk tahmin modeli tasarlanmıştır.

Önerilen topluluk modelini oluşturmadan önce, Standard ve Poor's 500 Endeksi (SP500), Nikkei 225 Endeksi (NI225), Borsa İstanbul 100 Endeksi (XU100), Kore Bileşik Hisse Fiyat Endeksi (KOSPI), Deutscher Aktien Endeksi (DAX) ve Financial Times Borsa 100 Endeksi (FTSE100) olmak üzere seçilen altı büyük borsa endeksinin günlük kapanış fiyatlarının veri ön işlemesi ve ölçeklendirilmesi, yapılmıştır.

Topluluk modelinin oluşturulmasının ilk adımı olarak, orijinal serinin karmaşıklığını azaltmak için zaman serisi verileri EMD tekniği kullanılarak ayrıştırılmıştır, böylece veri setleri daha yönetilebilir alt bileşenlere (yani IMF'ler) ayrıştırılmıştır. Daha sonra, IMF veri setlerinin her birinin tahmini için deneyler yapılmıştır. ANNR ve ANNC yardımıyla her bir IMF için sırasıyla regresyon ve sınıflandırma tahminleri yapılmıştır. ANNR ve ANNC modelleri, 100 düğümlü ve ReLU aktivasyon fonksiyonu kullanan iki gizli katmana sahiptir. Bu iki model, 30 örnekli küçük-yığın (mini-batch) boyutuyla 300 dönem (epochs) için eğitilmiştir. ANNR'nin çıktı katmanı, hiperbolik teğet aktivasyon fonksiyonu ile sayısal bir değeri tahmin etmek için tek bir düğüme sahiptir ve stokastik gradyan inişinin Adam versiyonunu kullanarak ortalama karesel hata (MSE) kayıp fonksiyonunu en aza indirmek için eğitilmiştir. ANNC ise ikili çapraz entropi kayıp fonksiyonu ile çıkış katmanında sigmoid aktivasyon fonksiyonuna sahiptir. Elde edilen sonuçlar Tablo 5.1'de özetlenmiştir.

Tablo 5.1. Her bir IMF'nin ANNR ve ANNC ile tahmin doğrulukları

Endeks	Model	IMF0	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7
SP500	ANNR	0.7330	**0.8945**	**0.9736**	**0.9744**	**0.9936**	**0.9968**	**0.9960**	**1.0000**
	ANNC	**0.8826**	0.8610	0.8754	0.9169	0.9361	0.7995	0.4657	**1.0000**
NI225	ANNR	0.8248	0.8725	**0.9729**	**0.9926**	**0.9942**	**0.9984**	**0.9959**	**0.9984**
	ANNC	**0.8924**	**0.8915**	0.9441	0.9499	0.9745	0.9573	0.8094	0.9326
XU100	ANNR	0.7894	**0.8991**	**0.9792**	**0.9888**	**0.9920**	**0.9984**	**0.9960**	**0.9976**
	ANNC	**0.8928**	0.8888	0.9240	0.9400	0.9048	0.8776	0.8472	0.9960
KOSPI	ANNR	0.8103	**0.8896**	**0.9657**	**0.9926**	**0.9943**	**0.9943**	**0.9984**	**1.0000**
	ANNC	**0.8840**	0.8856	0.9371	0.9322	0.9175	0.7042	0.4608	**1.0000**
DAX	ANNR	0.8211	0.8847	**0.9833**	**0.9873**	**0.9944**	**0.9976**	**0.9968**	**1.0000**
	ANNC	**0.9039**	**0.8864**	0.9055	0.9436	0.9357	0.9325	0.6958	**1.0000**
FTSE100	ANNR	0.7645	0.8789	**0.9680**	**0.9924**	**0.9933**	**0.9958**	**0.9975**	**0.9975**
	ANNC	**0.8840**	**0.8950**	0.9168	0.9303	0.9529	0.9815	0.6395	0.3782

Tahmin sonuçları, $h_0(t)$ ile gösterilen birinci IMF (IMF0) için sınıflandırma tahminlerinin doğruluğunun, regresyon tahminlerinden önemli ölçüde daha iyi olduğunu göstermektedir. $h_0(t)$ regresyon tahmini için tahmin edilmesi en zor kısım olduğundan, $h_0(t)$ için tahmin doğruluğundaki iyileşmenin orijinal fiyat serisinin tahmin modelinin genel başarısına katkıda bulunabileceği düşünülmektedir. Öte yandan, diğer IMF'lerin ANNR tahmin sonuçları ANNC tahminlerinden biraz daha iyi olmaktadır. Bu nedenle, $h_0(t)$ dışındaki tüm IMF'lerin regresyon tabanlı tahmin modeli ile tahmin edilmesi ve $h_0(t)$'nin sınıflandırma tabanlı tahmin modeli ile tahmin edilmesi, genel tahmin başarısını artırmak için uygun bir yaklaşım olarak değerlendirilmektedir.

Ancak bu aşamada bir sorun ortaya çıkmaktadır. EMD-ANN modelinin tahmin prosedürü, geleneksel olarak, tahmin edilen sürekli değerli IMF'lerin toplamı ile yapılır, böylece IMF'lerin toplu tahminleri, orijinal fiyat serilerinin tahminini temsil etmektedir. Öte yandan, $h_0(t)$ sınıflandırma tahmin sonuçları sigmoid fonksiyon çıktılarıdır ($\sigma(x)$ ile gösterilir ve 0 ile 1 arasında değişen sürekli değerler alır). ANN sınıflandırma modeli, tahminleri $\sigma(x) > 0{,}5$ ise 1 ve $\sigma(x) \leq 0{,}5$ için 0 olarak etiketler. Bu bakımdan, $h_0(t)$'nin sınıflandırma tahminlerini diğer IMF'lerin

regresyon tahminleriyle toplamak anlamsızdır. Bu engeli aşmak için yeni bir deney tasarımı önerilmiştir.

Daha önce açıklandığı gibi, $h_0(t)$ serisi ANNC tarafından tahmin edilmektedir. ANNC, $h_0(t-1), \dots, h_0(t-4)$ girdi olarak beslenecek şekilde $h_0(t)$'nin dört gün gecikmeli değerleri ile eğitilmiş ve $h_0(t)$'nin hareket yönü olarak ifade edilmiştir. Burada $D_t\big(h_0(t)\big)$

$$D_t\big(h_0(t)\big) = \begin{cases} 0, if\ h_0(t) - h_0(t-1) \leq 0 \\ 1, if\ h_0(t) - h_0(t-1) > 0 \end{cases}$$

modelin çıktısı olmaktadır. ANNC ikili bir sınıflandırıcı olduğundan, tahminler, $\hat{C}_{y,t}$, yapay sinir ağının sınıflandırıcı tasarımı tarafından ifade edilen $\hat{C}_{y,t} \in \{0,1\}$ sınıf etiketleridir. ANNC'nin aktivasyon fonksiyonu, makine öğrenmesi terminolojisinde sigmoid fonksiyonu olarak da adlandırılan ve lojistik fonksiyonun özel bir şekli olan $\sigma_t(x)$ ile gösterilen ve 0 ile 1 arasında değerler alan bir fonksiyondur. ANNC, girdileri $\sigma_t(x)$'nin tahmin edilen değerlerine göre sınıflandırır, öyle ki

$$\hat{C}_{y,t} = \begin{cases} 0, if\ \hat{\sigma}_t(x) \leq 0.5 \\ 1, if\ \hat{\sigma}_t(x) > 0.5 \end{cases}$$

şeklindedir.

Son tahmin adımı için ön deneyleri göstermiştir ki, sigmoid fonksiyonunun çıktısı bire veya sıfıra yakın olan tahminler, orijinal zaman serilerinin sırasıyla yukarı ve aşağı hareketlerini daha başarılı bir şekilde tahmin edebilmektedir. Bu gerekçeden hareketle ANNC'nin çıkış değeri olarak $\hat{\sigma}_t(x)$ tutulmuş, başka bir deyişle sınıf etiketi dönüşümü yapılmamıştır. Öte yandan, tüm IMF'ler de ANNR modeli tarafından tahmin edilmektedir ve her bir zaman periyodu için orijinal fiyat serisine ait tahminler, tahmin edilen IMF'lerin toplamı neticesinde elde edilmektedir. $\hat{y}_t$, t zaman dilimindeki tahmin ve $d_t(\hat{y})$ fark serisi olsun, burada

$$d_t(\hat{y}) = \hat{y}_t - \hat{y}_{t-1}$$

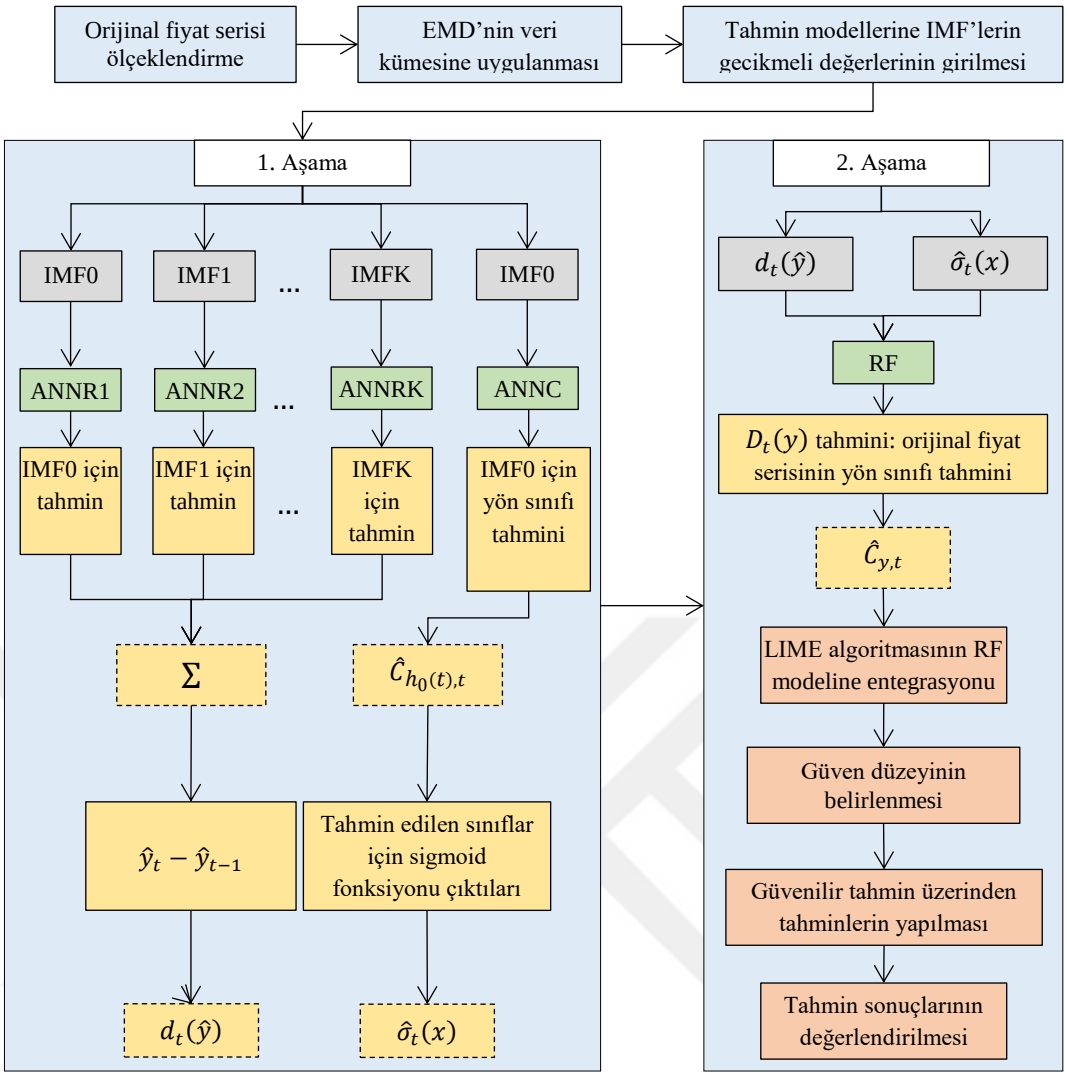
$d_t(\hat{y})$ her bir gün için hesaplanmaktadır. Burada $d_t(\hat{y})$ ANNR'nin tahmin yönü büyüklüğünü ve ardışık tahminlerdeki artış veya azalmanın şiddetini gösterir. Burada $d_t(\hat{y})$'yi elde etmenin arkasındaki motivasyon, $\hat{\sigma}_t(x)$ ile benzerdir. Benzer şekilde, $d_t(\hat{y})$ sıfırdan uzaklaştıkça yukarı ve aşağı hareket yönü tahminlerinin isabet oranlarının artması beklenmektedir. Önerilen modelin birinci aşaması burada sona ermektedir ve bu aşamanın çıktıları, $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$, ikinci aşamanın ML modelinin girdileri olmaktadır.

Özetle, ANNR ve ANNC modelleri eğitim kümesinde eğitilmiş ve bu iki model kullanılarak $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$ elde etmek için doğrulama kümesi için tahminler yapılmış ve böylece birinci aşama tamamlanmıştır. Bu iki modelin çıktıları ikinci aşamadaki tahmin modelini besleyecek girdiler olmaktadır. Bir sonraki aşamada, bir sınıflandırıcı, yani bir rastgele orman modeli, t zaman adımında çıktı olarak orijinal fiyat serisinin hareket yönünü, $D_t(y)$, elde etmek için girdi olarak $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$ ile eğitilmiştir. Burada

$$D_t(y_t) = \begin{cases} 0, if\ y_t - y_{t-1} \leq 0 \\ 1, if\ y_t - y_{t-1} > 0 \end{cases}$$

İkinci aşamanın son tahmin adımında, söz konusu hipotezleri test etmek için test kümesindeki ANNR ve ANNC tahminlerinden sırasıyla $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$ elde edilmiştir. Son olarak, her bir zaman adımında orijinal fiyat serisinin hareket yönünü tahmin etmek için $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$, 250 ağaç sayısına, 37 maksimum ağaç derinliğine sahip rastgele orman sınıflandırıcısına girdi olarak beslenmektedir. Diğer bir deyişle, her t adımı için orijinal fiyat serisinin tahmin edilen hareket yönleri elde edilmiştir. Daha sonra sonuçlar doğruluk, kesinlik (Kesinlik), Duyarlılık, F1 skoru ve ROC-AUC açısından tekli EMD-ANNR tahmin modeli ile karşılaştırılmıştır. Tüm tahmin prosedürü Şekil 5.1' de tasvir edilmektedir.

Özetle, tahmin modelinin son versiyonunda tüm veri kümesi EMD ile ayrıştırılmış, ardından ANNR ve ANNC kullanılarak topluluk modelinin ilk tahmin prosedürü tamamlanmıştır. Son olarak, topluluk modelinin ikinci tahmin prosedürü için rasgele orman kullanılmış ve burada nihai model EMD-ANNRC-RF olarak adlandırılmıştır. İkinci aşamanın son adımı olarak, LIME algoritması EMD-ANNRC-RF'ye entegre edilmiş ve LIME'ın temel modele entegrasyonunun detayları bir sonraki alt bölümde açıklanmıştır.



Şekil 5.1. EMD-ANNRC-RF-LIME modelinin şematik gösterimi

EMD-ANNRC-RF-LIME modelinin tahmin prosedürü iteratif olarak aşağıda verilmiştir:

1. Orijinal veri kümesinin ölçeklendirilmesi.
2. Tüm orijinal veri kümesinim ayrıştırılması ve IMF serisinin edilmesi.
3. IMF veri setlerinin her birinin eğitim, doğrulama ve test setlerine ayrılması.
4. Girdi olarak IMF serisinin n gün gecikmeli değerlerini ve her bir IMF için çıktı olarak en son değeri içeren ANN modelini eğitilmesi. Burada kullanılan ANN modeli, ANN regresyonu için ANNR olarak adlandırılmaktadır.
5. Girdiler olarak IMF0 veri kümesinin n gün gecikmeli değerleri olan $IMF0_{t-1}, \dots, IMF0_{t-n}$ ve IMF0'ın en son günün hareket yönü, $D_t(IMF0)$ ile başka bir ANN modeli eğitilmektedir. Bu aşamada kullanılan ANN bir sınıflandırıcı olmak üzere ANNC olarak adlandırılmaktadır.

6. Tüm tahmin edilen IMF'lerin toplamı ile ANNR yoluyla doğrulama kümesindeki orijinal veri kümesini tahmin edin ve birbirini izleyen her bir t ve t-1 zaman periyodu için orijinal veri kümesi tahminleri arasındaki farkı hesaplayarak her gün için $d_t(\hat{y})$ elde edilir.
7. ANNC ile doğrulama kümesinde IMF0'ın hareket yönünü tahmin edilir. Her sınıflandırmanın her t zaman periyodu için tahmin edilen sigmoid fonksiyon çıktıları $\hat{\sigma}_t(x)$ elde edilir.
8. Birinci aşamada elde edilen $d_t(\hat{y})$ ve $\hat{\sigma}_t(x)$ değerleri, ikinci aşamada $D_t(y)$ kullanılarak RF ile doğrulama kümesinde bir sınıflandırma modeli eğitilir.
9. LIME algoritmasını doğrulama kümesindeki RF modeli nesnesinde önceki adımda açıklandığı gibi aynı giriş/çıkış kurulumuyla çalıştırılır ve LIME algoritması edilir. Güven düzeyi belirlenir ve verilen kriterlere göre hangi RF tahminlerinin bu kriteri yeterince sağladığı tespit edilir, ardından RF'nin tahminleri güven kriterine göre yeterince güvenilir ise tahminler yapılır.

5.1. LIME Algoritmasının EMD-ANNRC-RF Modeli ile Birleşimi

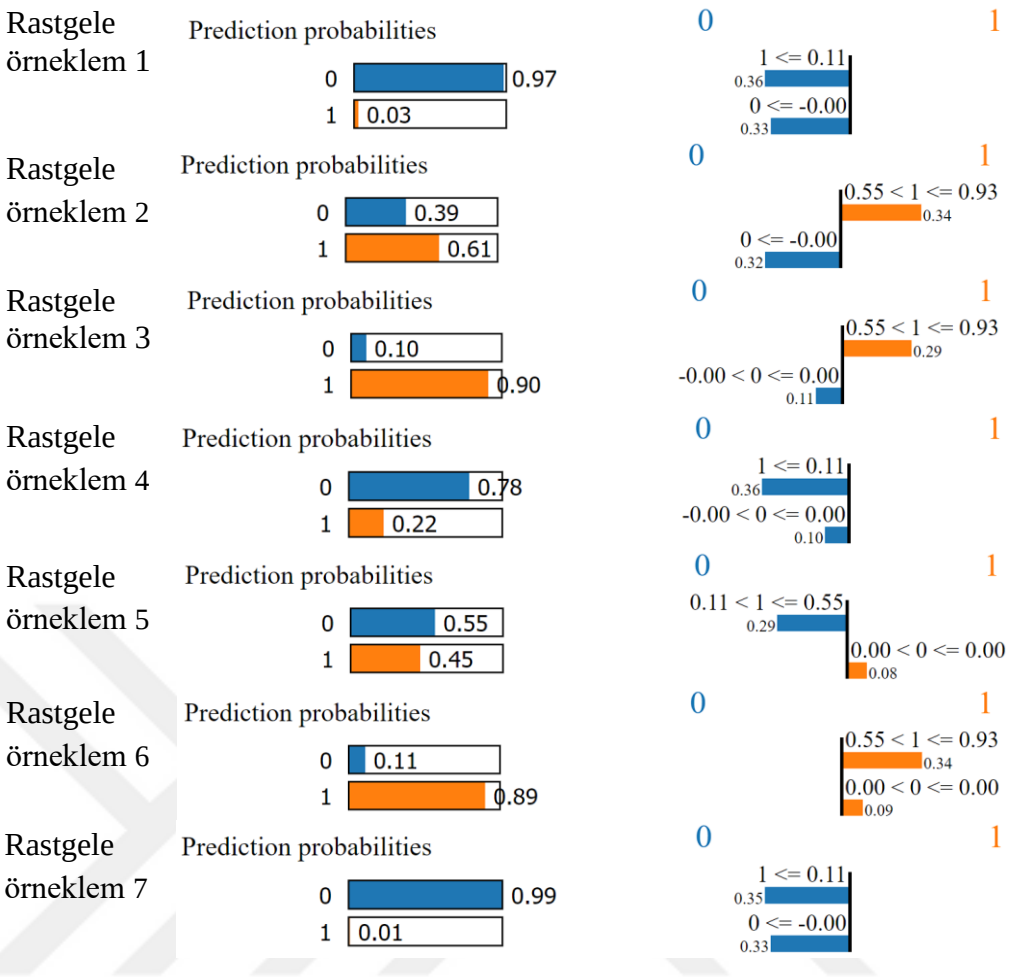
Daha yüksek doğrulukta tahmin, borsa tahmini için hayati öneme sahiptir. Bu nedenle doğruluk oranı yüksek bir tahmin modeli kullanmak daha başarılı sonuçlara olanak sağlayacaktır. Ancak tahmin modelinin doğruluğunun daha fazla artırılamadığı durumlarda, tahminlerin başarısını artırmak için başka bir yaklaşım önerilebilir. Tahminlerin güvenilirliğinin bir model tarafından test edilmesi bir seçenek olarak görünmektedir. Bir tahminin güvenilir olup olmadığına dair geliştirilebilecek herhangi bir yaklaşım, tahminin ne ölçüde güvenilir olduğunu veya hangi olasılıkla gerçekleşebileceğini ölçme çabasıdır.

Her bir tahminin ne kadar güvenilir olduğuna dair bir ölçü olması halinde, gerçekleşme olasılığı görece daha yüksek olan tahminler, karar verici tarafından belirlenen güven düzeyine göre karar vermede kullanılabilir. Öte yandan, karar verici güvenilir olmayan tahminler için (yani, daha düşük gerçekleşme olasılığı olan) karar vermekte çekimser kalacaktır. Bu nedenle, karar vermede yalnızca güvenilirliği yüksek tahminler kullanılacağından, tahminlerin başarısında nihai bir artış beklemek yanlış olmayacaktır. Makine öğrenimi literatüründe bu yaklaşım genel olarak model açıklanabilirliği olarak bilinmektedir. Modelin açıklanabilirliği, tahmin modelinin yorumlanmasına olanak tanıdığı için makine öğrenimi modeline olan güveni artırır. Bir modeli yorumlamanın iki farklı yolu vardır: genel ve yerel. Genel yorum tüm

modeli açıklarken, yerel yorum sadece tahminleri açıklar (Heuillet vd., 2021). Genel yorumlama, modelin tüm davranışını açıklarken yerel yorumlama, modelin tek bir örnek için nasıl karar verdiğini anlamaya ve bireysel tahminleri açıklamaya yardımcı olur. LIME ve SHAP, yerel yorumlama için ortak algoritmalardır. Bu çalışmada, bu amaçla LIME algoritması kullanılmıştır. Model tarafından yapılan her tahmin açıklanarak karar verilirken ilgili ölçümlere başvurulmak istendiğinden, bu çalışmada model açıklanabilirliği için yerel yorumlanabilirlik yaklaşımı kullanılmıştır.

Bu çalışmadaki deney tasarımı iki ana nokta etrafında gelişmiştir. Bunlardan biri, sonuçlarda da görülebileceği gibi, temel modelin tahmin başarısını artırmak, diğeri ise LIME algoritmasını kullanarak tahmin prosedürüne başka bir boyut eklemektir. Bu nedenle, bu çalışmanın ikinci boyutuna göre, LIME algoritması kullanılarak yeni bir tahmin süreci önerilmiştir.

Esasen, $D_t(y_t)$'yi doğrudan tahmin etmek için ANNC'nin sınıf etiketleri yerine $\hat{\sigma}_t(x)$ kullanmanın ve benzer şekilde ANNR'nin doğrudan çıktısı olan $\hat{y}_t$ yerine $d_t(\hat{y})$ kullanmanın önemi bu aşamada ortaya çıkmaktadır. Bir önceki bölümde bahsedildiği gibi ön deneylerde $\hat{\sigma}_t(x)$ ve $D_t(y_t)$ değerleri arasında kısmen de olsa bir paralellik gözlenmiştir. $\hat{\sigma}_t(x)$ değerleri 1 veya 0'a yaklaştıkça, sırasıyla $D_t(y_t)$ 1 veya 0 değerleri için tahminlerin doğruluğu artmaktadır. İsabet oranlarındaki artışın $d_t(\hat{y})$ için de benzer olduğu görülmektedir. $d_t(\hat{y})$değerlerindeki mutlak artış/azalış miktarı arttıkça, sırasıyla $D_t(y_t)$ 1 veya 0 değerleri için yapılan tahminlerin doğruluk oranları artmaktadır. Gözlenen bu olgudan daha sistematik bir şekilde faydalanmak için LIME algoritmasından yararlanılmıştır. Böylece, tahmin modelinin sonuçlarının güvenilirliği, her bir $\hat{\sigma}_t(x)$ ve $d_t(\hat{y})$ değeri için ölçülebilir ve yalnızca güven koşulunu sağlayan tahminler, önceden belirlenmiş belirli bir güven düzeyine göre tahmin yapmak için kullanılmaktadır.



Şekil 5.2. Test kümesi tahminlerine ait örnek LIME açıklamaları.

LIME algoritması tahmin modeline entegre edilerek RF modelinin her bir tahmini test kümesinde açıklanmıştır. LIME tekniği, doğrulama kümesindeki RF modelinin davranışına dayalı olarak bu açıklamaları sağlar. Sonuç olarak, test kümesindeki her bir RF modeli tahmininde, her bir sınıf etiketi için tahmin olasılıkları veya açıklama tahmin olasılıkları (explanation prediction probability (EPP)) hesaplanmıştır. RF'ye herhangi bir girdi örneği verildiğinde, bu tahmin modeli ikili sınıflandırıcı olduğundan, tahmin çıkışı olarak 0 ve 1 değerlerinden birini üretir.

LIME algoritması temel modele entegre edildikten sonra, her bir sınıf etiketi için oluşma olasılığının bir olasılık hesabıyla elde edilmesine olanak sağlamıştır. Şekil 5.2'de yedi farklı rasgele tahmin örneği verilmiştir. Rastgele örnek 1'e göre, sınıf 0 (aşağı doğru tahmin) için LIME, 0,97 tahmin olasılığı hesaplarken, sınıf 1 (yukarı tahmin) için 0,03 değeri hesaplanmaktadır. Karşılık gelen girdi değerleri, şeklin sağ tarafında görülebilir. Buna göre 0,97 olasılıkla aşağı yönlü hareket veya 0,03 olasılıkla yukarı yönlü hareket oluşması beklenmektedir. Bu durumda, karar

verici önceden güven düzeyi olarak 0,70 değerini belirlerse, aşağı doğru tahmin olasılığı (0,97) güven düzeyinden büyük veya ona eşit olduğu için rasgele örneklem 1'deki tahmine güvenecektir. Rastgele örneklem 2 için, 0 ve 1 sınıfının tahmin olasılıkları sırasıyla 0,39 ve 0,61'dir. Rastgele örneklem 2'de ise her iki sınıfın tahmin olasılığı 0,70'den küçük olduğu için karar verici tahmin yapmaktan çekinecek ve karar vermekten kaçınacaktır.

6. BULGU VE DEĞERLENDİRMELER

6.1. Veri kümeleri ve deney ortamı

Bu çalışmadaki tüm deneyler Python 3.9'da gerçekleştirilmiştir. Python kütüphanelerinden Pandas (Reback vd., 2020), NumPy (Harris vd., 2020), PyEMD (*PyEMD's documentation — PyEMD 0.2.13 documentation*, t.y.), Scikit-learn (Pedregosa vd., t.y.), Keras (*Keras*, 2015/2022), Matplotlib (Caswell vd., 2022), xgboost (*XGBoost | Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, t.y.), lime (*Local Interpretable Model-Agnostic Explanations (lime) — lime 0.1 documentation*, t.y.) kullanılmıştır. Deneyler için her bir veri kümesinde on yıllık tarihsel veriler (2012-2021) kullanılmıştır. Borsa endekslerinin her birinin test dönemlerini görselleştirmek için Şekil 6.1'de günlük kapanış fiyatları gösterilmiştir. Ayrıca deneyleri yapmadan önce, her veri kümesi maksimum-minimum yöntemiyle ölçeklendirilmiştir, böylece;

$$s(y_t) = \frac{y_t - min(\,y_t)}{max(\,y_t) - min(\,y_t)}$$

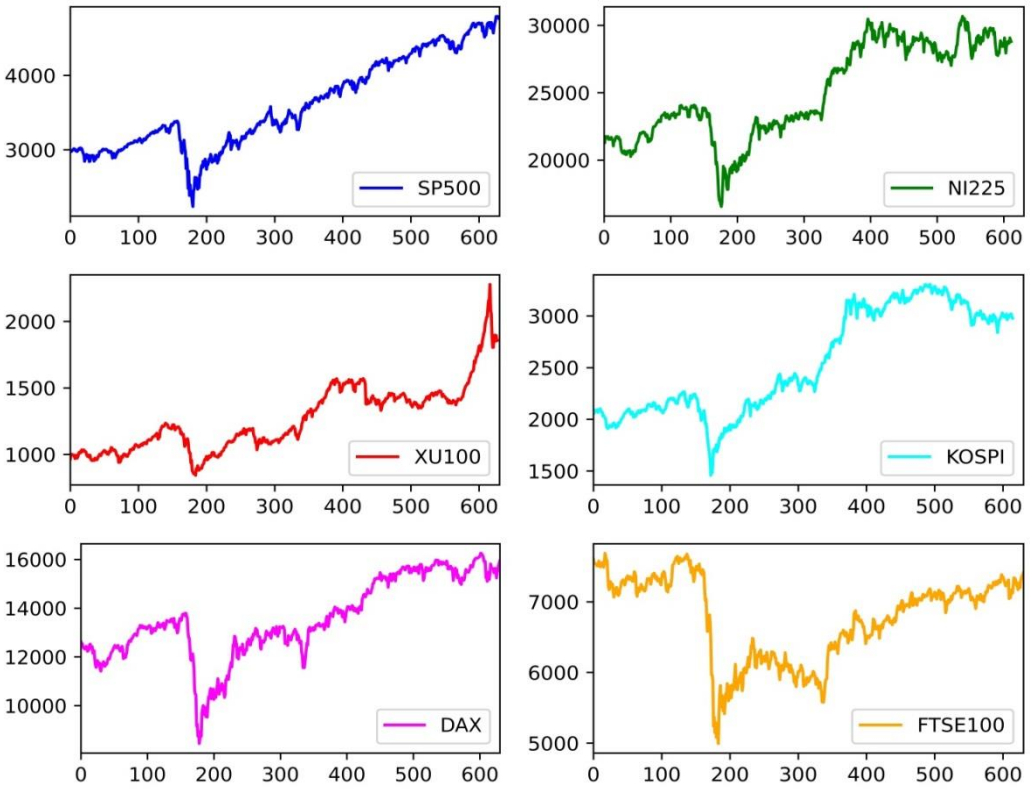
Burada $s(y_t)$, ölçeklendirilmiş veri kümesini; $min(\,y_t)$, y_t'nin minimum değerini; $max(\,y_t)$, y_t'nin maksimum değerini temsil etmektedir. Veri normalizasyonu, makine öğrenimi ve derin öğrenme görevleri için yaygın bir uygulamadır çünkü ölçeklendirilmiş verilerle tahmin modelleri daha verimli çalışabilir.

Tablo 6.1. Deney verilerine ait özet bilgiler

Endeks	Tarih Aralıkları	Toplam Veri	Eğitim Kümesi	Doğrulama Kümesi	Test Kümesi
SP500	2012-01-03 ~ 2021-12-31	2517	1261	625	626
NI225	2012-01-04 ~ 2021-12-30	2445	1224	608	608
XU100	2012-01-02 ~ 2021-12-31	2512	1258	624	625
KOSPI	2012-01-02 ~ 2021-12-30	2460	1232	611	612
DAX	2012-01-02 ~ 2021-12-30	2530	1267	629	629
FTSE100	2012-01-04 ~ 2021-12-31	2529	1267	628	629

SP500, NI225, XU100, KOSPI, DAX ve FTSE100 olmak üzere altı farklı hisse senedi endeksi üzerinde deneyler yapılmıştır. Bir sonraki gün için hisse senedi endekslerinin yukarı ve aşağı hareket yönünü eğitmek ve tahmin etmek amacıyla,

tüm veri kümesi sırasıyla eğitim, doğrulama ve test setlerinin %50, %25 ve %25 alt kümelerine bölünmüştür. İlk aşama'da tahmin için IMF'lerin her birinin dört gün gecikmeli değerleri girdi olarak kullanıldığından, dört gün veri eksiktir. Ayrıca, ANNR tahminlerinden elde edilen fark serisi, $d_t(\hat{y})$, kullanıldığından bir gün daha kayıp yaşanır. Bu durum sonuç olarak toplamda beş gün veri kaybına neden olmaktadır. Tüm veri setleri ve alt kümelerinin (eğitim kümesi, doğrulama kümesi ve test kümesi) tarih aralıkları Tablo 6.1'de verilmiştir.



Şekil 6.1. Tahmin edilen endekslerin test periyotlarının grafiği

Piyasa takvimi ülkeler arasında farklılık gösterdiğinden, veri setlerinin toplam uzunlukları ve başlangıç-bitiş tarihleri arasında önemsiz farklılıklar oluşur. Tüm deneysel veri setleri tradingview web sitesinden (https://tr.tradingview.com/ (2023, Kasım 11)) indirilmiştir.

6.2. Deney bulguları ve nihai çıkarımlar

EMD-ANNR, EMD-ANNRC-XGBoost ve EMD-ANNRC-RF modellerinin tüm doğruluk sonuçları, her bir hisse senedi endeksi için Tablo 6.2'de sunulmuştur. Elde edilen sonuçlar, önerilen iki aşamalı ensemble tahmin modelinin performans değerlendirme metriklerine göre EMD-ANNR'den önemli ölçüde daha iyi olduğunu ortaya koymaktadır. Ayrıca, EMD-ANNRC-RF'nin tüm durumlarda EMD-ANNRC-

XGBoost'tan biraz daha iyi olduğu görülmektedir. Bu durumda, XGBoost algoritmasının RF'ye göre daha fazla hiper parametreye sahip olduğu ve dolayısıyla daha iyi ayarlanması gerektiği söylenebilir[8].

Tablo 6.2. Tahmin modellerinin karşılaştırılması

		SP500	NI225	XU100	KOSPI	DAX	FTSE100
EMD-ANNR	İsabet	0.6784	0.6903	0.7352	0.7328	0.6529	0.6476
	Kesinlik	0.7112	0.6979	0.7519	0.7715	0.6715	0.6636
	Duyarlılık	0.7409	0.7241	0.8122	0.7514	0.6875	0.6909
	F1-score	0.7258	0.7108	0.7809	0.7613	0.6794	0.6770
	ROC-AUC	0.6675	0.6885	0.7203	0.7299	0.6503	0.6443
EMD-ANNRC-XGBoost	İsabet	0.7796	0.7878	0.7872	0.7810	0.7727	0.7727
	Kesinlik	0.8101	0.8065	0.8142	0.8257	0.7898	0.7699
	Duyarlılık	0.8056	0.7837	0.8209	0.7781	0.7827	0.8138
	F1-score	0.8078	0.7949	0.8176	0.8012	0.7862	0.7912
	ROC-AUC	0.7750	0.7880	0.7807	0.7815	0.7719	0.7701
EMD-ANNRC-RF	İsabet	**0.7987**	**0.8010**	**0.7888**	**0.7925**	**0.7806**	**0.7838**
	Kesinlik	0.8287	0.8133	0.8113	0.8313	0.7929	0.8050
	Duyarlılık	0.8194	0.8056	0.8292	0.7954	0.7976	0.7808
	F1-score	0.8240	0.8094	0.8202	0.8130	0.7953	0.7927
	ROC-AUC	0.7951	0.8007	0.7810	0.7920	0.7794	0.7840

En son teknik ve yaklaşımları öneren çeşitli çalışmaların listelendiği Tablo 6.3'de, temel tahmin modelimiz (EMD-ANNRC-RF) ile karşılaştırmalar yapılmıştır. Bu çalışmaların tahmin doğrulukları yaklaşık olarak %0,60 ile %0,90 arasında değişmektedir. Bu nedenle, temel tahmin modelimizin mevcut tekniklerle yaklaşık olarak benzer sonuçlar ürettiği söylenebilmektedir.

[8] Kütüphanenin varsayılan hiper parametreleri kullanılmıştır.

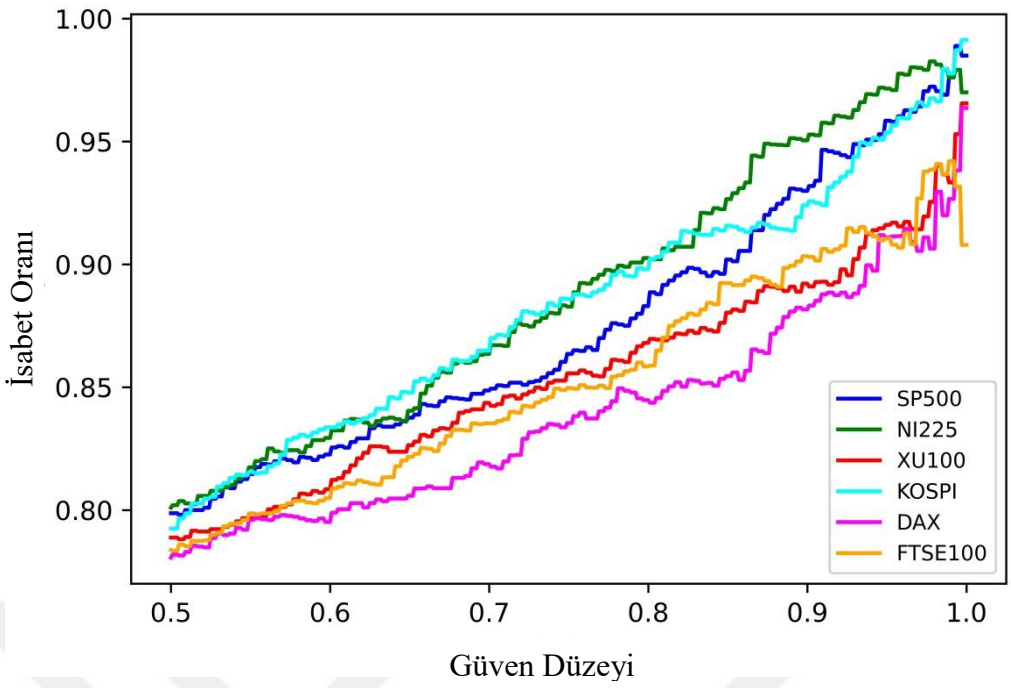
Tablo 6.3. Literatürdeki bazı güncel çalışmaların tahmin performansı özeti.

Yazar(lar)	Yöntemler	Veri Kümesi	İsabet (%)
Bisoi vd. (2019)	VMD, evolutionary robust kernel extreme learning machine (RKELM)	FTSE	85.83
Börjesson ve Singull (2020)	Causal dilated convolutional neural networks	S&P 500	74.96
Chung ve Shin (2020)	Genetic algorithms, multi-channel convolutional neural network	KOSPI Indices	73.74
Ghorbani ve Chong (2020)	Covariance information based , principal component analysis	ABD merkezli 50 şirket hissesi	80
Long vd. (2020)	Deep neural network, knowledge graph	CITIC Securities	73.59
Niu vd. (2020)	VMD, LSTM	S&P 500	85.83
Thakkar ve Chaudhari (2020)	Term frequency–inverse document frequency, neural networks	S&P 500	75
Yu ve Yan (2020)	Phase-space reconstruction, LSTM	DJIA	61.51
Gunduz (2021)	Variational auto-encoders , LSTM	BIST 30	68.5
Thakkar vd. (2021)	Pearson Correlation Coefficient-based Neural Network	HDFC hissesi	78.48
Yin vd. (2021)	Optimized random forest	Integrated Electronics Corporation hissesi	89
Yun vd. (2021)	GA-XGBoost	KOSPI	93.82
Agrawal vd. (2022)	LSTM	HDFC	65.64

Chandar (2022)	Convolutional neural network	Bank of America hissesi	89.6
Lee vd. (2022)	Attention-based BiLSTM	TPE0050	68.83
Yang vd. (2022)	Group penalized logistic regressions	Amazon hissesi	73.2

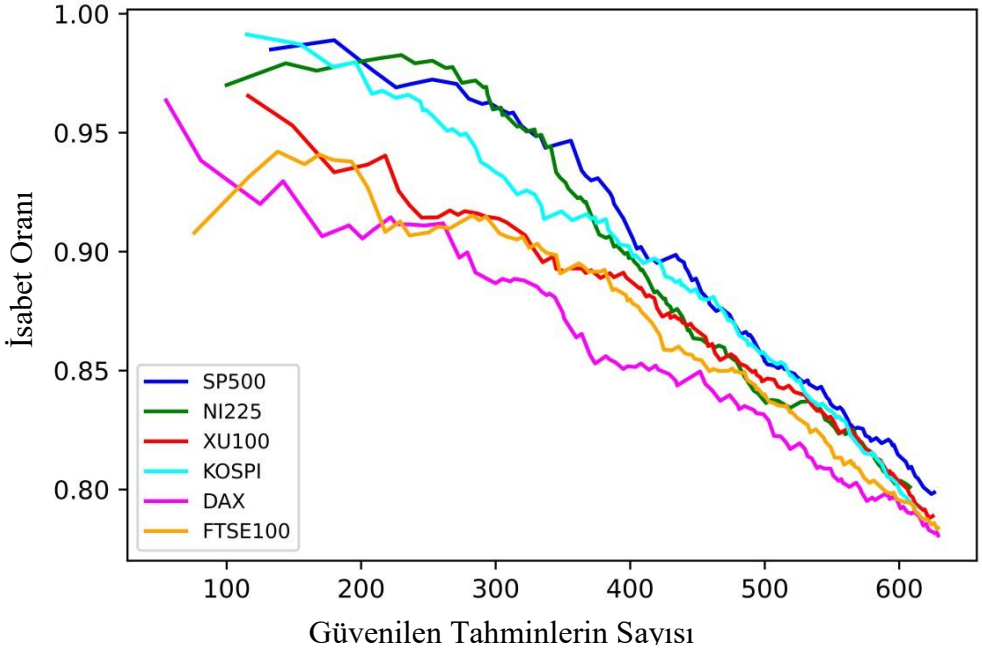
LIME algoritmasının EMD-ANNRC-RF modeline uygulanmasıyla elde edilen bulgular aşağıda tartışılmış ve elde edilen model "iki aşamalı EMD-ANNRC-RF-LIME" olarak adlandırılmıştır. Tahmin modeline LIME algoritmasının entegre edilmesi, karar vericinin yalnızca güven koşulunu sağlayan tahminlere güvenebilmesini mümkün kılmaktadır. Bu nedenle, bu aşamada güven düzeyinin değeri belirlenmelidir. Ardından, belirlenen güven düzeyine göre test dönemi boyunca tahminler yapılmakta ve bu tahminlerin hangi günler için yapılacağı ve ne ölçüde doğru olduğu hesaplanmaktadır. Ayrıca, güven düzeyi arttıkça, doğruluk oranında bir artış olup olmadığı test edilmelidir. Şekil 6.2'de, altı farklı pazar endeksinin test kümeleri için güven düzeyi (yatay eksen) ve İsabet oranı (dikey eksen) ilişkisi grafiğe çizilmiştir.

Güven düzeyi, 0.5'ten başlayarak 1'e kadar değişmektedir. Güven düzeyinin 0.5 olması halinde LIME tekniğini kullanmadığımız, diğer bir deyişle doğrudan RF'nin yapmış olduğu bütün tahminlere güvendiğimiz (EMD-ANNRC-RF modelinin tahminleri), duruma tekabül etmektedir. Güven düzeyini 1'e doğru yavaşça artırarak, karar verici basitçe modelden daha güvenilir tahminler elde etmek isteğini ortaya koymaktadır. Daha önce de belirtildiği gibi, güvenilir tahminlerin sayısı ile güven arasındaki denge nedeniyle, tüm tahmin süresi için güvenilir tahminler elde etmek beklenemez. Ancak, güven düzeyinin artmasıyla güvenilen tahmin sayısı giderek azalsa da doğruluk tahmin oranının da arttığı gözlemlenmektedir. Tüm veri kümeleri için, güven düzeyinin artması genel olarak isabet oranlarını küçük sapmalarla artırmaktadır.



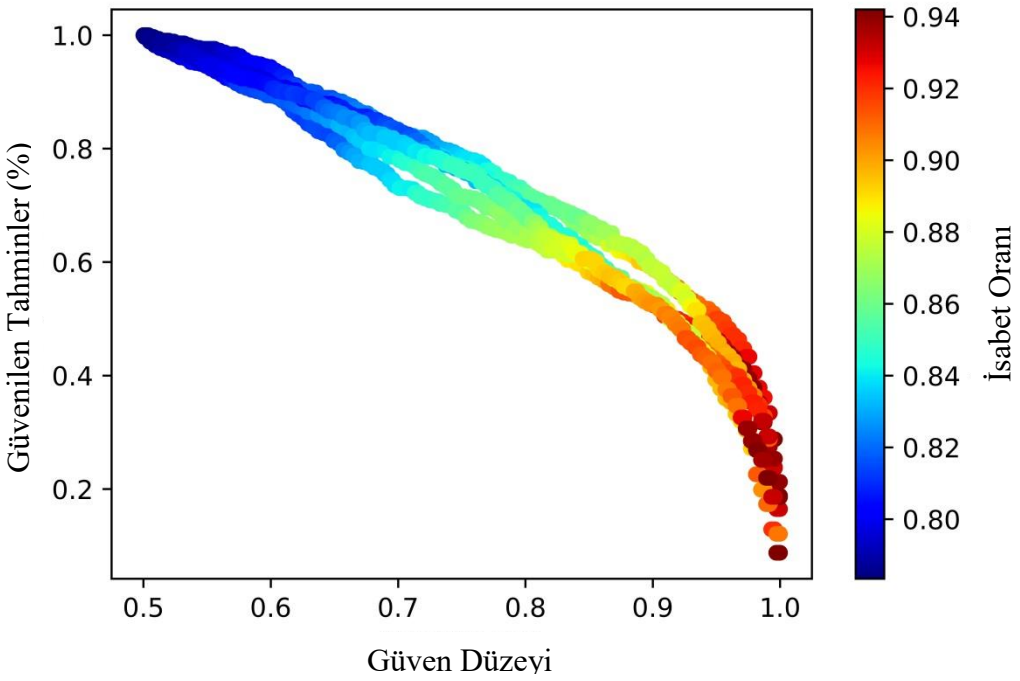
Şekil 6.2. Test kümelerindeki güven seviyelerine göre isabet oranları.

Güvenilir tahminlerin yapılabileceği gün sayısı ve buna karşılık gelen İsabet oranları Şekil 6.3'de gösterilmektedir. Normatif olarak, yalnızca tahmin edilen günler isabet oranlarının hesaplanmasına dahil edilir. Bu nedenle, her bir sınıf etiketi için hesaplanan açıklama tahmin olasılıkları değerinden herhangi biri belirlenen güven düzeyinin üzerinde değilse, tahmin yapılmaz ve o gün için TP, TN, FP ve FN değerleri oluşmaz.



Şekil 6.3. Güvenilir tahmin sayılarına karşılık gelen isabet oranları.

Güvenilir bir tahminin yapılabileceği gün sayısı ve buna karşılık gelen isabet oranları Şekil 6.3'de gösterilmektedir. Doğal olarak, sadece güvenilir tahminler doğruluk oranlarının hesaplanmasına dahil edilir. Bu nedenle, her bir sınıf etiketi için hesaplanan açıklama tahmin olasılıkları değerinden herhangi biri belirlenen güven düzeyinin üzerinde değilse, model tahmin yapmaktan çekinir ve o gün için TP, TN, FP veya FN değerlerinden hiçbiri oluşmaz. Test dönemi boyunca yapılan tahminlerin doğruluk oranları, alt kısımda 629 güvenilir tahmin için 0,7806 (DAX) (alt sınır) ve 608 güvenilir tahmin için 0,8010 (NI225) (üst sınır) arasında değişmektedir. Şekil 6.3'e göre 76 güvenilir tahmin için 0,9079 (FTSE100) (alt sınır) ve 115 güvenilir tahmin için 0,9913 (KOSPI) (üst sınır) aralığında değişmektedir. Güven düzeyi arttıkça, beklenildiği gibi güven koşulunu karşılayan tahminlerin sayısı azalır. Sonuç olarak, güvenilir tahminlerin sayısındaki artış, dolaylı olarak güven düzeyi ile isabet oranı arasındaki ilişki sonucunda isabet oranını artırır.



Şekil 6.4. Güven düzeyi ve isabet oranına göre güvenilir tahminlerin yüzdesi.

Şekil 6.4'de görüldüğü üzere, altı hisse senedi piyasası endeksi için test kümelerinde güvenilir tahminlerin oranı, güven düzeyi arttıkça azalırken, isabet oranı ise artmaktadır. Öte yandan, güven düzeyi yaklaşık olarak 0.90 seviyesini aştığında, ilişkinin lineer yapısı bozulmaya başlamakta ve güvenilir tahminlerin oranı büyük ölçüde azalırken, ortalama olarak doğruluk oranında eşdeğer bir artış gözlenmemektedir.

EMD-ANNRC-RF-LIME modelinin deneysel sonuçları Tablo 6.4'te özetlenmiştir. Güven düzeyinin %50, %85, %95 ve %100'e eşit veya daha büyük olduğu durumlar için isabet oranları ve buna karşılık gelen güvenilir tahmin sayıları Tablo 6.4'te yer almaktadır. Dikkat edilmesi gereken nokta, Güven ≥ 0,50 için işlem günlerinin sayısının tüm endeksler için test dönemi uzunluğuna eşit olmasıdır ve bu, herhangi bir güven düzeyinin belirlenmemesi durumuna eşdeğer bir işlemdir. Diğer bir deyişle, EMD-ANNRC-RF modelini kullanarak tüm test kümesini tahmin etmek olarak yorumlanabilir. EMD-ANNRC-RF-LIME'nin en yüksek doğruluğu, KOSPI veri kümesinde 155 tahmin yapılarak Güven = 1 için 0,9913 isabet elde edilmiştir. Öte yandan, en düşük isabet FTSE100 endeksinde, Güven = 1 için 76 tahmin yapılan durumda 0,9079 olarak gözlemlenmektedir.

Tablo 6.4. EMD-ANNRC-RF-LIME tahmin sonuçları.

Endeks	Güven = 1		Güven ≥ 0.95		Güven ≥ 0.85		Güven ≥ 0.50	
	İsabet	Güvenilir Tahminler	İsabet	Güvenilir Tahminler	İsabet	Güvenilir Tahminler	İsabet	Güvenilir Tahminler
SP500	0.985	133	0.958	313	0.902	408	0.798	626
NI225	0.970	100	0.971	281	0.926	354	0.801	608
XU100	0.965	116	0.916	286	0.880	418	0.788	625
KOSPI	0.991	115	0.953	259	0.915	368	0.792	612
DAX	0.963	55	0.910	247	0.854	385	0.780	629
FTSE100	0.907	76	0.910	257	0.892	381	0.783	629

Burada dikkat çekilmesi gereken bir nokta bulunmaktadır. Önerilen EMD-ANNRC-RF modelinin altı farklı veri kümesi için tahmin doğrulukları birbirine yakın olsa da, güven düzeyi arttıkça altı veri kümesinin isabet oranları farklılaşmaya başlamaktadır (Şekil 6.3 ve Şekil 6.4'e bakınız).

7. TARTIŞMALAR

Bu çalışmanın temel yaklaşımı, bir topluluk sınıflandırıcı modelinin her bir tahminini LIME algoritmasıyla açıklamaktır. Kısacası, daha güvenilir olan tahminler belirlenir ve sadece bu tahminlerin karar sürecinde dikkate alınması sağlanır. Bu bağlamda, literatüre katkısını açıkça göstermek için, yaklaşık benzer başarılı tahminler yapabilen çağdaş çalışmalara örnek olabilecek bir temel model oluşturulmuştur. Ardından, LIME algoritmasının temel modele entegre edilmesiyle elde edilen iyileştirme ortaya çıkarılmıştır. Daha kesin olmak gerekirse, temel model EMD-ANNRC-RF ile altı farklı sermaye piyasası endeksinde ortalama %0.7909 isabet oranına ulaşılmıştır (Tablo 6.2'ye bakınız). Daha sonra, EMD-ANNRC-RF-LIME modeliyle tahmin güven düzeyi kavramı tanıtılmış ve tahmin süreci, daha yüksek güven düzeylerine sahip tahminlerle devam ettirilmiştir.

Bu bağlamda, karar verici, hangi tahminlerin daha güvenilir olduğunu belirleyebilmektedir. Tablo 6.4'te, karşılık gelen güven düzeyleri için olası sonuçları özetlemek amacıyla %50, %85, %95 ve %100 güven düzeylerine ulaşılan sonuçları yer almaktadır. Ayrıca, %50-100 aralığındaki tüm güven düzeyleri için elde edilen sonuçlar Şekil 6.4'de gösterilmektedir. Güven düzeyinin %50 olarak belirlendiği noktada, karar verici temel tahmin modeli (EMD-ANNRC-RF) tarafından yapılan tahminlerin tamanına güvenmekedir.

Diğer yandan, güven düzeyini artırdıkça, LIME algoritmasının öne çıktığı, EMD-ANNRC-RF modelinin tahminlerinin geliştirildiği gözlemlenmiştir. Bu kavramı daha iyi anlamak için, SP500 endeksinde güvenilirliği %85 ve üstünde olan her bir tahminin yürütülmesi, bu gereksinimi karşılamayanları ortadan kaldırmaktadır. Böylece, toplam 626 günlük test süresinin 408 günü için sadece tahminler yapılabilmekedir. Bu nedenle, test kümesinin 218 günü için yeterli güven sağlanamadığından bazı tahminlerden kaçınılmıştır. Sonuç olarak, SP500 endeksi için test kümesinde 408 tahmin için %0.9020 doğruluk oranı elde edilmiştir.

Önerilen model için altı farklı hisse senedi piyasası endeksinde en yüksek %100 güven düzeyi belirlendiğinde, ortalama tahmin edilen gün sayısı yaklaşık olarak 622'den 100'e (yaklaşık %84 azalma) düşmüş, ortalama doğruluk oranı ise %21.87 artarak 0.9639'a ulaşmıştır. Bu nedenle, bu yaklaşımın güçlü tarafı daha başarılı tahminler yapılabilmesine olanak sağlamasıdır ancak zayıf yönü ise tüm tahmin süreci için tahmin yapmaya izin vermemesidir.

Literatürde ensemble deneysel mod ayrıştırma (EEMD) ve VMD gibi daha güncel deneysel mod ayrıştırma yöntemleri bulunmakla birlikte, EMD deneylerimizde başarılı sonuçlar sağlamıştır. VMD'nin EMD'ye göre parametre ayarlaması gerektirdiği için EMD'nin uygulanması daha kolaydır. Bununla birlikte, Niu vd. (2020), VMD'nin hisse senedi piyasası yön tahmininde EMD'ye göre daha başarılı olduğunu göstermiştir.

Yapay sinir ağları, zaman serilerini tahmin etmede yeterince başarılı olsa da, LSTM'nin daha başarılı olduğu gösterilmiştir (Wu vd., 2020). Gelecekteki çalışmalarda, LSTM algoritmasının ince ayarlı hiper parametrelerle daha iyi araştırılmasının faydalı olabileceği düşünülmektedir. Aynı çıkarım XGBoost modeli için de geçerlidir. Yun vd. (2021) gibi çalışmalar, XGBoost modelinin RF'den daha başarılı sonuçlar üretebileceğini göstermiştir. Ensemble modelimizin ikinci aşamasında RF'nin tercih edilme nedeni, doğruluk oranları açısından RF'nin tatmin edici sonuçlar üretmesidir. Ayrıca RF, XGBoost'a kıyasla daha az hiper parametreye sahip olması nedeniyle kullanımı daha basittir.

Deneysel mod ayrıştırması ile ilgili dikkate alınması gereken bir diğer husus ise, bu çalışmada ve mevcut literatürde tüm veri kümesinin (eğitim, doğrulama ve test) önceden ayrıştırılması gerekmektedir. Dolayısıyla, herhangi bir makine öğrenimi tekniğini uygulamadan önce, eğitim kümesiyle birlikte doğrulama ve test kümesinin de ayrıştırılması gerekmektedir. Ancak uygulamada, sadece gözlenen veriler ayrıştırılabilir ve alt bileşenlerin bir sonraki gün değerleri tahmin edilebilir. Daha kesin olmak gerekirse, tüm zaman serisinin gözlenen değerlerini $(t-k, t-k+1, \ldots, t)$ olarak gösterirsek $(t-k, t-k+1, \ldots, t)$ serisi ayrıştırıldıktan sonra t+1 zamanındaki alt bileşen değerleri tahmin edilebilir. Sonrasında $(t-k+1, t-k+2, \ldots, t+1)$ serisi yeni gözlenen orijinal zaman serisi haline gelir, bu dizi ayrıştırılır ve $t+2$ zamanındaki ayrıştırılmış alt bileşen değerleri tahmin edilir ve tahmin süreci bu şekilde devam eder. Bu tahmin prosedürüne kayan pencere (*sliding window*) adı verilir. Kısacası, her pencere için bu işlemler ardışık olarak gerçekleştirilir ve tahminler yapılır. Veri ayıştırması ile birlikte yapılan finansal zaman serisi tahmin modellerinde özellikle uygulayıcılara rehberlik etmesi adına bu tür bir deney tasarımının geliştirilmesi önerilmektedir.

8. SONUÇ VE ÖNERİLER

Literatürde, LIME algoritması, doğrudan günlük yön tahmini yapmak için makine öğrenimi teknikleriyle entegre bir şekilde kullanıldığına rastlanmamıştır. Bu nedenle, LIME'ın bu şekliyle entegre edilmesinin literatüre orijinal bir katkı sağladığı düşünülmektedir. LIME algoritmasının modelin "güvenilmez" tahminlerini önleme yeteneği ve daha az sayıda ancak daha güvenilir tahminlere izin vermesi sayesinde, önerdiğimiz EMD-ANNRC-RF-LIME çerçevesinin belirgin bir şekilde başarılı olduğu kanıtlanmıştır ancak EMD-ANNRC-RF-LIME modelinin daha yüksek güven düzeyleri için sağlamlık (robustness) açısından iyileştirme gerektirebileceği söylenebilir.

Bu modelin, sermaye piyasalarında yatırımcılar için faydalı bir karar verme aracı olarak kullanılacağı düşünülmektedir. Bir piyasa katılımcısının günlük ticaret rutininde genellikle birden fazla fiyat serisi tahmin edilmektedir. Bu nedenle, aynı dönemde birden fazla fiyat serisi tahmin edildiğinde, bazı tahminler bazı varlıklar için kaçınılacakken diğer tahminler yerine getirilecektir. Bu nedenle, tek bir varlığın fiyatını tahmin etmek yerine, aynı anda birden fazla fiyat serisi için tahminler yapmak ve güven koşulunu sağlayan varlıklara güvenmek mümkün olacaktır. Böylece, her tahmin döneminde güven koşulunu karşılayan belli bir sayıda varlık için yön tahminleri yapılırken, güven koşulunu karşılamayan tahminlerden kaçınılacak ve tahmin edilen hisse senetleri periyodik olarak değişecektir.

Münferit hisse senedi fiyatları arasındaki doğal ilişki sonucunda, çoğu hisse senedi piyasa koşullarına bağlı olarak aynı veya ters yönde hareket edebilmektedir. Elde edilen bulgulara göre, ters yönlü korelasyona sahip hisse senetlerinin spekülatif potansiyeli kullanılarak, portföy çeşitlendirmesini mümkün kılan yeni bir perspektif ortaya çıkmaktadır. Bu nedenle, ileri çalışmalar için çok sayıda varlığın yer aldığı büyük bir veri kümesi üzerinde aynı anda deney yapılması önerilmektedir.

KAYNAKÇA

Bisoi, R., Dash, P. K., ve Parida, A. K. (2019). Hybrid Variational Mode Decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis. *Applied Soft Computing*, *74*, 652-678. https://doi.org/10.1016/j.asoc.2018.11.008

Bouveyron, C., Celeux, G., Murphy, T. B., ve Raftery, A. E. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge University Press. https://doi.org/10.1017/9781108644181

Cadzow, J. A. (1994). Least Squares, Modeling, and Signal Processing. Digital Signal Processing, 4(1), 2–20. https://doi.org/10.1006/dspr.1994.1002

Caswell, T. A., Droettboom, M., Lee, A., De Andrade, E. S., Hoffmann, T., Klymak, J., Hunter, J., Firing, E., Stansby, D., Varoquaux, N., Nielsen, J. H., Root, B., May, R., Elson, P., Seppänen, J. K., Dale, D., Jae-Joon Lee, McDougall, D., Straw, A., … Ivanov, P. (2022). *matplotlib/matplotlib: REL: v3.5.2* (v3.5.2) [Software]. Zenodo. https://doi.org/10.5281/ZENODO.6513224

Das Gupta, S. (1980). Discriminant Analysis. Içinde S. E. Fienberg ve D. V. Hinkley (Ed.), *R.A. Fisher: An Appreciation* (ss. 161-170). Springer. https://doi.org/10.1007/978-1-4612-6079-0_16

Dash, R., Samal, S., Dash, R., ve Rautray, R. (2019). An integrated TOPSIS crow search based classifier ensemble: In application to stock index price movement prediction. *Applied Soft Computing*, *85*, 105784. https://doi.org/10.1016/j.asoc.2019.105784

Dragomiretskiy, K., ve Zosso, D. (2014). Variational Mode Decomposition. *IEEE Transactions on Signal Processing*, *62*(3), 531-544. https://doi.org/10.1109/TSP.2013.2288675

Fix, E., ve Hodges, J. L. (1989). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. International Statistical Review / Revue Internationale de Statistique, 57(3), 238–247. https://doi.org/10.2307/1403797

Ghorbani, M., ve Chong, E. K. P. (2020). Stock price prediction using principal components. *PLOS ONE*, *15*(3), e0230124. https://doi.org/10.1371/journal.pone.0230124

Gunduz, H. (2021). An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination. *Financial Innovation*, *7*(1), 28. https://doi.org/10.1186/s40854-021-00243-3

Hao, Y., ve Gao, Q. (2020). Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning. *Applied Sciences*, *10*(11), 3961. https://doi.org/10.3390/app10113961

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), Article 7825. https://doi.org/10.1038/s41586-020-2649-2

Heuillet, A., Couthouis, F., ve Díaz-Rodríguez, N. (2021). Explainability in deep reinforcement learning. *Knowledge-Based Systems*, *214*, 106685. https://doi.org/10.1016/j.knosys.2020.106685

Holmes, M. H. (2013). Introduction to Perturbation Methods (Vol. 20). Springer. https://doi.org/10.1007/978-1-4614-5477-9

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., ve Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903-995. https://doi.org/10.1098/rspa.1998.0193

Ismail, M. S., Md Noorani, M. S., Ismail, M., Abdul Razak, F., ve Alias, M. A. (2020). Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur Stock Exchange. *Applied Soft Computing*, *93*, 106422. https://doi.org/10.1016/j.asoc.2020.106422

Jebara, T. (2012). *Machine Learning: Discriminative and Generative*. Springer Science ve Business Media.

Jin, Z., Yang, Y., ve Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, *32*(13), 9713-9729. https://doi.org/10.1007/s00521-019-04504-2

Katz, V. J. (1998). A History of Mathematics: An Introduction. Addison-Wesley.

Keras: Deep Learning for humans. (2022). [Python]. Keras. https://github.com/keras-team/keras (Original work published 2015)

Lee, M.-C., Chang, J.-W., Yeh, S.-C., Chia, T.-L., Liao, J.-S., ve Chen, X.-M. (2022). Applying attention-based BiLSTM and technical indicators in the design and performance analysis of stock trading strategies. *Neural Computing and Applications*, *34*(16), 13267-13279. https://doi.org/10.1007/s00521-021-06828-4

Liu, H., ve Long, Z. (2020). An improved deep learning model for predicting stock market price time series. *Digital Signal Processing*, *102*, 102741. https://doi.org/10.1016/j.dsp.2020.102741

Liu, S., Zhang, X., Wang, Y., ve Feng, G. (2020). Recurrent convolutional neural kernel model for stock price movement prediction. *PLOS ONE*, *15*(6), e0234206. https://doi.org/10.1371/journal.pone.0234206

Local Interpretable Model-Agnostic Explanations (lime)—Lime 0.1 documentation. (t.y.). Geliş tarihi 08 Ekim 2022, gönderen https://lime-ml.readthedocs.io/en/latest/index.html

Lundberg, S. M., ve Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, *7*(1), 77-91. https://doi.org/10.2307/2975974

McCulloch, W. S., ve Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133. https://doi.org/10.1007/BF02478259

Mirkin, B. (1996). *Mathematical Classification and Clustering* (C. 11). Springer US. https://doi.org/10.1007/978-1-4613-0457-9

Mitchell, T. M. (1997). *Machine Learning* (1st edition). McGraw-Hill Education.

Mohandes, M., Deriche, M., ve Aliyu, S. O. (2018). Classifiers Combination Techniques: A Comprehensive Review. *IEEE Access*, *6*, 19626-19639. https://doi.org/10.1109/ACCESS.2018.2813079

Molnar, C. (2022). Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. Independently published.

Nabipour, M., Nayyeri, P., Jabani, H., S., S., ve Mosavi, A. (2020). Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis. *IEEE Access*, *8*, 150199-150212. https://doi.org/10.1109/ACCESS.2020.3015966

Nieto Juscafresa, A. (2022). An introduction to explainable artificial intelligence with LIME and SHAP. https://diposit.ub.edu/dspace/handle/2445/192075

Niu, H., Xu, K., ve Wang, W. (2020). A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network. *Applied Intelligence*, *50*(12), 4296-4309. https://doi.org/10.1007/s10489-020-01814-0

Patel, J., Shah, S., Thakkar, P., ve Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259-268. https://doi.org/10.1016/j.eswa.2014.07.040

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., ve Cournapeau, D. (t.y.). Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*, 6.

PyEMD's documentation—PyEMD 0.2.13 documentation. (t.y.). Geliş tarihi 02 Aralık 2022, gönderen https://pyemd.readthedocs.io/en/latest/index.html

Reback, J., McKinney, W., Jbrockmendel, Bossche, J. V. D., Augspurger, T., Cloud, P., Gfyoung, Sinhrks, Klein, A., Roeschke, M., Hawkins, S., Tratner, J., She, C., Ayd, W., Terji Petersen, Garcia, M., Schendel, J., Hayden, A., MomIsBestFriend, … Mortada Mehyar. (2020). *pandas-dev/pandas: Pandas 1.0.3* (v1.0.3) [Software]. Zenodo. https://doi.org/10.5281/ZENODO.3715232

Ribeiro, M. T., Singh, S., ve Guestrin, C. (2016). *"Why Should I Trust You?": Explaining the Predictions of Any Classifier* (arXiv:1602.04938). arXiv. http://arxiv.org/abs/1602.04938

Rothman, D. (2020). *Hands-On Explainable AI (XAI) with Python: Interpret, visualize, explain, and integrate reliable AI for fair, secure, and trustworthy AI apps*. Packt Publishing Ltd.

Shen, J., ve Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*(1), 66. https://doi.org/10.1186/s40537-020-00333-6

Schneider, K., ve Farge, M. (2006). Wavelets: Mathematical Theory. In J.-P. Françoise, G. L. Naber, ve T. S. Tsun (Eds.), Encyclopedia of Mathematical Physics (pp. 426–438). Academic Press. https://doi.org/10.1016/B0-12-512666-2/00153-X

Schneider, P., ve Xhafa, F. (2022). Chapter 8 - Machine learning: ML for eHealth systems. In P. Schneider ve F. Xhafa (Eds.), Anomaly Detection and Complex Event Processing over IoT Data Streams (pp. 149–191). Academic Press. https://doi.org/10.1016/B978-0-12-823818-9.00019-5

Telegraph | Invention, History, ve Facts | Britannica. (2023, Kasım 4). https://www.britannica.com/technology/telegraph

Thakkar, A., ve Chaudhari, K. (2020). Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing*, *96*, 106684. https://doi.org/10.1016/j.asoc.2020.106684

TradingView – Tüm Piyasaları Takip Edin. (2023, Kasım 11). TradingView. https://tr.tradingview.com/

TÜBA - Türkçe Bilim Terimleri Sözlüğü. (t.y.). Geliş tarihi 12 Mayıs 2023, gönderen http://terim.tuba.gov.tr/

Wu, D., Wang, X., Su, J., Tang, B., ve Wu, S. (2020). A Labeling Method for Financial Time Series Prediction Based on Trends. *Entropy*, *22*(10), 1162. https://doi.org/10.3390/e22101162

XGBoost | Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (t.y.). Geliş tarihi 08 Ekim 2022, gönderen https://dl.acm.org/doi/abs/10.1145/2939672.2939785

Xiao, J., Zhu, X., Huang, C., Yang, X., Wen, F., ve Zhong, M. (2019). A New Approach for Stock Price Analysis and Prediction Based on SSA and SVM. *International Journal of Information Technology ve Decision Making*, *18*(01), 287-310. https://doi.org/10.1142/S021962201841002X

Xu, F., ve Tan, S. (2021). Deep learning with multiple scale attention and direction regularization for asset price prediction. *Expert Systems with Applications*, *186*, 115796. https://doi.org/10.1016/j.eswa.2021.115796

Yang, C., Zhai, J., ve Tao, G. (2020). Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory. *Mathematical Problems in Engineering*, *2020*, 1-13. https://doi.org/10.1155/2020/2746845

Yang, Y., Hu, X., ve Jiang, H. (2022). Group penalized logistic regressions predict up and down trends for stock prices. *The North American Journal of Economics and Finance*, *59*, 101564. https://doi.org/10.1016/j.najef.2021.101564

Yujun, Y., Yimei, Y., ve Jianhua, X. (2020). A Hybrid Prediction Method for Stock Price Using LSTM and Ensemble EMD. *Complexity*, *2020*, 1-16. https://doi.org/10.1155/2020/6431712

Yujun, Y., Yimei, Y., ve Wang, Z. (2021). Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition. *Soft Computing*, *25*(21), 13513-13531. https://doi.org/10.1007/s00500-021-06122-4

Yun, K. K., Yoon, S. W., ve Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, *186*, 115716. https://doi.org/10.1016/j.eswa.2021.115716

Zhang, X., Gu, N., Chang, J., ve Ye, H. (2021). Predicting stock price movement using a DBN-RNN. *Applied Artificial Intelligence*, *35*(12), 876-892. https://doi.org/10.1080/08839514.2021.1942520

Zhou, F., Zhou, H., Yang, Z., ve Yang, L. (2019). EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, *115*, 136-151. https://doi.org/10.1016/j.eswa.2018.07.065

Zhou, Z., Gao, M., Liu, Q., ve Xiao, H. (2020). Forecasting stock price movements with multiple data sources: Evidence from stock market in China. *Physica A: Statistical Mechanics and Its Applications*, *542*, 123389. https://doi.org/10.1016/j.physa.2019.123389

EKLER

EK1: TEZİN KAYNAK KODLARI

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# from ta import add_all_ta_features

# from ta.utils import dropna

from PyEMD import EMD

from scipy.ndimage.interpolation import shift

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import İsabet_score, Kesinlik_score, Duyarlılık_score, f1_score, roc_auc_score

from keras.models import Sequential

from keras.layers import Dense

train_ratio = 0.501

ann_epoch = 300

lag_number = 4

def lagged_data(data, lag_num):

 data = np.array(data).reshape(data.shape[0],)

 lag = []

 for i in range(lag_num+1):

 lag.append(shift(data, i, cval=np.NaN))

 return np.transpose(np.array(lag))[~np.isnan(lag).any(axis=0)]

dataset = pd.read_csv("../tez_veri_setleri/4KRX KOSPI, 1D.csv")

dataset = dataset[dataset['time'].between('2012-01-01', '2022-01-01')]

dataset = dataset[['close']]

dataset = dataset.dropna()
```

```
dataset = dataset.iloc[:, 0:(len(dataset.columns))].values
sc = MinMaxScaler()
dataset_scaled = sc.fit_transform(dataset)
dataset_scaled = dataset_scaled[: (dataset_scaled.shape[1]-1)]
emd = EMD()
imfs_dataset = emd(dataset_scaled)
imfs_dataset = np.transpose(imfs_dataset, axes=None)
imf_pred = []
for i in range(0, imfs_dataset.shape[1]):
 imfs_dataset0 = imfs_dataset[:, i]
 dataset_scaled0 = lagged_data(imfs_dataset0, lag_number)
 dataset_scaled0 = dataset_scaled0[~np.isnan(dataset_scaled0).any(axis=1)]
 training_set_scaled = dataset_scaled0[0:int(len(dataset)*train_ratio)]
 x_train = training_set_scaled[:, 1:(training_set_scaled.shape[1])]
 y_train = training_set_scaled[0:training_set_scaled.shape[0], 0]
 ###################################
 #ANN build
 ann = Sequential()
 ann.add(Dense(100, input_dim = (x_train.shape[1]), activation='relu'))
 ann.add(Dense(100, activation='relu'))
 ann.add(Dense(units=1, activation='tanh'))
 ann.compile(optimizer = 'adam', loss = 'mean_squared_error')
 ann.fit(x_train, y_train, batch_size = 30, epochs = ann_epoch)
########################
 #predicting the test set
```

```
 dataset_test                                                                    =
dataset_scaled0[int(len(training_set_scaled)):int(len(dataset_scaled0))]
 x_test = dataset_test[:, 1:(dataset_test.shape[1])]
 y_test = dataset_test[0:dataset_test.shape[0], 0]
 imf_pred.append(ann.predict(x_test))
imf_predx = np.take(np.array(imf_pred), 0, axis=2, out=None, mode='raise')
imf_predx = np.transpose(imf_predx, axes=None)
ypred = np.sum(imf_predx, axis=1, dtype=None, out=None)
dataset_test = dataset[int(len(training_set_scaled)):int(len(dataset))]
y_test_price = sc.fit_transform(dataset_test)[0:training_set_scaled.shape[0], 0]
y_test_price = y_test_price.reshape(y_test_price.shape[0],-1)
y_test_price = y_test_price[4:, :]
######################
#Hit_Rate calculation
y_test_price_differ = []
for i in range(0 (len(y_test_price)-1)):
 y_test_price_differ.append(y_test_price[i+1]- y_test_price[i])
ypred_differ = []
for i in range(0 (len(ypred)-1)):
 ypred_differ.append(ypred[i+1]- ypred[i])
#Confusion Matrix
y_test_price_direction = []
for i in range(0 (len(y_test_price_differ))):
 if (y_test_price_differ[i] > 0):
 y_test_price_direction.append(1)
 else:
```

```
 y_test_price_direction.append(0)
ypred_direction = []
for i in range(0 (len(ypred_differ))):
 if (ypred_differ[i] > 0):
 ypred_direction.append(1)
 else:
 ypred_direction.append(0)
ypred_differa = np.array(ypred_differ)
y_test_price_directiona = np.array(y_test_price_direction)
ypred_directiona = np.array(ypred_direction)
#########################################################################
train_ratio = 0.501
mode_order = 0
imfs_dataset0 = imfs_dataset[:, mode_order]
dataset_scaled0 = lagged_data(imfs_dataset0, lag_number)
dataset_scaled0_return = pd.DataFrame(dataset_scaled0)
dataset_scaled0_return.rename(columns={ dataset_scaled0_return.columns[0]:
"return" }, inplace = True)
dataset_scaled0_return['expected_change'] = dataset_scaled0_return['return'].diff() /
dataset_scaled0_return['return'].abs().shift()
def f(row):
 if row['expected_change'] > 0:
 val = 1
 else:
 val = 0
 return val
dataset_scaled0_return['return01'] = dataset_scaled0_return.apply(f, axis=1)
```

```
dataset_scaled0_return = dataset_scaled0_return['return01'].to_numpy()
dataset_scaled0_return = dataset_scaled0_return.reshape(-1, 1)
dataset_scaled0 = np.concatenate((dataset_scaled0_return, dataset_scaled0), axis=1)
training_set_scaled = dataset_scaled0[0:int(len(dataset)*train_ratio)]
x_train = training_set_scaled[:, 2:(training_set_scaled.shape[1])]
y_train = training_set_scaled[0:training_set_scaled.shape[0], 0]
###################################
#ANN build
ann = Sequential()
ann.add(Dense(100, input_dim = (x_train.shape[1]), activation='relu'))
ann.add(Dense(100, activation='relu'))
ann.add(Dense(units=1, activation='sigmoid'))
ann.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['İsabet'])
ann.fit(x_train, y_train, batch_size = 30, epochs = ann_epoch)
########################
#predicting the test set
dataset_test = dataset_scaled0[int(len(training_set_scaled)):int(len(dataset_scaled0))]
x_test = dataset_test[:, 2:(dataset_test.shape[1])]
y_test = dataset_test[0:dataset_test.shape[0], 0]
imf_pred0 = ann.predict(x_test)
imf_pred0 = imf_pred0[1:imf_pred0.shape[0]]
imf_pred0 = imf_pred0.reshape(imf_pred0.shape[0])
#########################################################################
train_ratio = 0.50
dataset_RF = np.stack((y_test_price_directiona, ypred_differa, imf_pred0), axis = 1)
training_set_scaled = dataset_RF[0:int(len(dataset_RF)*train_ratio)]
```

```
x_train = training_set_scaled[:, 1:(training_set_scaled.shape[1])]
y_train = training_set_scaled[0:training_set_scaled.shape[0], 0]
###################################
#random_forest build
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 250, random_state = 190,
  n_jobs = -1, max_depth = 37)
classifier.fit(x_train, y_train)
# ###################################
#predicting the test set
dataset_test = dataset_RF[int(len(training_set_scaled)):int(len(dataset_RF))]
x_test = dataset_test[:, 1:(dataset_test.shape[1])]
y_test = dataset_test[0:dataset_test.shape[0], 0]
ypred = classifier.predict(x_test)
plt.plot(dataset_scaled[-100:], color = 'blue')
plt.plot(np.sum(imf_predx, axis=1, dtype=None, out=None)[-100:] , color = 'red')
ypred = np.where(ypred>0.5,1,0)
ypred = ypred.reshape(ypred.shape[0])
df = pd.DataFrame({'y_test_price_direction' : y_test, 'ypred_direction' : ypred})
confusion_matrix = pd.crosstab(df['y_test_price_direction'], df['ypred_direction'],
rownames=['Actual'], colnames=['Predicted'])
print(confusion_matrix)
print(İsabet_score(y_test, ypred))
print(Kesinlik_score(y_test, ypred, average='binary'))
print(Duyarlılık_score(y_test, ypred, average='binary'))
print(f1_score(y_test, ypred, average='binary'))
```

```
print(roc_auc_score(y_test, ypred, average='micro'))

###################################

#local interpretable model-agnostic explanations (LIME)

from lime import lime_tabular

explainer = lime_tabular.LimeTabularExplainer(x_train, mode="classification")

explanation_predict_proba = []

for i in range(0, int(x_test.shape[0])):

 explanation = explainer.explain_instance(x_test[i], classifier.predict_proba, num_features=2)

explanation_predict_proba.append(np.transpose(explanation.predict_proba[0:2].reshape(explanation.predict_proba[0:2].shape[0],-1), axes=None))

explanation_predict_proba = np.array(explanation_predict_proba)

explanation_predict_proba = explanation_predict_proba[:, 0]

""" Explanation_predict_proba elde edildikten sonra "0" ya da "1" sütununda öncelikle

"accepted_proba"'den büyük bir değer olup olmadığına bakılır. Şayet yoksa tahmin yapılmayarak

ilgili periyod boş geçilir. Yani ilgili satır için tahmin değeri "NA" olarak atanır. Diğer taraftan

"accepted_proba"'den büyük bir değer varsa ve eğer bu değer "0" sütununda yer alıyorsa ve eğer yine aynı

periyot için y_test değeri de "0" olarak gözlemlenmiş ise ilgili satır "1" aksi durumda "0" değerini alır.

"accepted_proba"'den büyük değer eğer "1" sütununda yer alıyorsa ve eğer yine aynı periyot için y_test

değeri de "1" olarak gözlemlenmiş ise yine ilgili satır "1" aksi durumda "0" değerini alır. Nihai olarak
```

"1", "0", ve "NA" değerlerinden oluşan tek bir sütun elde edilmiş olur. Bu sütunda yer alan bütün 1'ler

toplandıktan sonra toplam "0" ve "1" sayısına bölünerek nihai final isabet oranı elde edilmiş olur. Taslak formül;

EĞER(YADA(explanation_predict_proba[i, 0]>0.8;explanation_predict_proba[i, 1]>0.8);

EĞER(explanation_predict_proba[i, 0]>explanation_predict_proba[i, 1];

EĞER(y_test[i]=0;1;0);EĞER(y_test[i]=1;1;0));"NAN") """

reliable_prob = 0.50

final = []

for i in range(0, int(x_test.shape[0])):

if explanation_predict_proba[i, 0] >= reliable_prob or explanation_predict_proba[i, 1] >= reliable_prob:

if explanation_predict_proba[i, 0] >= explanation_predict_proba[i, 1]:

if y_test[i] == 0:

final.append(1)

else:

final.append(0)

elif y_test[i] == 1:

final.append(1)

else:

final.append(0)

else:

final.append("NAN")

LIME hitrate calculation

res = final.count(1)+final.count(0)

hitrate = final.count(1)/(res)

```
# print(hitrate)


# Plot the hitrate and Number of trusted predictions trade-off
reliability_low = 500
reliability_high = 1000
hitrate = []
res = []
reliable_prob = []
for j in range(reliability_low, reliability_high+1): #The range() method does not include the end number, so add 1 to the end.
 final = []
 for i in range(0, int(x_test.shape[0])):
 if explanation_predict_proba[i, 0] >= (j/(reliability_high)) or explanation_predict_proba[i, 1] >= (j/(reliability_high)):
 if explanation_predict_proba[i, 0] >= explanation_predict_proba[i, 1]:
 if y_test[i] == 0:
 final.append(1)
 else:
 final.append(0)
 elif y_test[i] == 1:
 final.append(1)
 else:
 final.append(0)
 else:
 final.append("NA")
 res.append(int(final.count(1)+final.count(0)))
 hitrate.append(final.count(1)/(res[j-reliability_low]))
```

```
 reliable_prob.append((j)/(reliability_high))
# ##################################
# #predicting the test set with XGBoost
# #XGBoost build
# from xgboost import XGBClassifier
# classifier = XGBClassifier( metrics={'İsabet'},
# # eval_metric='mlogloss',
# use_label_encoder=False,
# verbosity = 3,
# subsample=1)
# classifier.fit(x_train, y_train)
# ###################################
# #predicting the test set
# dataset_test = dataset_RF[int(len(training_set_scaled)):int(len(dataset_RF))]
# x_test = dataset_test[:, 1:(dataset_test.shape[1])]
# y_test = dataset_test[0:dataset_test.shape[0], 0]
# ypred = classifier.predict(x_test)
# ypred = np.where(ypred>0.5,1,0)
# ypred = ypred.reshape(ypred.shape[0])
# df = pd.DataFrame({'y_test_price_direction' : y_test, 'ypred_direction' : ypred})
# confusion_matrix = pd.crosstab(df['y_test_price_direction'], df['ypred_direction'],
rownames=['Actual'], colnames=['Predicted'])
# print(confusion_matrix)
# print(İsabet_score(y_test, ypred))
# print(Kesinlik_score(y_test, ypred, average='binary'))
# print(Duyarlılık_score(y_test, ypred, average='binary'))
```

```
# print(f1_score(y_test, ypred, average='binary'))

# print(roc_auc_score(y_test, ypred, average='micro'))

# hitrate_SP500 = hitrate

# res_SP500 = res

# hitrate_res_SP500 = pd.DataFrame({'hitrate_SP500': hitrate_SP500,'res_SP500': res_SP500, 'reliable_prob': reliable_prob})

# hitrate_res_SP500.to_excel("hitrate_res_SP500.xlsx")

# hitrate_NI225 = hitrate

# res_NI225 = res

# hitrate_res_NI225 = pd.DataFrame({'hitrate_NI225': hitrate_NI225,'res_NI225': res_NI225, 'reliable_prob': reliable_prob})

# hitrate_res_NI225.to_excel("hitrate_res_NI225.xlsx")

# hitrate_XU100 = hitrate

# res_XU100 = res

# hitrate_res_XU100 = pd.DataFrame({'hitrate_XU100': hitrate_XU100,'res_XU100': res_XU100, 'reliable_prob': reliable_prob})

# hitrate_res_XU100.to_excel("hitrate_res_XU100.xlsx")

# hitrate_KOSPI = hitrate

# res_KOSPI = res

# hitrate_res_KOSPI = pd.DataFrame({'hitrate_KOSPI': hitrate_KOSPI,'res_KOSPI': res_KOSPI, 'reliable_prob': reliable_prob})

# hitrate_res_KOSPI.to_excel("hitrate_res_KOSPI.xlsx")

# hitrate_DAX = hitrate

# res_DAX = res

# hitrate_res_DAX = pd.DataFrame({'hitrate_DAX': hitrate_DAX,'res_DAX': res_DAX, 'reliable_prob': reliable_prob})

# hitrate_res_DAX.to_excel("hitrate_res_DAX.xlsx")
```

```
# hitrate_FTSE100 = hitrate

# res_FTSE100 = res

# hitrate_res_FTSE100 = pd.DataFrame({'hitrate_FTSE100': hitrate_FTSE100,'res_FTSE100': res_FTSE100, 'reliable_prob': reliable_prob})

# hitrate_res_FTSE100.to_excel("hitrate_res_FTSE100.xlsx")

# plt.plot(res_SP500, hitrate_SP500, color = 'blue', label="SP500")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel('Number of trusted predictions', fontsize=8)

# plt.plot(res_NI225, hitrate_NI225, color = 'green', label="NI225")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel(", fontsize=8)

# plt.plot(res_XU100, hitrate_XU100, color = 'red', label="XU100")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel('Number of trusted predictions', fontsize=8)

# plt.plot(res_KOSPI, hitrate_KOSPI, color = 'cyan', label="KOSPI")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel('Number of trusted predictions', fontsize=8)

# plt.plot(res_DAX, hitrate_DAX, color = 'magenta', label="DAX")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel('Number of trusted predictions', fontsize=8)

# plt.plot(res_FTSE100, hitrate_FTSE100, color = 'orange', label="FTSE100")

# plt.legend(loc="lower left", fontsize=8)

# plt.xlabel(", fontsize=8)

# plt.savefig('hitrates_traing_days.png', dpi=1200, bbox_inches='tight')

# plt.plot(reliable_prob, hitrate_SP500, color = 'blue', label="SP500")

# plt.legend(loc="lower right", fontsize=8)
```

```
# plt.xlabel('Reliability level', fontsize=8)
# plt.plot(reliable_prob, hitrate_NI225, color = 'green', label="NI225")
# plt.legend(loc="lower right", fontsize=8)
# plt.xlabel('Reliability level', fontsize=8)
# plt.plot(reliable_prob, hitrate_XU100, color = 'red', label="XU100")
# plt.legend(loc="lower right", fontsize=8)
# plt.xlabel('Reliability level', fontsize=8)
# plt.plot(reliable_prob, hitrate_KOSPI, color = 'cyan', label="KOSPI")
# plt.legend(loc="lower right", fontsize=8)
# plt.xlabel('Reliability level', fontsize=8)
# plt.plot(reliable_prob, hitrate_DAX, color = 'magenta', label="DAX")
# plt.legend(loc="lower right", fontsize=8)
# plt.xlabel('Reliability level', fontsize=8)
# plt.plot(reliable_prob, hitrate_FTSE100, color = 'orange', label="FTSE100")
# plt.legend(loc="lower right", fontsize=8)
# plt.xlabel('Reliability level', fontsize=8)
# plt.savefig('hitrates_reliability.png', dpi=1200, bbox_inches='tight')
```

ÖZ GEÇMİŞ

Taha Buğra ÇELİK, İstanbul Bahçelievler Lisesi'ni bitirdikten sonra Sakarya Üniversitesi İktisadi ve İdari Bilimler Fakültesi, İşletme bölümünden lisans, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, Sayısal Yöntemler Anabilim dalından yüksek lisans eğitimini tamamladı. 2016 yılında OMÜ LEE İşletme Ana Bilim Dalı Doktora programına girdi. Lisans eğitimini tamamladığından bu yana Araştırma görevlisi olarak görev yapmaktadır. Temel ilgi alanları sermaye piyasaları, matematik, felsefe ve bilgisayar teknolojileridir.

İletişim Bilgileri

ORCID ID : https://orcid.org/0000-0003-2286-286X

Yayınlar:

1. Ican, O., ve Celik, T. B. (2017). Stock market prediction performance of neural networks: A literature review. International Journal of Economics and Finance, 9(11), 100-108.

2. İcan, Ö., ve Çelik, T. B. (2021). A review on smart energy management systems in microgrids based on power generating and environmental costs. Applied Operations Research and Financial Modelling in Energy: Practical Applications and Implications, 51-67.

3. Çelik, T. B., İcan, Ö., ve Bulut, E. (2023). Extending machine learning prediction capabilities by explainable AI in financial time series prediction. Applied Soft Computing, 132, 109876.

4. İcan, Ö., ve Çelik, T. B. (2023). Weak-form market efficiency and corruption: a cross-country comparative analysis. Journal of Capital Markets Studies (ahead-of-print).