GAZİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BİTİRME PROJESİ

Kredi Kartı Dolandırıcılık Tespitinde Yapay Zeka Temelli Yöntemler

Final Raporu

11100 Kelime

Hazırlayan

Hilal Nur Tek (181180068)

Sezen Sude Gül (181180036)

Bitirme Projesi Danışmanı

Dr. Öğr. Üyesi Yılmaz Atay

Ankara-2022

İÇİNDEKİLER

1. GİRİŞ

Binlerce yıldır insanların ihtiyaçlarını karşılamak amacıyla kullandıkları alışveriş eylemi, teknolojinin gelişmesi ile bağlantılı olarak ilerlemiştir. Günümüzde sanal ortamlarda da sıklıkla gerçekleştirilen alışveriş, beraberinde ödeme yönteminin de sanallaşmasını gerektirmiştir. Online ödeme işlemi için kullanılan yöntemlerden biri de kredi kartlarıdır.

5464 sayılı Banka Kartları ve Kredi Kartları Kanunu'nun üçüncü maddesinin e bendinde kredi kartı: "Nakit kullanımı gerekmeksizin mal ve hizmet alımı veya nakit çekme olanağı sağlayan basılı kartı veya fiziki varlığı bulunmayan kart numarasını" ifade eder şeklinde tanımlanmıştır. Bu tanımdan da yola çıkarak insanların alışverişlerinde kullanmasına, nakit para çekmesine ve son zamanlarda özellikle artan temassız ödeme olanaklarını sağladığı için kredi kartı kullanımına yönelik taleplerin arttığı gözlemlenebilmektedir. Bu durum kötü niyetli kişiler tarafınca da fark edildiğinden dolayı, kredi kartı dolandırıcılığı, fiziki kredi kartı hırsızlığı ve kart bilgilerinin çalınması gibi olumsuz davranışlar da artış göstermiştir. Dolandırıcılık eylemi sonucunda şahıslar, mal veya hizmet sunan işletmeler ve bankalar bu durumdan etkilenmektedir [1]. Bu olumsuz durumların tespiti ve önüne geçilmesi için geleneksel veya çeşitli yazılım yöntemleri kullanılmalıdır. Sanal ortamda ödeme imkanlarının artması ile, dolandırıcılık yöntemlerinin de doğru orantılı olarak artması, dolandırıcılık eyleminin gerçek zamanlı olarak tespitini ve önüne geçilmesini zorlaştırmaktadır. Bu nedenle bu çalışmada, "Kredi Kartı Dolandırıcılık Tespitinde Yapay Zeka Temelli Yöntemler" ele alınmıştır.

Kredi kartı dolandırıcılık tespitinde yapay zeka yöntemlerinin uygulanabilmesi için kredi kartı ve kullanıcının alışverişlerinden yola çıkarak elde edilen veriler kullanılarak analizler yapılmaktadır. Bu analizlerden yola çıkarak geliştirilen algoritmalar sayesinde bir dolandırıcılık tespiti için kullanılması diğer algoritmalara göre daha verimli algoritma Kesinlik (Precision) değeri ROC Eğrisi (ROC AUC) Değeri gibi başarı değerlendirme ölçütleri kullanılarak sunulacaktır.

2. LİTERATÜR TARAMASI

Literatürde bulunan çeşitli makaleler incelendiğinde, kredi kartı dolandırıcılık tespitinde sıklıkla kullanılan algoritmaların makine öğrenme ve derin öğrenmeye yönelik algoritmalar oldu tespit edilmiştir. Bu bölümde, literatürde karşılaşılan makaleler incelenip yorumlanmıştır.

2.1. APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions

Düzensiz davranışları yakalayabilecek önemli özellikleri tespit edebilmek, dolandırıcılık tespit sürecindeki en önemli adımlardan biridir. Bu çalışmada, içsel ve ağla ilgili özellikler birleştirilmiştir. İçsel özellikler, işlemi analiz eder ve işlemin normal müşteri profiline uyup uymadığını karşılaştırır. Bu özellikler, kredi kartı sahibinin geçmişte yaptığı işlemlerin RFM değerleri (Yenilik, Sıklık ve Parasal Değer) türetilerek oluşturulur. Ağ tabanlı özellikler ise,

işlemler aracılığıyla ilişkili kredi kartı sahipleri ve üye işyerlerinden oluşan bir ağ oluşturup bu ağı analiz ederek her işlemi karakterize eder. Büyük bir Belçikalı kredi kartı düzenleyicisinden elde edilen ve yaklaşık 3,3 milyon işlemden oluşan benzersiz bir veri seti ile sınırlı sayıda onaylanmış dolandırıcılık işlemi kullanılarak ağ üzerinden hileli etkiyi yaymak için toplu bir çıkarım algoritması kullanılmakta ve bir ifşa puanı türeterek her bir ağ nesnesinin şüpheliliğine karar verilmektedir. Önerilen yöntem logistic regression, neural networks, random forests modelleriyle karşılaştırılmış olup 0.98'den yüksek AUC skoru elde edilmiştir [2].

2.2. Fraud detection using self-organizing map visualizing the user profiles

Bu makalede, kullanıcı hesaplarının görselleştirilmesine ve eşik tipi tespitine dayalı bir dolandırıcılık tespit yöntemi önerilmektedir. Yaklaşımda kullanılan görselleştirme tekniği Kendi Kendini Düzenleyen Harita'dır (SOM). SOM tekniği orijinal haliyle sadece vektörleri görselleştirdiğinden ve çalışma içerisinde kullanıcı hesapları, kullanıcı sıralı etkinliklerini yansıtan bir kayıt koleksiyonunu depolayan matrisler olarak temsil edildiğinden, SOM ızgarasında matris görselleştirmesi için bir yöntem önerilmektedir. Ayrıca, SOM U matrisi temelinde bir algılama eşiği ayarı yöntemi önerilmektedir. Kullanıcı hesaplarının SOM'a yansıtılmasının ardından, eşik tipi ikili sınıflandırma algoritması kullanılarak hileli hesaplar tespit edilmiş ve belirli bir SOM'nin U matrisindeki çıkıntıyı bularak sınıflandırma eşiği ayarı için bir yöntem önerilmiştir. Görselleştirme yoluyla dolandırıcılık tespiti, 2 boyutlu bir uzaya yansıtılan yüksek boyutlu verinin mümkün olduğu gerçeğinden dolayı faydalı ve çekici bir veri analizi yaklaşımıdır. Çalışma, 1.01.2005 ile 1.03.2005 tarihleri arasında Polonya'nın Varşova şehrinde 10.000 kredi kartı sahibinin hesabından oluşturulan ve işlemde harcanan para miktarı, işlemin gerçekleştirildiği yer, işlem zamanı özellikleriyle karakterize edilmiş bir veri seti üzerinde sürdürülmüştür. SOM-clustering-based, GHSOM-based, GMM-based metodlarıyla karşılaştırılan yöntem 1.0 değerinde F1-score elde etmiştir [3].

2.3. Critical Analysis of Machine Learning Based Approaches for Fraud Detection in Financial Transactions

Bu makalenin amacı, finansal işlemler için bir dolandırıcılık tespit sistemine entegre edilebilecek seçilmiş makine öğrenimi ve aykırı değer tespit tekniklerini gözden geçirmektir. Karşılaştırma öncesinde dolandırıcılık analizinin ve veri setlerinin özellikleri hakkında modeli eğitime hazırlama süreci ve eğitim sonrası performans değerlendirmesi özelinde birtakım açıklamalar yapılmıştır. Ardından; Bayesian Networks, Recurrent Neural Networks, Support Vector Machines, Fuzzy Logic, Hidden Markov Model, K-Means Clustering, K-Nearest Neighbor gibi çeşitli makine öğrenme algoritmaları ve bunların dolandırıcılık tespit alanındaki mevcut uygulamaları en iyi yaklaşımı bulmak amacıyla her modelin olumlu ve olumsuz özellikleri göz önünde bulundurularak karşılaştırmalı olarak tartışılmıştır [4].

2.4. A cost-sensitive decision tree approach for fraud detection

Bu çalışmada, her bir terminal olmayan düğümde ayırma özniteliği seçilirken yanlış sınıflandırma maliyetlerinin toplamını en aza indiren, maliyete duyarlı yeni bir karar ağacı yaklaşımı geliştirilmiş ve bu yaklaşımın performansı iyi bilinen geleneksel sınıflandırma modelleriyle (YSA, SVM) karşılaştırılmıştır. Sunulan yaklaşımda, yanlış sınıflandırma maliyetleri değişken olarak alınır. Sonuçlar, bu maliyete duyarlı karar ağacı algoritmasının, doğruluk ve gerçek pozitif oran gibi iyi bilinen performans ölçütlerine ve aynı zamanda yeni tanımlanmış bir maliyete duyarlı ölçüt özgüllüğüne göre verilen problem setinde mevcut iyi bilinen yöntemlerden daha iyi performans gösterdiğini göstermektedir. Buna göre, hile tespit sistemlerinde bu yaklaşımın uygulanması ile hileli işlemlerden kaynaklanan mali kayıplar daha da azaltılabilmektedir. Gerçek veriler üzerinde çalışılan yöntemin karar ağaçları, YSA ve SVM gibi klasik modellerden daha iyi sonuç verdiği gözlemlenmiştir [5].

2.5. An Efficient Real Time Model For Credit Card Fraud Detection Based On Deep Learning

Bu çalışma, derin öğrenme paradigmalarını araştıran karşılaştırma çalışmalarının az ve gerçek zamanlı yaklaşımlara yeterli önemin verilmemesi dolayısıyla ortaya çıkmış olup kredi kartı dolandırıcılığı problemi ile başa çıkmak amacıyla derin sinir ağı (deep neural network) teknolojisine dayanan canlı bir kredi kartı dolandırıcılık tespit sistemi önermektedir. Çalışma içerisindeki gerçek zamanlı veri akış hatlarının oluşturulması Kafka teknolojisi ile gerçekleştirilmiştir. Kullanılan veri seti, Worldline ve ULB'nin Machine Learning Group'un büyük veri madenciliği ve dolandırıcılık tespiti üzerine bir araştırma işbirliği sırasında toplanan ve analiz edilen, Eylül 2012'de Avrupa kartları tarafından iki gün içinde yapılan işlemleri içermektedir. Yapılan testler ve bazı tipik gerçek zamanlı ikili sınıflandırıcıların Deep NN Auto encoders'a karşı; Linear SVM Regression, Logistic Regression, NN Based Classification ve Non Linear Auto Regression ile oluşturulan performans bazlı karşılaştırma çalışmasından sonra, kıyaslama deneyleri Deep NN Auto encoders'ın F1 puanına dayanarak umut verici sonuçlara sahip olduğunu göstermektedir [6].

2.6. A Review of Credit Card Fraud Detection Techniques

Bu çalışmada Hindistan öncelikli olmak üzere dünya üzerinde gerçekleştirilen dolandırıcılık türleri ve bu türlere karşın kullanılabilecek makine öğrenmesi algoritmaları karşılaştırılmıştır. Karşılaştırma sonucunda her algoritmanın odak noktası ve zayıf yönleri üzerinde durulmuştur. Random Forest ve KNN gibi birkaç strateji, küçük bir veri kümesinde iyi çalışır fakat büyük bir veri kümesinde yeterince esnek değildir. SVM ve Karar Ağacı gibi bazıları ön işlemeden geçmiş ve örneklenmiş veriler üzerinde iyi sonuçlar verirken, ANN gibi algoritmalar önceki durumdan öğrenir ve Genetik Algoritma tespit etmede hızlıdır. Bulanık sistemler ve Lojistik Regresyon gibi bazı teknikler, ham örneklenmemiş verilerle daha iyi sonuç hataları verir. Çalışmada sonuç olarak, asıl üzerinde durulması gereken problemin dolandırıcılık işleminin gerçek zamanlı tespit edilmesi olduğu vurgulanmıştır [7].

2.7. Real Time Credit Card Fraud Detection on Huge Imbalanced Data using Meta-Classifiers

Bu çalışmada, dengesiz ve büyük verilerdeki dolandırıcılık eylemlerini tespit etmek amacıyla kullanılabilecek gerçek zamanlı ağaç tabanlı bir meta sınıflandırıcı TBMC sunulmaktadır. Geliştirilen meta sınıflandırıcı tabanlı model, iki düzeyde tahminlere dayalı olarak çalışmaktadır. Birinci seviyedeki tahminler Random Forest sınıflandırıcısı tarafından, ikinci seviyedeki tahminler ise Karar Ağaçları ve Gradient Boosted Trees ile oluşturulan bir topluluk tarafından gerçekleştirilmektedir. İki seviyede elde edilen sonuçlar ise, nihai tahminleri oluşturmak amacıyla birleştirilmektedir. Önerme için UCSD-FICO verileriyle yapılan deneyler sonrasında sonuçlar mevcut modellerle karşılaştırılmıştır. Deneysel sonuçlar, geliştirilen TBMC modelinin performansının arttığını göstermektedir. Bununla birlikte, geliştirilen model, dengesiz veriler için uzmanlaşmış dengeleme önlemleri olan orta düzeyde MCC ve BCR sergilemektedir [8].

2.8. Real-time Credit Card Fraud Detection Using Machine Learning

Bu çalışma, gerçek dünyadaki işlemlerde tespit edilen dört ana dolandırıcılık olayına odaklanmaktadır. Çalışma dahilinde her bir dolandırıcılık örneğinin makine öğrenmesi algoritmaları kullanılarak incelenmesi, bu şekilde en iyi yöntemin seçilmesi ve hile türlerine göre en uygun algoritmanın seçilmesi için kapsamlı bir kılavuz oluşturulması amaçlanmıştır. İki veri kaynağının birleştirilmesiyle oluşan projede ele alınan bir diğer önemli unsur da hilelerin gerçek zamanlı tespitidir. Bunun için, gerçekleştirilen eylemin hileli olup olmadığına karar verilmesi amacıyla örnekleme yöntemleri ile değerlendirilen, çarpık dağılıma sahip veri üzerinde tahmine dayalı analitik ve bir API modülü uygulanmıştır. Bu uygulama sonucu dört dolandırıcılık modelinde (100$'ın üzerindeki İşlem, Bilinmeyen web adresi, Riskli MCC, ISO-Yanıt Kodu) en yüksek doğruluk oranlarına (sırasıyla %74, %72, %83, ve %91) makine öğrenmesi modellerinin LR, LR, NB ve SVM olduğu gözlemlenmiştir [9].

2.9. Data mining for credit card fraud: A comparative study

Bu makale, kredi kartı sahtekarlığını daha iyi tespit etme (ve dolayısıyla kontrol etme ve kovuşturma) girişiminin bir parçası olarak iki gelişmiş veri madenciliği yaklaşımını, destek vektör makinelerini ve rastgele ormanları, iyi bilinen lojistik regresyonla birlikte değerlendirir. Çalışmada, Ocak 2006-Ocak 2007 dönemine ait uluslararası bir kredi kartı operasyonundan elde edilen işlemler kullanılmaktadır. Performans değerlendirmesi için, farklı düşük örnekleme seviyelerine sahip eğitim veri kümelerinden çok daha düşük sahtekarlık oranına (% 0,5) sahip bir test veri kümesi kullanılmıştır. Duyarlılık, G-mean ve ağırlıklı-doğruluk, eğitim verilerinde daha düşük dolandırıcılık oranları ile azalırken, kesinlik ve özgüllüğün ters bir eğilim gösterdiği; F-score ve AUC'de lojistik regresyon, eğitim

verilerinde değişen oranlarda dolandırıcılık ile benzer performansı korurken, RF ve SVM, AUC'de azalan bir eğilim ve F'de artan bir eğilim göstermiştir. Lojistik regresyon, farklı düşük örnekleme seviyeleri ile benzer performansı korurken, üst dosya derinliklerindeki SVM performansı, eğitim verilerinde daha düşük dolandırıcılık oranı ile artma eğilimi göstermiştir. Rastgele ormanlar, üst dosya derinliklerinde çok daha yüksek performans göstermiştir. Böylece, üst derinliklerde daha az yanlış pozitif ile daha fazla dolandırıcılık vakasını yakamaktadırlar; bu, dolandırıcılık tespit modellerinin gerçek hayatta kullanımında önemli bir husustur. Bu çalışmada tekniklerin parametrelerini optimize etmek için kasıtlı bir girişimde bulunulmamıştır. Parametre ayarlama DVM için önemli olabilir ve dengeli örneklemenin, dengesiz veriler üzerinde Rastgele Ormanların kullanılmasında avantajlı olduğu not edilmiştir [10].

2.10. Yapay Zeka Destekli ÇOKS Yöntemi ile Kredi Kartı Sahtekarlığının Tespiti

Bu makalede; öncelikle kredi kartının günlük yaşamdaki kullanıma ve dolandırıcılık terimlerine değinilmiştir. Dolandırıcılık eyleminin önlenmesi ve ayırt edilebilmesindeki önem vurgulanmıştır. Kredi kartı sahteciliğini tespite yönelik araştırmalarda çoğunlukla makine öğrenmesi algoritmalarından ve farklı sınıflandırma algoritmalarının kullanıldığı görülmektedir. Fakat kredi kartı sahteciliğini tespit etmek amacıyla karar ağacı, KNN ve naive bayes classifierlarının bir arada kullanıldığı bir çalışmanın literatürde mevcut olmadığı belirtilmiştir. Bu gözlemden yola çıkarak bu makaleye yönelik yapılan çalışma kapsamında kredi kartı dolandırıcılığa yönelik karar ağacı, KNN ve naive bayes makine öğrenmesi algoritmalarından yararlanan ve Çoğunluk Oyu ile Karar Verme Sistemi (ÇOKS) olarak adlandırılan yeni bir sezgisel algoritma geliştirilmiştir. Bu algoritmanın ortak karar verme mekanizması için de bir sayısal devre tasarımı lojik fonksiyonu olan çoğunluk fonksiyonundan faydalanılmıştır. ÇOKS'nin etkinliği her biri 30 farklı özniteliğe sahip 284,807 kredi kartı işleminin yer aldığı bir veri kümesi üzerinde test edilmiştir. Yürütülen testler finansal güvenliği hedefleyen bu yeni yöntemin %99,93 doğruluk oranı, %95,60 kesinlik oranı ve %80,0 ROC AUC değeri ile veri kümesindeki bir işlemi sahte veya yasal işlem olarak sınıflandırabilmeyi başarmıştır [11].

2.11. Credit Card Fraud Detection using Machine Learning Algorithms

Bu makale, kredi kartı dolandırıcılıklarının kolaylaşmasından ve çevrimiçi ödeme kullanılan sitelerin artmasının dolandırıcılık riskini de artırdığından bahsetmektedir. Dolandırıcılık oranlarındaki artış, farklı makine öğrenimi yöntemlerini kullanarak kredi kartı sahtekarlıklarını tespit ve analiz etmeye yönlendirmiştir. Makale kapsamında yapılan çalışmaların temel amacı, müşterilerin geçmiş işlem detaylarını analiz etmek amacıyla Streaming Transaction Data için yeni bir dolandırıcılık tespit yöntemi geliştirmektir. Bu kapsamda öncelikle kart sahiplerinin işlem tutarlarına göre, işlemler farklı gruplara ayrılır. Daha sonra farklı gruplardan yapılan işlemleri bir araya getirmek için kayan pencere stratejisi kullanılarak, gruplar ayarlanır Sonrasında ise farklı sınıflandırıcılar, gruplar üzerinde ayrı ayrı eğitilir. Böylelikle en iyi sınıflandırma yöntemi seçilir ve kavram kayması problemini çözmek

için bir geri besleme mekanizması oluşturulmuş olur. Sonuç olarak bu makalede, müşteriler işlemlerine göre gruplanır ve her kart sahibi için bir profil oluşturularak kişilerin davranış kalıplarının çıkartıldığı yeni bir dolandırıcılık tespiti yöntemi geliştirilmiştir. Daha sonra bu veriler üç farklı gruba ayrılarak farklı sınıflandırma teknikleri uygulanır ve daha sonra her sınıflandırıcı türü için puanlama yapılır. Araştırma sonunda Lojistik regresyon, karar ağacı ve random forest algoritmalarının daha iyi sonuç veren algoritmalar olduğunu gözlemlenmiştir [12].

2.12. Credit Card Fraud Detection Using Autoencoder Neural Network

Bu makalede öncelikle kredi kartı dolandırıcılıklarının artma nedenlerinden ve kredi kartı dolandırıcılık nedenlerinden bahsedilmiştir. Bunun asıl sebebinin güvenlik açıkları olduğu belirtilmiştir. Bundan dolayı kredi kartı dolandırıcılık tespitinin önemine değinilmiştir. Tespit için kullanılan yöntemlerin genellikle big data ve makine öğrenmesi algoritmaları olduğu bilinmektedir. Burada kullanılan classification problemlerinin en büyük sorunu büyük çaplı veri setlerindeki dengesiz verilerdir. Çünkü bu veri setlerinde asıl ilgilenilmek istenen konu dolandırıcılık verileriyken, yasal olan verilerin sayıları çok çok daha fazladır. Bu nedenle bu makalede gürültüden arındırılmış outoencoder ve oversampling üzerine durulmuştur. Yeni azınlık sınıfı örneklerini sentezlemek için oversampling algoritması kullanılır, ancak bu algoritmanın problemi gürültü ihtimalidir. Bu makalede bu gibi gürültü sorunlarına değinilmiş, yalnızca azınlık sınıfı örneğini yanlış sınıflandırma maliyetiyle oversampling yapmayan aynı zamanda örneklenen veri kümesini gürültüden arındırabilen ve sınıflandırabilen bir gürültü giderme autoencoder sinir ağı (DAE) algoritması geliştirilmiştir [13].

2.13. Spotting Collective Behaviour of Online Frauds in Customer Reviews

Bu makalede; öncelikle dolandırıcılık faliyetlerinden ve özellikle mail veya siteler aracılığıyla insanları kandırmaya çalışan toplu dolandırıcılık faliyetlerinden bahsedilmiştir. Sonrasında ise grup spam tespitinin, grubun belirsiz tanımlanması, gruplar arası dinamiklerin çeşitlive etiketli grup düzeyinde spam verilerinin az olması nedenleriyle bireysel sahtekarlık tespitinden daha zor olduğu anlatılmaktadır. Bu makalede, dolandırıcılık işlemini tespit etmek için unsupervised bir yöntem olan DeFrauder'ı önerilmektedir. DeFrauder yöntrmi; İlk olarak, temel ürün inceleme grafiğinden yararlanır ve gözden geçirilen veriler arasında çok yönlü işbirliğini modelleyen çeşitli davranışsal sinyalleri birleştirerek olması muhtemel dolandırıcılık gruplarını tespit eder. Ardından, gözden geçirenleri bir yerleştirme alanına eşler ve her gruba bir spam puanı atar, böylece oldukça benzer davranışsal özelliklere sahip spam göndericilerden oluşan gruplar yüksek spam puanı elde eder. Ve Buna göre dolandırıcılık puanı verilmiş olur.

Dört gerçek dünya veri setindeki beş temel ile karşılaştırıldığında, DeFrauder veri setlerinde %17,11 daha yüksek NDCG@50 (ortalama olarak) ile en iyi temel çizgiden daha iyi

performans göstererek üstün performans gösterir. Veri setleri olarak ise; Amazon, Playstore, YelpNYC ve YelpZip verisetlerinden yararlanılmıştır [14].

2.14. Kredi Kartı Dolandırıcılık Tespitinin Makine Öğrenmesi Yöntemleri İle Tahmin Edilmesi

Bu makalede kredi kartı, kredi kartı kullanımı ve dünyadaki yerinden genel olarak bahsedilmiş ve haliyle artan dolandırıcılık girişimlerinin tespitinin önemine değinilmiştir. Günümüzde herhangi bir kredi kartı bilgilerine ulaşmanın oldukça kolaylaşması kredi kartı dolandırıcılarına fırsat sunmaktadır. Bunun önüne geçebilmek için, kredi kartı ile gerçekleştirilen hesap hareketleri değişikliğinde zaman ve harcamaların analiz edilmesi sayesinde dolandırma amaçlı verilerin analizi ile dolandırıcılığı önlemeye yönelik yöntemler geliştirilebilinir edebilir. Bu makale kapsamında yapılan çalışmalarda Kaggle veritabanından elde edilen Kredi Kartı Dolandırıcılık Teşhis veri seti kullanılarak Çok Katmanlı Yapay Sinir Ağı ve Naive Bayes yöntemleri ile modelleme yapılmıştır.

Bu çalışma sonucunda kişilerin kredi kartlarını kullanma zaman aralıkları Naive Bayes ve çok katmanlı yapay sinir ağları yöntemleriyle analiz edilerek yaptıkları işlemin farklı kişi tarafından yapıldığını tespit etmek amaçlanmıştır. Sonuç olarak birbirlerine kıyasla çok katmanlı yapay sinir ağlı ile daha yüksek bir başarı oranı edilmiştir [15].

2.15. Credit Card Fraud Detection Using Machine Learning: A Study

Bu makalede kredi kartının tanımından ve kullanım alanlarından bahsedilerek genel bir giriş yapılmıştır. Kredi kartı dolandırıcılıkları kategorilerine ayrılmış ve açıklanmıştır. Bunlar; Kart sahibi ile, satıcı tarafından ve zorla el koyarak dolandırıcılık şeklindedir. Dünyanın hızla dijitalleşmeye doğru ilerlemesi ve para işlemlerinin nakit kullanmadan gerçekleşmeye artarak devam etmesiyle birlikte kredi kartı kullanımı oldukça artmıştır. Bununla ilişkili dolandırıcılık faaliyetleri artmakta ve bu da banka ve şirketler için büyük bir kayba yol açmaktadır. Bu nedenle dolandırıcılık işlemleri, dolandırıcılık olmayan normal işlemlerden analiz edip tespit etmek gerektiğinden bahsedilmiştir. Bu makalede dolandırıcılık olaylarını farkedip önüne geçmek için kullanılabilecek yöntemlerden bahsedilmiştir. Bu metodolojiler arasında Gizli Markov Modeli (HMM), Karar Ağaçları, Random Trees, Bayesian Belief Networks, Genetik algoritmalar, Lojistik Regresyon, Destek Vektör Makineleri (SVM), KNN, Fuzzy Clustering ve Neural Networks bulunmaktadır. Makalede bu tekniklerin kapsamlı bir analizi sunulmaktadır.

Sonuç olarak bu makale, kredi kartı dolandırıcılık işlemini anlamamızı ve dolandırıcılık işlemini, normal işlemlerden ayırmamıza yardımcı olabilecek algoritmaları tanımamızı, aralarındaki farkları, avantaj ve dezavantajlarıyla anlatarak doğru algoritmayı seçme konusunda yardımcı olmaktadır [16].

2.16. Credit Card Fraud Detection in e-Commerce: An Outlier Detection Approach

Bu makalede dolandırıcılık ve spam e-posta belirleme görevlerindeki zorluğun uygun denetimli öğrenme modellerini eğitmek için gereken tüm olası kalıpların olmamasından kaynaklandığı vurgulanmıştır. Bu sorun, dolandırıcılık örneklerinin az ve aynı zamanda zamanla değiştiği durumlar olduğundan dolayı daha da göze çarpmaktadır. Dolandırıcılık modelindeki değişiklik, dolandırıcıların sahtekarlığı önlemek için alınan önlemleri atlatmak için yeni yollar geliştirmeye devam etmesinin sonucu olarak ortaya çıkmaktadır. Sınırlı veri ve sürekli değişen yöntemler makinelerin öğrenmesini zorlaştırmaktadır. Bu makale kapsamında yasal davranışın zamanla değişmediğini ve yasal davranışı temsil eden veri noktalarının farklı gruplamalar altında tutarlı uzamsal imzaya sahip olduğunu varsayılmıştır. Bu hipoteze dayanarak, bir clustering yöntemleri topluluğu kullanarak her bir veri noktasına bir tutarlılık puanı atayarak büyük veri kümelerindeki aykırı değerleri tespit eden bir yaklaşım önerilmiştir. Bu çalışmadaki asıl hedef, büyük veri kümelerindeki aykırı değerleri tespit edebilen ve değişen dolandırıcılık kalıplarına karşı dayanıklı yeni bir yöntem önermektir. Ayrıca bu makalede, aykırı değer saptama yöntemlerini değerlendirmek için yaygın olarak kullanılan bir metrik olsa da, ROC eğrisi altındaki alanın doğru metrik olmadığı savunulmaktadır. Aykırı değer algılama sorunlarının sınıfların çarpık dağılımına sahip olduğundan, kesinlik-hatırlama eğrileri daha uygundur çünkü kesinlik, yanlış pozitifleri gerçek negatiflerle (aykırı değerler) karşılaştırır ve bu nedenle sınıf dengesizliği sorunundan etkilenmez. Makalenin devamında kesinlik-hatırlama eğrisinin altındaki alanın bir değerlendirme ölçüsü olarak ROC'den daha iyi olduğu belirtilmektedir.

Veri Setleri olarak; Landsat uydu veri setinin değiştirilmiş versiyonu, ann-tiroid veri setinin değiştirilmiş versiyonu ve Kaggle aracılığıyla mevcut olan büyük bir gerçek dünya kredi kartı dolandırıcılık tespit veri setleri kullanılmıştır [17].

2.17. Detection of Credit Card Fraud in E-Commerce Using Data Mining

Bu makalede e-ticaret sitelerinin en çok tercih edilen yönteminin kredi kartları ile olduğu, kredi kartı bilgilerinin çalınması halinde de dolandırıcılık işlemlerinin gerçekleşebileceği vurgulanmıştır. Dolandırıcılık işleminin önüne geçmek için siparişlerin incelenmesi gerektiği üzerine durulmuştur. Dolandırıcılık şüphesi olan siparişler, sadece dolandırılan kişi ve banka için değil aynı zamanda alışveriş siteleri için de büyük endişe kaynağıdır. Sahtekarlık işlemleri sadece müşterileri değil, aynı zamanda şirketleri ve bankaları da etkiler. Bu nedenle, e ticaret siteleri siparişleri kategorize etmeli ve şüpheli işlemlere karşı önlemler almalı ve üzücü bir durum yaşanmadan onu engelleyebilmelidir. E-ticaret sitelerinde müşteriler hakkında daha az bilgi olması nedeniyle sınıflandırma işlemini yapabilmek daha zordur. Bu makale kapsamında yapılan çalışmada, bir e-ticaret sitesinin gerçek sipariş verileri incelenmiş ve şüpheli işlemler belirlenmiştir. Öncelikle, tüm sipariş verileri analiz edilip, filtrelenmiştir. Sınıflandırma için en iyi değişkenler değişken seçim algoritmaları ile belirlenmiştir. Daha sonra sınıflandırma algoritmaları uygulanmış ve %92 başarı oranı ile şüpheli siparişler belirlenmiştir. Karşılaştırmalı veri madenciliği yöntemleri olarak Naive Bayesian, Karar Ağaçları ve Yapay Sinir Ağı kullanılmıştır.

Sonuç olarak bu makale kapsamında e-ticaret sitesi için veri elde etmek amacıyla manuel olarak dolandırıcılık işlemleri yapılmıştır. Bu veriler sayesinde yapılan çalışmalar sonucunda elde edilen sınıflandırma verileri ile e-ticaret sitelerinin iş süreçlerini büyük ölçüde iyileştirmesi hedeflenmiştir [18].

2.18. Credit Card Fraud Detection Using Convolutional Neural Networks

Bu makalede, dolandırıcılık faaliyetlerini tespit etmek için derin öğrenme tekniklerinin kullanımı anlatılmaktadır. Özellikle kredi kartı dolandırıcılığı tespiti için Convolutional Neural Networks (CNN) kullanımından bahsedilmektedir. Makale kapsamında yapılan çalışmada kredi kartı işlem verileri toplanır ve öncelikle öznitelik teknikleri kullanılarak işlenerek, CNN'nin eğitimi için uygun hale getirilmiştir. Ardından, önerilen CNN mimarisi kullanılarak, kredi kartı dolandırıcılığı tespiti için bir sınıflandırma modeli oluşturulmuştur. Makalede, önerilen yöntemin performansı, doğruluk, hassasiyet, özgüllük ve F1 puanı gibi performans metrikleri kullanılarak değerlendirilmiştir. Sonuçlar olarak ise, önerilen CNN yönteminin, kredi kartı dolandırıcılığı tespiti için mevcut diğer yöntemlerden daha iyi performans gösterdiği gösterilmiştir [19].

2.19. Recurrent Neural Networks with Attention for Credit Card Fraud Detection

Bu makalede, kredi kartı dolandırıcılığı tespiti için tekrarlayan sinir ağları (RNN) ve dikkat mekanizmalarını bir arada kullanarak yeni bir yöntem önerilmektedir. Makalede, kredi kartı işlem verileri toplandı ve öznitelik teknikleri kullanılarak işlenmiştir. Daha sonra, özellikle LSTM (uzun-kısa süreli bellek) tabanlı RNN'ler kullanılarak dolandırıcılık tespiti için bir sınıflandırma modeli oluşturulmuştur. Bunun yanı sıra, dikkat mekanizmaları kullanılarak, modelin daha spesifik özniteliklere odaklanması sağlandı. Dikkat mekanizmaları, her işlem verisinin farklı ağırlıklandırılmış öznitelikleri üzerinde yoğunlaşarak, modelin dolandırıcılık tespitinde daha hassas ve doğru hale gelmesine yardımcı oldu. Makalede, önerilen yöntemin performansı, doğruluk, hassasiyet, özgüllük ve F1 puanı gibi performans metrikleri kullanılarak değerlendirilmiştir. Sonuç olarak, önerilen RNN modelinin, kredi kartı dolandırıcılığı tespiti için mevcut diğer yöntemlerden daha iyi performans gösterdiğini gösterdi [20].

2.20. A Deep Learning Approach to Credit Card Fraud Detection

Bu makalede, kredi kartı dolandırıcılığı tespiti için derin öğrenme tekniklerinin kullanımını araştırmaktadır. Yöntem olarak, sahte işlemleri tespit etmek için bir sınıflandırma modeli kullanılmaktadır. Makalede, kredi kartı işlem verileri toplanmış ve öncelikle öznitelik teknikleri kullanılıp işlenerek bir derin öğrenme modelinin eğitimi için uygun hale getirilmiştir. Bu amaçla MLP ve derin öğrenme yöntemleri olan autoencoder ve deep belief network kullanılmıştır. Önerilen yöntem, kredi kartı işlemlerini doğru şekilde sınıflandırmak

için iki aşamalı bir model kullanır: öncelikle, işlemin doğal olup olmadığını belirlemek için bir MLP kullanılır, ardından, işlem dolandırıcılığına karar vermek için autoencoder veya deep belief network kullanılmaktadır. Makalede, önerilen yöntemin performansı, doğruluk, hassasiyet, özgüllük ve F1 puanı gibi performans metrikleri kullanılarak değerlendirildi. Sonuçlar, önerilen derin öğrenme yönteminin, kredi kartı dolandırıcılığı tespiti için mevcut diğer yöntemlerden daha iyi performans gösterdiği belirtilmiştir [21].

2.21. Deep Learning for Credit Card Fraud Detection

Bu makalede, Han ve diğerleri, kredi kartı dolandırıcılığı tespiti için derin öğrenme yöntemlerinin kullanımını araştırmaktadır. Bu yöntemler arasında, özellikle çoklu öğrenme yöntemleri kullanarak dolandırıcılık işlemlerinin tespiti için bir sınıflandırma modeli geliştirdiler. Makalede, öncelikle, kredi kartı işlem verileri toplandı ve öznitelik mühendisliği teknikleri kullanılarak işlenerek, bir derin öğrenme modelinin eğitimi için uygun hale getirildi. Daha sonra, sınıflandırma modeli için çeşitli çoklu öğrenme yöntemleri, özellikle de kombine (ensemble) yöntemler kullanılarak, doğal ve sahte işlemlerin ayrımı yapmak üzere eğitildi. Ayrıca, bu çalışmada, dolandırıcılık tespitindeki hataların nedenleri ve sınıflandırma performansını etkileyen faktörler de incelenmiştir. Bu faktörler arasında, işlem miktarı, dolandırıcılık türü ve veri dengesizliği gibi faktörler yer almaktadır. Makalede, önerilen yöntemin performansı, doğruluk, hassasiyet, özgüllük ve F1 puanı gibi performans metrikleri kullanılarak değerlendirildi. Sonuçlar, önerilen çoklu öğrenme yöntemlerinin, kredi kartı dolandırıcılığı tespiti için mevcut diğer yöntemlere göre daha yüksek bir doğruluk oranı sağladığını gösterdi. Sonuç olarak, Han ve diğerleri, çoklu öğrenme yöntemlerinin kredi kartı dolandırıcılığı tespiti için etkili bir araç olduğunu öne sürüyorlar ve bu alanda daha fazla araştırma yapılması gerektiğini vurguluyorlar. Ayrıca, dolandırıcılık tespiti için veri dengesizliği gibi faktörlerin de dikkate alınması gerektiğini belirtiyorlar [22].

2.22. A Hybrid Deep Learning Framework for Fraud Detection in Credit Card Transactions

Bu makalede, kredi kartı işlem dolandırıcılığının tespiti için hibrit bir derin öğrenme çerçevesi öneriyorlar. Bu çerçeve, birleşik bir öznitelik çıkarımı ve sınıflandırma modeli içerir. Öncelikle, veri toplama ve ön işleme adımları gerçekleştirildi. Daha sonra, öznitelik mühendisliği teknikleri kullanılarak, işlem verileri uygun bir şekilde işlendi ve derin öğrenme modeli için uygun hale getirildi. Bu öznitelikler, işlem miktarı, işlem tarihi ve saat gibi temel bilgilerin yanı sıra, işlem yerinin coğrafi konumu, işlem tutarı ve önceki işlem geçmişi gibi diğer bilgileri de içermektedir. Daha sonra, önerilen hibrit derin öğrenme çerçevesi, iki adımdan oluşan bir süreçte uygulandı. İlk olarak, önceden eğitilmiş bir derin öğrenme ağı, öznitelikleri kullanarak sahte ve gerçek işlemleri ayırmak için eğitildi. Ardından, sınıflandırma modeli için özellikle tasarlanmış bir derin öğrenme ağı kullanılarak, dolandırıcılık tespiti için işlem verileri sınıflandırıldı. Sonuçlar, önerilen hibrit derin öğrenme

çerçevesinin, kredi kartı dolandırıcılığı tespiti için yüksek bir doğruluk oranı sağladığını gösterdi. Ayrıca, bu çerçevenin, diğer mevcut dolandırıcılık tespit yöntemlerine kıyasla daha yüksek bir hassasiyet ve özgüllük oranı da sağladığı belirlendi [23].

2.23. A Comparative Study on Credit Card Fraud Detection Using Deep Learning Techniques

Bu makalede, kredi kartı dolandırıcılığı tespiti için derin öğrenme tekniklerinin performansını karşılaştıran bir çalışma yapılmıştır. Çalışma, farklı derin öğrenme teknikleri arasındaki performans farklarını değerlendirmek için üç farklı veri kümesi kullanarak yapıldı. Kullanılan teknikler arasında derin sinir ağları (DNN), CNN ve rekürrent sinir ağları (RNN) bulunmaktadır. Veri kümesi, işlem verileri, işlem tarihi, işlem yerinin coğrafi konumu gibi temel bilgilerin yanı sıra, işlem tutarı, önceki işlem geçmişi ve işlem açıklamaları gibi ek bilgiler içeriyordu. Çalışma sonuçları, RNN'nin diğer tekniklere kıyasla en yüksek dolandırıcılık tespiti başarısı oranına sahip olduğunu gösterdi. Ayrıca, tüm tekniklerin doğruluğunun arttığı, ancak RNN'nin en hızlı şekilde doğruluk oranını arttırdığı görüldü. Daha fazla veri kullanarak yapılan denemelerde, tüm tekniklerin doğruluğunun arttığı ve DNN'nin diğer tekniklerle benzer bir performans gösterdiği belirlendi [24].

2.24. Autoencoder-Based Fraud Detection in Credit Card Transactions

Bu makalede, kredi kartı dolandırıcılığı tespiti için otomatik kodlayıcı (autoencoder) tabanlı bir yöntem öneren bir çalışma yaptılar. Bu çalışmada, kredi kartı işlemlerinden oluşan bir veri kümesi kullanıldı. Autoencoder, veri kümesindeki normal işlemleri öğrenmek için eğitildi. Dolandırıcılık işlemleri, autoencoder'ın normal işlemleri tanımlayamayacağı şekilde yapılandırıldı. Autoencoder, normal işlemlerle dolandırıcılık işlemleri arasındaki farkı tespit edebilen bir sınıflandırıcı olarak kullanıldı. Çalışma sonuçları, autoencoder tabanlı yöntemin, diğer yöntemlere kıyasla yüksek doğruluk oranları ve düşük yanlış pozitif ve yanlış negatif oranları ile dolandırıcılık tespiti için etkili bir yöntem olduğunu gösterdi. Autoencoder ayrıca, diğer yöntemlere kıyasla daha az sayıda yanlış pozitif ve yanlış negatif sonuç ürettiği için daha az insan müdahalesi gerektiğini belirtti [25].

2.25. A Deep Learning Framework for Fraud Detection in Financial Statements

Bu makalede, finansal tablolardaki dolandırıcılık faaliyetlerini tespit etmek için derin öğrenme tabanlı bir çerçeve öneren bir çalışma yaptılar. Çalışmada, dolandırıcılık faaliyetlerinin tespiti için finansal tablolardaki değişkenlerin (örneğin, varlıklar, borçlar, gelirler vb.) zaman serisi verileri kullanıldı. Bu veriler, derin öğrenme algoritmalarının (özellikle RNN) kullanılması için düzenlendi. Ayrıca, dolandırıcılık faaliyetlerinin tespiti için RNN'lerin yanı sıra CNN'ler de kullanıldı. Çalışma sonuçları, önerilen derin öğrenme

çerçevesinin finansal tablolardaki dolandırıcılık faaliyetlerini tespit etmek için yüksek doğruluk oranları ve düşük yanlış pozitif ve yanlış negatif oranları sağladığını gösterdi. Ayrıca, RNN'ler ve CNN'lerin bir arada kullanılmasının, dolandırıcılık faaliyetlerinin tespitinde daha iyi sonuçlar verdiği bulunmuştur [26].

2.26. Deep Learning-Based Fraud Detection in Bitcoin Transactions

Bu makalede, Bitcoin işlemlerinde dolandırıcılık faaliyetlerini tespit etmek için derin öğrenme tabanlı bir yöntem öneren bir çalışma yaptılar. Çalışmada, Bitcoin işlemlerinin yapısı göz önünde bulundurularak, işlem özellikleri (örneğin, giriş ve çıkış adresleri, işlem miktarları vb.) zaman serisi olarak düzenlendi. Bu zaman serisi verileri, dolandırıcılık faaliyetlerini tespit etmek için LSTM ve GRU gibi derin öğrenme algoritmaları kullanılarak analiz edildi. Önerilen yöntem, dolandırıcılık faaliyetlerini tespit etmek için geleneksel yöntemlerden daha yüksek doğruluk oranları sağladı. Ayrıca, yöntemin zaman serisi verilerinin özelliklerini dikkate alarak tasarlanması, performansının artmasına yardımcı oldu [27].

2.27. Deep Learning for Credit Card Fraud Detection: A Comparative Analysis of Classifiers

Bu makalede, kredi kartı dolandırıcılığı tespiti için derin öğrenme yöntemlerinin performansını karşılaştıran bir çalışma yaptılar. Çalışmada, birçok farklı derin öğrenme algoritması kullanılarak kredi kartı dolandırıcılığı tespiti için modeller eğitildi ve bu modellerin performansı karşılaştırıldı. Kullanılan algoritmalar arasında çok katmanlı algılayıcı (MLP), evrişimli sinir ağı (CNN), uzun-kısa süreli bellek (LSTM) ve çift yönlü LSTM (Bi-LSTM) bulunmaktadır. Deneyler, CNN ve Bi-LSTM gibi derin öğrenme yöntemlerinin diğer yöntemlere göre daha yüksek doğruluk oranlarına sahip olduğunu göstermiştir. Ayrıca, derin öğrenme modellerinin performansının, geleneksel makine öğrenimi yöntemlerine kıyasla daha iyi olduğu da belirtilmiştir [28].

2.28. Anomaly Detection in Credit Card Transactions Using Autoencoders and Local Outlier Factor

bu makalede kredi kartı işlemlerinde anormallik tespiti için otomatik kodlayıcılar ve yerel aykırı faktör (LOF) kullanarak bir yöntem önerdiler. Çalışmada, öncelikle bir otomatik kodlayıcı modeli eğitilerek normal kredi kartı işlemleri kodlandı. Daha sonra, LOF yöntemi kullanılarak normal olmayan işlemler tespit edildi. Deneyler, önerilen yöntemin hem doğruluk hem de algılama oranı açısından diğer yöntemlere göre daha iyi performans gösterdiğini göstermiştir. Ayrıca, LOF yönteminin, diğer aykırılık tespiti yöntemlerine kıyasla daha az yanlış pozitif sonuç ürettiği de belirtilmiştir [29].

2.29. Fraud Detection in Online Transactions Using Autoencoders and Logistic Regression

Bu makalede, çevrimiçi işlemlerde dolandırıcılık tespiti için otomatik kodlayıcılar ve lojistik regresyon kullanarak bir yöntem önermişlerdir. Çalışmada, öncelikle normal kredi kartı işlemleri otomatik kodlayıcı kullanılarak kodlandı. Daha sonra, lojistik regresyon kullanılarak normal olmayan işlemler tespit edildi. Yöntemin etkinliğini değerlendirmek için, gerçek bir veri kümesi kullanılarak deneysel çalışmalar yapılmıştır. Sonuçlar, önerilen yöntemin diğer yöntemlere göre daha yüksek doğruluk oranı ve daha az yanlış pozitif sonuç ürettiğini göstermiştir [30].

2.30. Credit Card Fraud Detection with Deep Convolutional Neural Networks

Bu makalede, derin evrişimli sinir ağları (CNN) kullanarak kredi kartı dolandırıcılığı tespiti için bir yöntem önermişlerdir. Çalışmada, öncelikle bir CNN modeli eğitilerek normal ve dolandırıcılık işlemleri ayrı ayrı öğrenildi. Daha sonra, test verileri kullanılarak modelin doğruluğu değerlendirildi. Deneyler, önerilen yöntemin diğer yöntemlere göre daha yüksek doğruluk oranı ve daha düşük yanlış pozitif sonuç ürettiğini göstermiştir. Ayrıca, önerilen yöntem, dolandırıcılık işlemlerini diğer normal işlemlerden daha iyi tespit etmiştir [31].

2.31. Anomaly Detection in Financial Transactions Using Deep Learning

Bu makalede, finansal işlemlerdeki anormallikleri tespit etmek için derin öğrenme kullanarak bir yöntem önermişlerdir. Önerilen yöntem, önce finansal verilerin özellikleri belirlenerek özellik mühendisliği yapılmıştır. Daha sonra, bir dizi derin öğrenme modeli (autoencoder, LSTM, GRU) kullanılarak işlemlerdeki anormallikler tespit edilmiştir. Deneyler, önerilen yöntemin diğer yöntemlere kıyasla daha yüksek doğruluk oranları ve daha düşük yanlış pozitif sonuçlar verdiğini göstermiştir. Ayrıca, özellikle autoencoder modelinin finansal işlemlerdeki anormallikleri başarıyla tespit ettiği görülmüştür [32].

2.32. Fraud Detection in Insurance Claims Using Recurrent Neural Networks

Bu makalede, sigorta taleplerindeki dolandırıcılıkları tespit etmek için tekrarlayan sinir ağlarını (RNN) kullanarak bir yöntem önermişlerdir. Önerilen yöntemde, sigorta taleplerinin özellikleri öncelikle belirlenerek özellik mühendisliği yapılmıştır. Daha sonra, RNN modeli, özelliklerin zaman serisi olarak temsil edilmesi için kullanılmıştır. Bu şekilde, taleplerdeki anormallikler zaman içindeki değişikliklerle birlikte tespit edilmiştir. Deneyler, önerilen yöntemin diğer yöntemlere kıyasla daha yüksek doğruluk oranları ve daha düşük yanlış pozitif sonuçlar verdiğini göstermiştir. Ayrıca, özellikle RNN modelinin sigorta taleplerindeki dolandırıcılıkları başarıyla tespit ettiği görülmüştür [33].

2.33. Credit Card Fraud Detection Using Deep Autoencoder Neural Networks

Bu makalede, kredi kartı dolandırıcılığı tespiti için derin otokodlayıcı sinir ağları (DAN) kullanarak bir yöntem önermişlerdir. Önerilen yöntem, kredi kartı işlemlerinden elde edilen verileri kullanarak bir DAN modeli eğitir. Bu model, normal işlemleri öğrenerek, dolandırıcılık işlemlerini tespit etmek için anormallikleri belirlemeye çalışır. Yazarlar, önerilen yöntemi gerçek bir veri kümesi üzerinde test ettiler ve doğruluk oranlarının% 98,2'ye kadar yükseldiğini ve yanlış pozitif sonuçların düşük olduğunu gösterdiler [34].

2.34. A Deep Learning Approach for Fraud Detection in Healthcare Claims

Bu makalede, sağlık hizmeti taleplerinde dolandırıcılık tespiti için bir derin öğrenme yaklaşımı önermişlerdir. Bu makalede, yazarlar, sağlık sigortası verilerinin dolandırıcılık tespiti için kullanılabileceğini belirtmektedirler. Yazarlar, dolandırıcılık tespitinde kullanılan geleneksel yöntemlerin sınırlamalarını açıklarlar ve bu nedenle derin öğrenme yöntemlerinin bu alanda kullanımına öncelik verirler. Yazarlar, derin bir sinir ağı modeli kullanarak dolandırıcılık tespiti için bir yöntem önerirler. Bu model, sağlık hizmeti taleplerini analiz ederek dolandırıcılık işlemlerini tespit etmeye çalışır. Yazarlar, modelin, geleneksel yöntemlere kıyasla daha yüksek doğruluk oranlarına sahip olduğunu ve dolandırıcılık tespiti için daha etkili bir araç olabileceğini belirtmektedirler [35].

2.35. Deep Learning for Fraud Detection in Mobile Money Transactions

Bu makale, mobil para işlemlerinde dolandırıcılığı tespit etmek için derin öğrenmenin kullanımını ele almaktadır. Yazarlar, mobil para transferleri için çeşitli algoritmalar ve teknikler kullanarak bir dolandırıcılık tespit sistemi geliştirmeyi amaçlamaktadır. Makalede, mobil para işlemlerinin özellikleri analiz edilerek, dolandırıcılık davranışlarının tanımlanmasında kullanılabilecek özellikler belirlenmiştir. Bu özellikler daha sonra derin öğrenme modellerinde kullanılmak üzere işlenmiştir. Deneysel sonuçlar, geliştirilen sistemin yüksek doğruluk oranlarına sahip olduğunu ve farklı dolandırıcılık senaryolarını tespit edebildiğini göstermektedir. Bu çalışma, mobil para işlemleri için dolandırıcılık tespiti için derin öğrenmenin kullanımının faydalı olduğunu göstermektedir [36].

2.36. Unsupervised Deep Learning for Fraud Detection in Credit Card Transaction Data

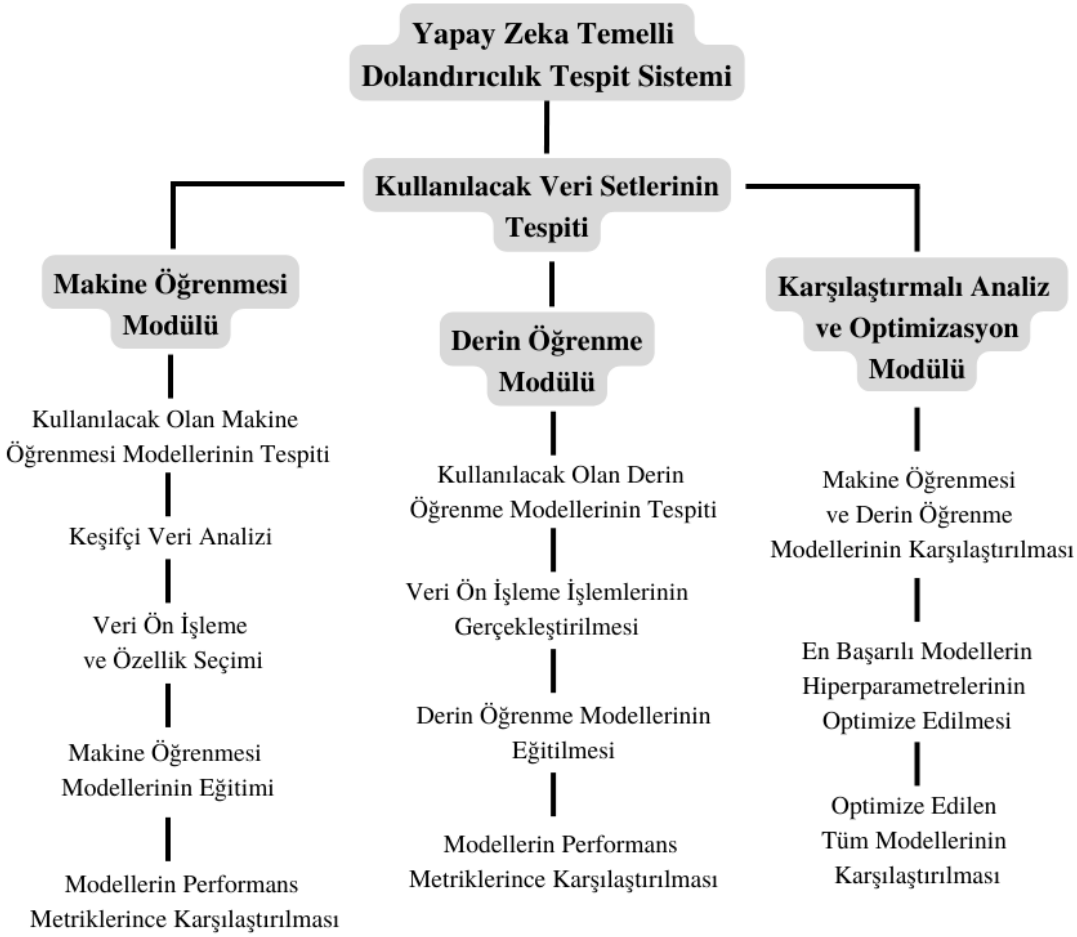
Bu makalede, kredi kartı işlem verilerindeki dolandırıcılığı tespit etmek için kullanılabilecek bir unsupervised (denetimsiz) derin öğrenme yöntemi olan Autoencoder Variational Bayesian Anomaly Detection (AVB-AD) önerilmektedir. AVB-AD, bir autoencoder modelini kullanarak normal işlemler için bir çıktı üretir ve daha sonra beklenen çıktıyı hesaplamak için

varyasyonel bayesian yöntemini kullanır. Bu yöntem, anomalileri tespit etmek için bir eşik değeri belirler ve bunları dolandırıcılık olarak etiketler. Yöntem, gerçek dünya verileri üzerinde test edilmiştir ve diğer yöntemlerle karşılaştırıldığında yüksek bir doğruluk oranı elde edilmiştir [37].

2.37. Autoencoder-Based Fraud Detection System for Electricity Market

Bu makale, otomatik kodlayıcılar kullanarak elektrik piyasalarındaki dolandırıcılık tespiti için bir sistem önermektedir. Bu sistem, tarihsel verileri kullanarak normal tüketim kalıplarını öğrenir ve bunları gelecekteki tüketimlerle karşılaştırarak anormallikleri tespit eder. Ayrıca, birbirinden bağımsız iki farklı otomatik kodlayıcı kullanılarak daha iyi bir doğruluk elde edilir. Araştırma sonuçları, önerilen sistemin mevcut dolandırıcılık tespit yöntemlerinden daha iyi performans gösterdiğini göstermektedir [38].

3. GELİŞTİRİLEN YAKLAŞIM VE BULGULAR



Tablo 3.1. Yapay Zeka Temelli Dolandırıcılık Tespit Sistemi Gösterimi

3.1. Kullanılacak Veri Setlerinin Araştırılması/Tespiti

İlk ve en önemli adımlardan biri olan çalışılacak veri setlerinin bulunması üzerine Kaggle [39], Google Dataset Search [40], UCI Machine Learning Repository [41], OpenML [42], DataHub [43], Papers with Code [44], EU Open Data Portal [45], Awesome Public Datasets [46] gibi çeşitli kaynaklardan detaylı bir araştırma süreci gerçekleştirilmiştir. Bu doğrultuda bulunan veri setlerinin özellikleri aşağıdaki tabloda (Tablo 3.2.) gösterilmiştir. Hem e-ticaret ve bankacılık olmak üzere çeşitli sektörde kullanılan hem de akademik kaynaklardan elde edilen veri setlerinin yapısal olarak incelenmesi ve işleme sürecinden önce veriler üzerinde yorum yapılabilmesi amacıyla veri setinin tanınması amaçlanmıştır. Tanıma süreci; satır ve sütun özelliklerinin/sayılarının, veri tiplerinin, eksik/aykırı yahut tekrar eden değerlerin, veri dengesizliğinin ve verilerin betimsel istatistik değerlerinin incelenmesi vb. şeklinde ilerlemiştir. Bir sonraki aşamada ise elde edilen bilgiler görselleştirilerek veri setleri hakkında genel bilgi edinilmesi sağlanmıştır.

İsim	Sektör	Boyut (satır, sütun)	Özellikler	Gizli mi?	Referanslar
Credit Card Fraud Detection [27]	Bankacılık	(284807, 31)	Time, Amount, Class, V1-V28*	Evet	[28-43]
Credit Card Transactions [44]	Bankacılık	(19999999, 15)	User, Card, Year, Month, Day, Time, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, Zip, MCC, Errors, Is Fraud?	Hayır	[45]
Simulated Credit Card Transactions generated using Sparkov [46]	Bankacılık	(1296675, 23), (555719, 23)	trans_date_trans_time, cc_num, merchant, category, amt, first, last, gender, street … lat, long, city_pop, job, dob, trans_num, unix_time, merch_lat, merch_long, is_fraud	Hayır	[45-48]
Fraud E- Commerce [49]	E-Ticaret	(151112, 11), (138846, 3)	user_id,, signup_time, purchase_time, purchase_value, device_id, source, browser, sex, age, ip_address, class	Hayır	[48]
IEEE CIS Fraud Detection [50]	E-Ticaret	(590540,433) (506691,432)	TransactionDT, TransactionAMT, ProductCD, card1- card6, addr, dist, email domain, C1-C14*, D1-D15*, M1-M9*, Vxxx …	Evet	[42]
Synthetic Data from a Financial Payment System [51]	Bankacılık	(594643, 10)	step, customer, age, gender, zipcodeOri, merchant, category, zipMerchant, amount, fraud	Hayır	[52-53]
Synthetic Financial Datasets For Fraud Detection [54]	Bankacılık	(6362620, 11)	PAYMENT, step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud	Hayır	[55-56]

Tablo 3.1.1. Literatür taraması sonucunda en çok kullanılan veri setlerinin karşılaştırılması

* Gizlilik nedeniyle sütunların ne olduğu bilinmemekte.

3.2. Makine Öğrenmesi Modülü

Bu bölümde gerçekleştirilen çalışma bünyesinde kullanılan makine öğrenmesi algoritmaları tanımlanacak ve çalışmada kullanılan modellerinin performans metrikleri ve çıktıları yorumlanacaktır.

3.2.1. Kullanılacak Olan Makine Öğrenmesi Modellerinin Tespiti

Literatür taramasında detaylıca araştırılmış olan makaleler tablo haline getirilmiş olup (Tablo 3.2.) karşılaştırmak amacıyla en fazla kullanılan algoritmalar renklendirilmiştir. Akademik çalışmalarda en fazla kullanılan algoritmalar **Logistic Regression, Random Forest Model, Support Vector Machines, K-Nearest Neighbor, Decision Tree, Naive Bayes** olarak belirlenmiştir ve bu algoritmaların kullanılmasına karar verilmiştir.

Makale / Algoritmalar	logistic regression	neural network	random forest model	Support Vector Machines	K-Nearest Neighbor	Decision Tree	Naive Bayes
APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions [1]	x	x	x				
Critical analysis of machine learning based approaches for fraud detection in financial transactions [3]				x	x		
A cost-sensitive decision tree approach for fraud detection [5]		x		x			
An efficient real time model for credit card fraud detection based on deep learning [6]	x	x		x			
A Review of Credit Card Fraud Detection Techniques [7]	x	x	x	x	x	x	
Real time credit card fraud detection on huge imbalanced data using meta-classifiers [8]			x			x	
Real-time credit card fraud detection using machine learning [9]	x			x			x
Data mining for credit card fraud: A comparative study [10]	x		x	x			
Yapay Zeka Destekli ÇOKS Yöntemi ile Kredi Kartı Sahtekarlığının Tespiti [11]					x		x
Credit Card Fraud Detection using Machine Learning Algorithms [12]	x		x			x	
Kredi Kartı Dolandırıcılık Tespitinin Makine Öğrenmesi Yöntemleri ile Tahmin Edilmesi [15]		x					x
Credit Card Fraud Detection using Machine Learning: A Study [16]	x	x	x	x	x	x	
Detection of Credit Card Fraud in E-Commerce Using Data Mining [18]		x				x	x
Approaches to Fraud detection on credit card transactions using artificial intelligence methods [19]			x	x	x	x	x

Tablo 3.2.1.1. Literatürde en fazla değinilen makine öğrenmesi algoritmaları

3.2.1.1. Logistic Regression

Lojistik regresyon, çoğunlukla ikili sınıflandırma problemlerinde kullanılan istatiksel bir algoritmadır. Amacı; bir olayın öngörülen olasılığını 0-1 arasında bir değer sağlayan sigmoid eğrisi ile sunarak ait olması gereken sınıfın tahmin edilmesini sağlamaktır.

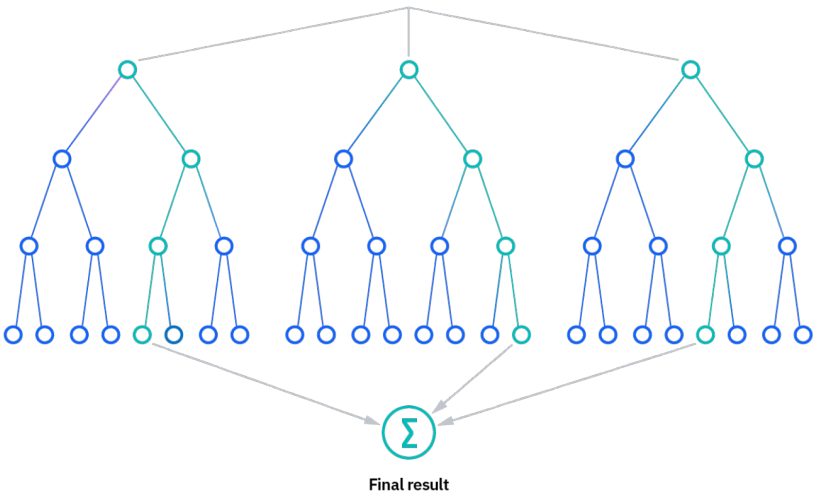
Lojistik regresyon denklemi şu şekilde ifade edilebilir.

$$P = \frac{1}{1+e^{-z}}$$

P: Bağımlı değişkenin bir kategoride olma olasılığını temsil eder.
e: Euler sayısını (e=2.7182818284..) ifade eder.
z: z=(b0 + b1x1 + b2x2 + … + bn*xn) olmak üzere;
b0 kesme terimini ifade eder.
b1,b2,...,bn bağımsız değişknelerin kat sayılarını ifade eder.
x1,x2,....,xn bağımsız değişkenleri ifade eder.

3.2.1.2. Random Forest Model

Hem sınıflandırma hem de regresyon problemlerinde kullanılan, birden fazla desicion tree üreterek bu karar ağaçlarının sonuçlarını birleştirir ve en yüksek puanlı değerin seçilmesini sağlar.



Şekil 3.2.1.2.1. Random Forest gösterimi [57].

Random forest işlemi için, şekilde de görülebileceği üzere [Şekil 3.2.1.2.1.] öncelikle veriler rastgele bir şekilde birden fazla alt kümeye bölünür. Dana

sonra verilerin her bir alt kümesi için Gini impurity formülü kullanılarak karar ağaçları oluşturulur.

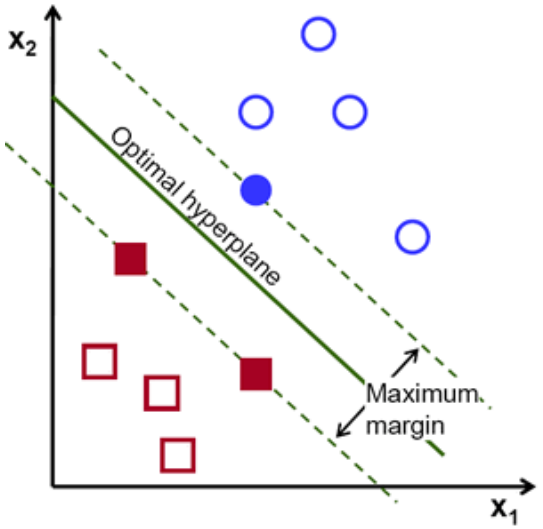
D veri seti için Gini impurity hesaplanması:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

Tüm karar ağaçları oluşturulduktan sonra, her bir ağacın prediction değerleri toplanarak tahmin yapılır. Tüm ağaçların ortalama prediction değerleri nihai tahmin olarak ele alınır.

3.2.1.3. Support Vector Machines

Bir veri kümesini sınıflandırmak amacıyla sanal bir düzlem üzerinde sınır çizgileri çekerek düzlemde bulunan iki grubu ayırmayı amaçlayan bir sınıflandırma algoritmasıdır. Bu ayırma işleminde kullanılacak sınır çizgileri ise şekilx. de görülebileceği gibi iki grubun bileşenlerinin birbirlerine en uzak olduğu yer baz alınarak çizilmelidir.



Şekil 3.2.1.3.1. Support Vector Machines sınır gösterimi [58].

Marjin değerini maksimum olacak şekilde seçebilmek sınıf ayrımının daha sağlıklı olmasını sağlamaktadır. Marjinin belirlenmesi ile ortaya çıkan Optimal hyperplane, gelecek olan yeni verilerin sınıflandırılmasında önemli rol oynar. Düzlemin üstünde/altında veya sağında/solunda kalma durumlarına göre ayrım yapılır. Bu ayrım yapılırken ise düzlemde bulunan her noktanın tanımı şu gösterim ile yapılır;

$$D = \left\{ x_i \in R^p, c_i \in \{-1, 1\} \right\}_{i=1}^{n}$$

Bu denklem ile veri setinde bulunan her üyenin x,c ikilisi için;
x: noktayı
c: noktanın -1 veya +1 olduğu değeridir.

Bu gösterimin iki boyutlu bir düzlem üzerinde olduğunu düşünürsek, bu gösterimdeki her noktayı;

$$wx - b = 0$$

Formülü ile ifade edebiliriz. Burada bulunan;
w: normal vektörü
x: nokta parametresi
b: kayma oranını ifade eder.

Yukarıda bulunan denkleme göre ise maksimum marjin değeri aşağıdaki denklem ile ifade edilebilir.

$$\frac{2}{\|w\|}$$

Bu formülden de yola çıkarak sınıfları birbirinden ayıran sınır çizgileri arasındaki mesafenin 2 birim olarak belirlendiği çıkarımına ulaşabiliriz.

3.2.1.4. K-Nearest Neighbor (KNN)

Yeni eklenmek istenen bir veri noktasının, belirlenen k tane komşu ile kıyaslanıp, bu komşular arasında en yakın mesafeye sahip olanın sınıfına dahil edilmesi ile sınıflandırma işlemi gerçekleştirilir. Burada yapılan mesafe hesaplaması öklid mesafesi, manhattan mesafesi gibi uzaklık ölçütleri kullanılarak yapılmaktadır.

KNN algoritmasındaki k değerinin seçimi overfitting ve aykırı değer hassasiyeti oluşturabileceği için büyük önem taşımaktadır. Optimum k değeri cross-validation yönteminden yararlanılarak belirlenebilir.

3.2.1.5. Decision Tree

Çok sayıda öğesi bulunan bir veri setini önceden belirlenen bir takım karar kurallarının kullanılması ile sınıflandırma sağlayan bir algoritmadır. Bu algoritmanın yapısında kararları ve sınıfları temsil etmek amacıyla düğüm ve dal yapısından faydalanılır. Ağacın son düğümleri ise sınıf etiketlerinin nihai sonuçlarını temsil eder.

Algoritmada sınıflandırma işlemi Gini impurity veya entropy kriterlerine göre yapılmaktadır. Bu kriterlerden uygun olanın seçilmesinden sonra sınıflandırma işlemi için uygun olan özellik belirlenmiş olur. Özellik belirlenmesinin ardından ise bir durumda oluşabilecek maksimum derinlik veya minimum örnek sayısı gibi durdurma kriteri gerçekleşene kadar ağaç dallarının bölünme işlemi devam eder. Ağaç oluşturulduktan sonra veri değerleri ağaç dallarında gezinerek uygun sınıflara atama sağlanır.

3.2.1.6. Naive Bayes

Naive Bayes Classifier algoritmasında yeni bir verinin sınıflandırma işlemi, her durumun olasılığının hesaplanması ve bu hesaplama sonucunda en yüksek olasılık değerine sahip sınıfa atanması ile gerçekleştirilir.

Bu algoritmada kullanılan Bayes Teoremini ifade etek gerekirse;

$$P(B) = \frac{P(A)\,P(A)}{P(B)}$$

$P(B)$: B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı
$P(A)$: A olayının gerçekleşme olasılığı
$P(A)$: A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı
$P(B)$: B olayının gerçekleşme olasılığını ifade eder.

3.2.2. Keşifçi Veri Analizi

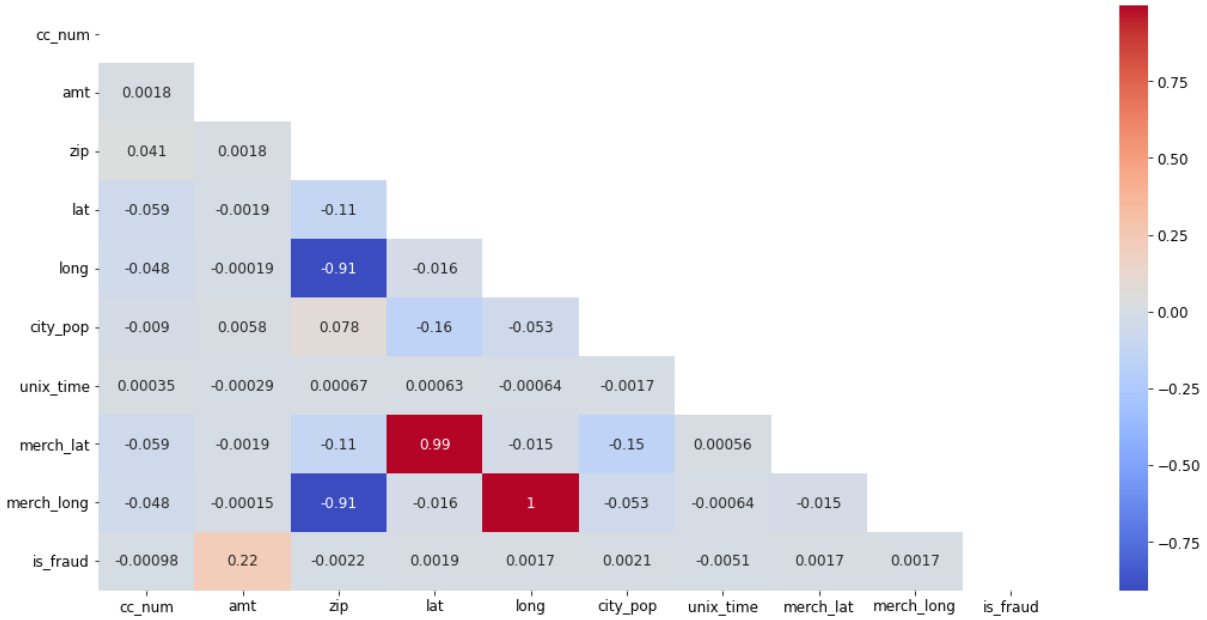
Keşifçi Veri Analizi, veri setini birinci gözden yorumlayabilmek adına kullanılan önemli bir aşamadır. Veri setini anlama, keşfetme ve görselleştirme işlemlerini içerir. Veri setindeki desenleri, ilişkileri, anormallikleri ve trendleri belirlemek için istatistiksel ve görsel tekniklerin kullanılmasını içerir. Veri setinin yapısını, dağılımını, eksik değerlerini, aykırı değerlerini ve ilişkilerini anlamayı amaçlar. Bu sayede veri hakkında önemli bilgiler elde edilir ve daha ileri analizler ve modelleme çalışmaları için sağlam bir temel oluşturulur.

Proje özelinde gerçekleştirilen keşifçi veri analizi aşamaları şunlardır;

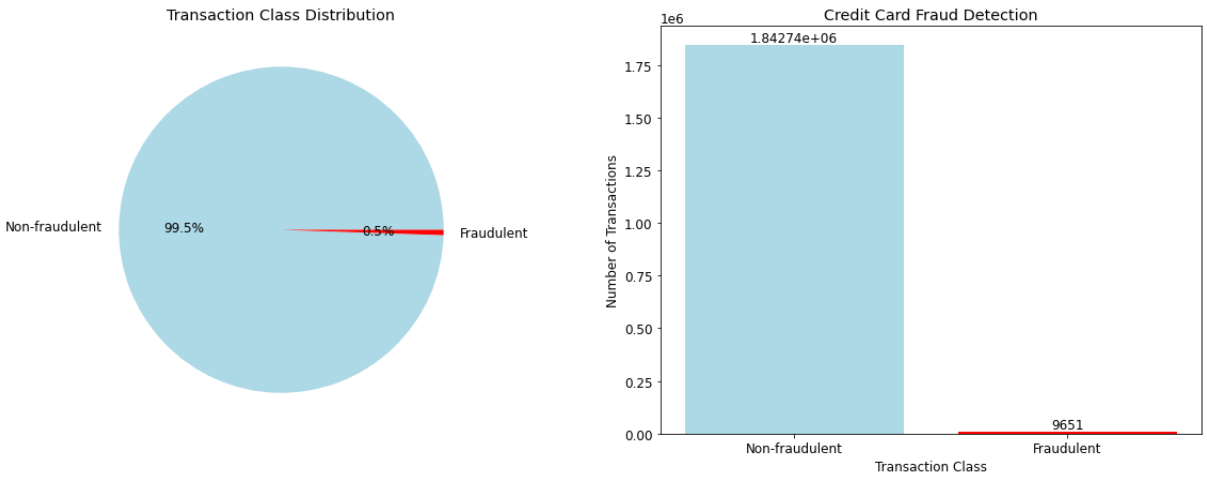
1. Veri setinin genel özelliklerinin anlaşılması: Veri tipleri, boyutları, örneklerin gözlemlenmesi.
2. Veri setindeki eksik veya bozuk verilerin belirlenmesi ve ilgili işlemlerin yapılması.
3. Değişkenler arasındaki ilişkilerin incelenmesi: Korelasyon analizi, dağılım grafikleri, kategorik değişkenlerin çapraz tabloları vb.
4. Veri setinin istatistiksel özetlerinin oluşturulması: Ortalama, medyan, varyans, yüzdelikler, merkezi eğilim ölçüleri vb.

5. Görselleştirme tekniklerinin kullanılması: Histogramlar, kutu grafikleri, scatter plotlar, yoğunluk grafikleri, heat mapler vb.

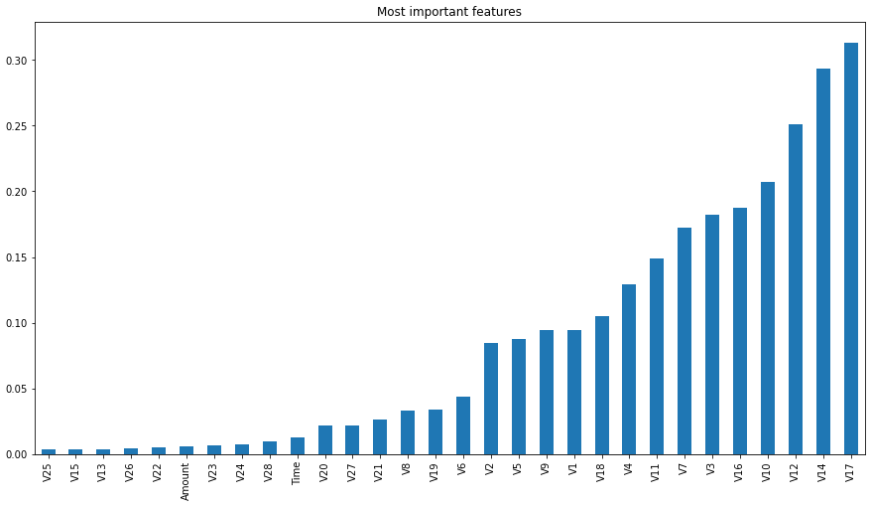
Keşifçi veri analizi sayesinde, veri seti üzerinde yapılan analizler sayesinde veri hataları, yanlış girişler veya anormallikler tespit edilebilir. Ayrıca, modelleme aşamasında hangi değişkenlerin önemli olduğunu ve hangi ilişkilerin daha derinlemesine incelenmesi gerektiğini belirlemek için bu aşamada bulunan sonuçlardan yararlanılabilir.



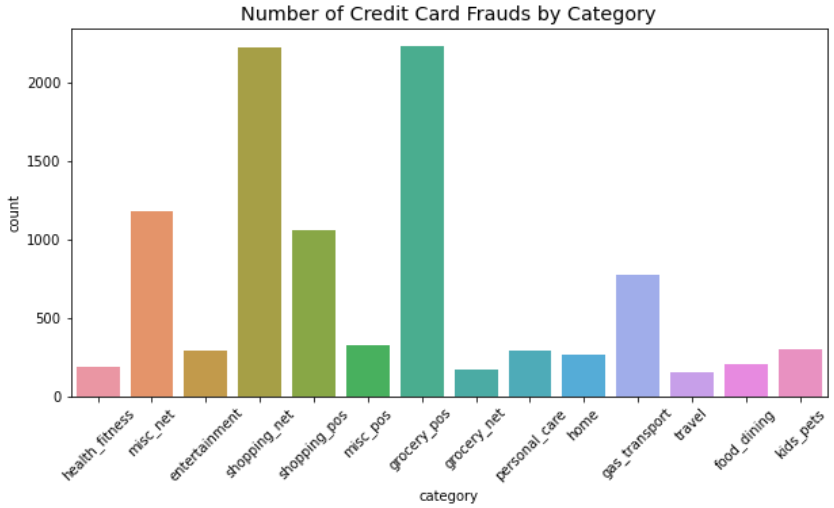
Şekil 3.2.2.1. Örnek korelasyon tablosu



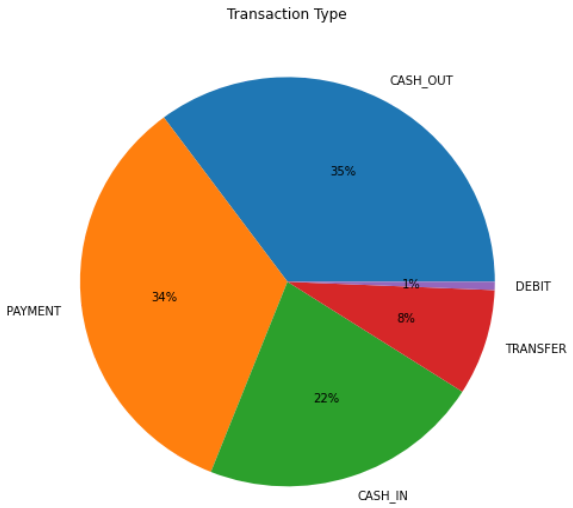
Şekil 3.2.2.2. Veri setindeki dengesiz dağılımın gösterilmesi



Şekil 3.2.2.3. Veri seti özelliklerinin önem dağılımı grafiksel gösterimi



Şekil 3.2.2.4. Kredi kartı dolandırıcılık eyleminin kategorik dağılımının gösterimi.



Şekil 3.2.2.5. Dolandırıcılık eyleminin gerçekleşme tipinin pie chart gösterimi

3.2.3. Veri Ön İşleme ve Özellik Seçimi

Makine öğrenmesi projelerinde en yüksek performansı sağlayabilmek adına üzerinde çalışılan veri setinin algoritmaya en düzgün şekilde verilmesi elzemdir. Bu şartı sağlayabilmek adına ise veri setlerinin belirli ön-işleme adımlarına tabi tutulması gerekmektedir. Bu proje özelinde veri setlerinin tabi tutulduğu ön işleme adımları şu şekildedir.

1. Veri kümesinden, yinelenenleri ve alakasız sütunlar kaldırılmış yahut işlenmiştir. Bu, sayede verilerin tutarlı ve güvenilir olması sağlanmıştır.
2. Var olan özelliklerden çeşitli dönüşümler aracılığıyla yeni özelliklerin elde edilmiştir.
3. Kategorik değişkenler encode edilerek nümerik değerlere dönüştürülmüştür.
4. Veri setindeki gözlemler gruplanarak veya özetlenerek yeni özellik şeklinde düzenlenmiştir.
5. Çeşitli teknikler aracılığıyla (PCA, Chi-Squared, Korelasyona dayalı) özellik seçimi yapılmıştır.
6. Aykırı değerler tespit edilmiştir.
    a. Bu aşamada yapılan testler sonucunda aykırı değerlerin hileli işlemler hakkında bilgi taşıdığı ve yapılan işlemlerin performans kaybına sebep olduğu gözlemlenmiş olup bu değerler son versiyonda baskılanmamış ve modele ham haliyle verilmiştir.

Yapılan ön işleme işlemlerinden sonra veri seti eğitilmek ve test edilmek üzere “train” ve “test” kümelerine ayrılır. Bu kısımda dengesizlik durumu göz önünde bulundurularak veri setini bölmek için “**Stratified Splitting**” tekniği kullanılmıştır. Bu teknik, veri setini bölerken sınıf etiketlerine göre bir örnekleme yapar. Örnekleme işlemi, her bir sınıfın oranını korumak için gerçekleştirilir. Bu şekilde, eğitim ve test setlerinin her birindeki sınıf dağılımı, orijinal veri setine daha yakın olur. Bu yöntem, modelin eğitim ve test aşamalarında her bir sınıfın temsilini koruyarak yanlılığı azaltır ve daha güvenilir sonuçlar elde etmeyi sağlar.

Bununla birlikte, hileli işlem verilerinin normal işlemlere göre daha az olması kaçınılmazdır. Bu da elimizdeki veri setlerini dengesiz kılmaktadır ve makine öğrenimi modelleri, çoğunluk sınıfına aşırı bir şekilde odaklanarak azınlık sınıfını yanlış sınıflandırma eğiliminde olabilir. Bu sorunla mücadele edebilmesi ve modellerin daha fazla dolandırıcılık verisiyle eğitilebilmesi için çeşitli veri seti dengeleme teknikleri kullanılmıştır.

1. **Random Oversampling:** Random oversampling, azınlık sınıfına ait gözlemlerin kopyalanarak veri setinin dengelenmesini sağlar. Bu yöntem, azınlık sınıfındaki gözlemlerin sayısını artırarak her iki sınıf arasındaki dengesizliği azaltır.
    a. Ancak, veri setindeki azınlık sınıfına ait gözlemlerin tekrarlanması, aşırı uyum (overfitting) sorununa yol açabilmektedir.

2. **Random Undersampling:** Random undersampling, çoğunluk sınıfına ait gözlemleri rastgele bir şekilde kaldırarak veri setinin dengelenmesini sağlar. Bu yöntem, çoğunluk sınıfındaki gözlemlerin sayısını azaltarak sınıf dengesizliğini düzeltir.
    a. Ancak, bu yöntemle bilgi kaybı yaşanabilir ve azınlık sınıfından önemli veri örnekleri atılabilir.
3. **SMOTE (Synthetic Minority Over-sampling Technique):** SMOTE, azınlık sınıfına ait gözlemlerin arasındaki ilişkileri kullanarak sentetik örnekler oluşturarak veri setini dengeler. Azınlık sınıfındaki örneklerin kopyalanması yerine, mevcut örnekler arasında interpolasyon yapılarak yeni sentetik örnekler üretilir. Bu yöntem, azınlık sınıfındaki örneklerin çoğaltılmasını ve veri setinin dengelenmesini sağlar.
4. **ADASYN (Adaptive Synthetic Sampling):** ADASYN, SMOTE yöntemine benzer şekilde azınlık sınıfına ait sentetik örnekler üretir, ancak ADASYN, örneklerin ağırlıklarını ayarlayarak adaptif bir şekilde çalışır. Bu yöntem, azınlık sınıfındaki zor örneklerin daha fazla vurgulanmasını sağlar, böylece veri setinin dengesini daha iyi korur.

```
Non-Frauds: 4448085 / 50.0 % of the dataset
Frauds: 4448085 / 50.0 % of the dataset
```

```
Non-Frauds: 5749 / 50.0 % of the dataset
Frauds: 5749 / 50.0 % of the dataset
```

Şekil 3.2.3.1. & Şekil 3.2.3.1. Veri seti oversampling ve undersampling çıktısı

Bu noktada kaçırılmaması gereken püf nokta veri setinin kullanılacak olan teknikten önce bölünmüş olmasıdır. Asıl amaç test setinin veri yapısını bozmamaktır. Veri setini bölmeden oversampling tekniğinin uygulanması, aynı gözlemlerin hem test hem de training setlerinde bulunmasına sebebiyet verebilir. Bu, modelin yalnızca belirli veri noktalarını ezberlemesine neden olabilir, overfitting ve test verilerinde zayıf genellemeye neden olabilir. Veri sızıntısı, tamamen geçersiz olmasa da aşırı iyimser tahmin modelleri oluşturulmasına sebep olabilir. Veri sızıntısı eğitim veri kümesinin dışından olan bilgiler ile model üretmek için kullanıldığı zaman ortaya çıkar.

3.2.4. Makine Öğrenmesi Modellerinin Oluşturulması ve Eğitilmesi

Belirlenen algoritmaların (bkz. 3.2.1.) veri setleri üzerinde uygulamaları yapılmıştır. Lakin model eğitimi ve tahminleme hızı istenilenin oldukça altında kalmıştır. Yapay zeka temelli dolandırıcılık tespit uygulamalarında hız en önemli metriklerden biridir. Bunun sebebi, dolandırıcılık olaylarının genellikle anlık olarak gerçekleşmesidir. Bu yüzden hileli işlemlerin hızlı bir şekilde tespit edilmeleri çok kritik öneme sahiptir. Hızlı bir sistem, dolandırıcılık olaylarını tespit ederek zararları minimize etmeye ve müşteri güvenini sağlamaya yardımcı olur. Bunun nedeni şu maddelerde daha detaylı şekilde açıklanabilir:

1. **Gerçek zamanlı tepki:** Dolandırıcılık olayları hızla gerçekleşebilir ve hızlı bir şekilde müdahale edilmezse büyük zararlara neden olabilir. Hızlı bir yapay zeka sistemine sahip olmak, dolandırıcılık olaylarını hemen tespit edebilir ve gerçek zamanlı olarak tepki verilebilir.
2. **Müşteri deneyimi:** Birçok dolandırıcılık tespit uygulaması, müşterilerin finansal işlemlerini güvende tutmak için kullanılır. Hızlı bir şekilde dolandırıcılık tespit edilemezse, müşteriler finansal kayıplara uğrayabilir ve güvenlerini kaybedebilir. Hızlı bir sistem, müşterilere güvenli bir deneyim sunar ve potansiyel zararları minimize eder.
3. **Büyük veri hacmi:** Dolandırıcılık tespiti için kullanılan yapay zeka sistemleri genellikle büyük miktarda veriyi işlemek zorundadır. Hızlı bir algoritma, bu büyük veri setlerini hızla analiz edebilir ve dolandırıcılık desenlerini tespit etmek için gerekli hesaplamaları yapabilir.
4. **Ölçeklenebilirlik:** Finansal kuruluşlar ve e-ticaret platformları gibi büyük ölçekli sistemler, milyonlarca işlemi aynı anda işlemek zorunda kalabilir. Hızlı bir yapay zeka sistemine sahip olmak, bu büyük ölçekli işlemleri hızla analiz edebilir ve dolandırıcılık olaylarını tespit edebilir.

Yukarıda bahsedilen sebepler dolayısıyla bahsedilen algoritmalara alternatif olarak çeşitli boosting algoritmalarının **(AdaBoost (Adaptive Boosting), Gradient Boosting, XGBoost (Extreme Gradient Boosting), LightGBM, CatBoost)** kullanılmasına karar verilmiş olup performans kaybetmeden zamandan tasarruf edilmiştir.

- **AdaBoost (Adaptive Boosting) Classifier:** Sınıflandırma problemlerinde kullanılan ve zayıf öğrenicileri bir araya getirerek güçlü bir sınıflandırıcı oluşturmayı hedefleyen bir boosting algoritmasıdır.

$$Entropi = -\sum_{k=1}^{n} p_k log_2 p_k$$

Oluşan karar ağacı hata değerlerinin hesaplanması

$$err_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq G_m\left(x_i\right)\right)}{\sum_{i=1}^{N} w_i}$$

- **Gradient Boosting Classifier:** Ardışık olarak eklenen zayıf öğrenicileri bir araya getirerek güçlü bir öğrenici oluşturmayı hedefler. Bu algoritma, özellikle sınıflandırma ve regresyon problemlerinde kullanılır.
- **XGBoost (Extreme Gradient Boosting) Classifier:** Gradient Boosting algoritmasının bir türevidir ve özellikle büyük veri kümeleri üzerinde yüksek performanslı sınıflandırma ve regresyon modelleri üzerinde sıklıkla kullanılır. Düşük bellek kullanımı, hızlı eğitim süreleri ve yüksek tahmin doğruluğu gibi özellikleriyle bilinir.

- **LightGBM Classifier:** Büyük veri kümeleri üzerinde hızlı ve yüksek performanslı sınıflandırma ve regresyon modelleri oluşturmayı amaçlar.
- **CatBoost Classifier:** Diğer gradient boosting algoritmaları gibi ardışık olarak zayıf öğrenicileri bir araya getirerek güçlü bir sınıflandırıcı oluşturur. Özellikle kategorik değişkenlerin işlenmesi konusunda diğer algoritmalardan farklıdır. Kategorik değişkenlerin kodlaması ve özellik mühendisliğiyle uğraşmak yerine, CatBoost, kategorik değişkenleri doğrudan kabul eder ve içsel olarak onları işler.

3.2.5. Makine Öğrenmesi Modellerinin Performans Metriklerince Karşılaştırılması

Bu modülde hem temel veri seti hem de veri dengeleme yöntemleri kullanarak oluşturulmuş veri setleri ile eğitilen modellerin karşılaştırmalı incelenmesi amaçlanmıştır.

3.2.5.1 Değerlendirme Metrikleri

Modellerin performans kıyaslamasını yapmadan önce başarım sonuçları için hangi metriklerin kullanıldığı üzerinde durmak önemlidir. Daha önce de bahsedildiği üzere üzerinde çalışılan veri setleri dengesiz veri setleridir. Bu durumda, kullanımına oldukça aşina olduğumuz doğruluk (accuracy) metriği yalnız başına kullanıldığında yanıltıcı sonuçlar verebilmektedir.

Doğruluk metriği, sınıfların doğru tahmin edilme oranını göstermektedir. Ancak, dengesiz veri setlerinde çoğunluk sınıfı daha fazla örnek içerirken, azınlık sınıfı daha az örneğe sahiptir. Model, çoğunluk sınıfını doğru tahmin ederek yüksek bir doğruluk elde edebilir, lakin bu veri setleri için önemli olan husus azınlık sınıfının doğru tahminlenebilmesidir. Bu durumda, doğruluk metriğine bakıldığında iyi sonuç verdiği görünen bir model aslında istenilen performansı göstermiyor olabilmektedir.

Bu nedenle, dengesiz veri setleri için doğruluk metriği yerine diğer değerlendirme metriklerine odaklanmak önemlidir. Aşağıdaki metrikler, dengesiz veri setleri için daha bilgilendirici sonuçlar sağlar:

- **Hassasiyet (Precision):** Doğru pozitif tahminlerin toplam pozitif tahminlere oranını gösterir. Azınlık sınıfın doğru tahmin edilme oranını ölçer.

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

TP (True Positive): Gerçekte pozitif olan örneklerin doğru bir şekilde pozitif olarak sınıflandırıldığı durumları ifade eder

FP (False Positive): Gerçekte negatif olan örneklerin yanlış bir şekilde pozitif olarak sınıflandırıldığı durumları ifade eder

- **Duyarlılık (Recall):** Doğru pozitif tahminlerin gerçek pozitiflerin toplamına oranını gösterir. Azınlık sınıfın tespit edilme yeteneğini ölçer.

$$Recall = \frac{TP}{TP + FN}$$

TP (True Positive): Gerçekte pozitif olan örneklerin doğru bir şekilde pozitif olarak sınıflandırıldığı durumları ifade eder

FN (False Negative): Gerçekte pozitif olan örneklerin yanlış bir şekilde negatif olarak sınıflandırıldığı durumları ifade eder.

- **F1-Skoru:** Hassasiyet ve duyarlılık arasında denge sağlayan bir metriktir. Hassasiyet ve duyarlılık değerlerinin harmonik ortalaması alınarak hesaplanır. Azınlık sınıfın tespit edilme başarısını ölçer.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **AUC-ROC:** Eğri altındaki alanı ölçen bir metriktir. Sınıflandırma modelinin doğruluk ve yanlış alarm oranlarını değerlendirir.

3.2.5.2 Örnek Değerlendirme Sonuçları

Bir önceki başlıkta bahsedilen modeller için ilk aşamada beş farklı test senaryosu oluşturulmuştur. Bunlar;

1. Yalnızca ön işleme işlemlerinden geçirilmiş temel veri seti ile model eğitimi
2. Random Oversampling yöntemiyle dengelenmiş veri seti ile model eğitimi
3. Random Undersampling yöntemiyle dengelenmiş veri seti ile model eğitimi
4. SMOTE yöntemiyle dengelenmiş veri seti ile model eğitimi
5. ADSYN yöntemiyle dengelenmiş veri seti ile model eğitimi

Aşağıdaki tablolarda (**Tablo 3.2.5.2.1.**, **Tablo 3.2.5.2.2.,** , **Tablo 3.2.5.2.3.,** , **Tablo 3.2.5.2.4.**) farklı veri setlerinden elde edilen sonuç değerleri örneklenmiştir.

model_name	precision	recall	f1_score	AUC
XGBClassifier_baseline	0.932203	0.774648	0.846154	0.887277
CatBoostClassifier_baseline	0.917355	0.781690	0.844106	0.890786
AdaBoostClassifier_baseline	0.806723	0.676056	0.735632	0.837893
GradientBoostingClassifier_baseline	0.822430	0.619718	0.706827	0.809747
LGBMClassifier_baseline	0.117143	0.288732	0.166667	0.642548

Tablo 3.2.5.2.1. Yalnızca ön işleme işlemlerinden geçirilmiş temel veri seti ile model eğitimi sonuçları

	model_name	precision	recall	f1_score	AUC
0	XGBClassifier_baseline	0.838736	0.697064	0.761366	0.848181
1	CatBoostClassifier_baseline	0.818792	0.632124	0.713450	0.815696
2	GradientBoostingClassifier_baseline	0.776934	0.565458	0.654538	0.782304
3	LGBMClassifier_baseline	0.603774	0.342660	0.437197	0.670741
4	AdaBoostClassifier_baseline	0.497312	0.255613	0.337668	0.627130

Tablo 3.2.5.2.2. Yalnızca ön işleme işlemlerinden geçirilmiş temel veri seti ile model eğitimi sonuçları

model_name	precision	recall	f1_score	AUC
XGBClassifier_baseline	0.932203	0.774648	0.846154	0.887277
CatBoostClassifier_baseline	0.917355	0.781690	0.844106	0.890786
XGBClassifier_RandomOverSampler	0.860294	0.823944	0.841727	0.911860
LGBMClassifier_RandomOverSampler	0.863636	0.802817	0.832117	0.901303
XGBClassifier_SMOTE	0.832117	0.802817	0.817204	0.901273
XGBClassifier_ADASYN	0.783784	0.816901	0.800000	0.908262
CatBoostClassifier_RandomOverSampler	0.761290	0.830986	0.794613	0.915275
CatBoostClassifier_SMOTE	0.705882	0.845070	0.769231	0.922241
CatBoostClassifier_ADASYN	0.670455	0.830986	0.742138	0.915152
AdaBoostClassifier_baseline	0.806723	0.676056	0.735632	0.837893
LGBMClassifier_SMOTE	0.662857	0.816901	0.731861	0.908104
LGBMClassifier_ADASYN	0.638889	0.809859	0.714286	0.904547
GradientBoostingClassifier_baseline	0.822430	0.619718	0.706827	0.809747
GradientBoostingClassifier_SMOTE	0.207612	0.845070	0.333333	0.919840
GradientBoostingClassifier_ADASYN	0.189062	0.852113	0.309463	0.923003
GradientBoostingClassifier_RandomOverSampler	0.160526	0.859155	0.270510	0.925823
AdaBoostClassifier_SMOTE	0.112558	0.852113	0.198850	0.920443
AdaBoostClassifier_ADASYN	0.098919	0.838028	0.176952	0.912636
LGBMClassifier_baseline	0.117143	0.288732	0.166667	0.642548
AdaBoostClassifier_RandomOverSampler	0.080422	0.859155	0.147077	0.921369
CatBoostClassifier_RandomUnderSampler	0.057481	0.922535	0.108220	0.948629
LGBMClassifier_RandomUnderSampler	0.045150	0.901408	0.085993	0.934776
XGBClassifier_RandomUnderSampler	0.039876	0.908451	0.076399	0.935950
GradientBoostingClassifier_RandomUnderSampler	0.039888	0.901408	0.076395	0.932576
AdaBoostClassifier_RandomUnderSampler	0.027432	0.901408	0.053245	0.924003

Tablo 3.2.5.2.3. Veri dengeleme teknikleri ve base model karşılaştırması

model_name	precision	recall	f1_score	AUC
XGBClassifier_baseline	0.838736	0.697064	0.761366	0.848181
CatBoostClassifier_baseline	0.818792	0.632124	0.713450	0.815696
GradientBoostingClassifier_baseline	0.776934	0.565458	0.654538	0.782304
CatBoostClassifier_RandomOverSampler	0.444150	0.869430	0.587947	0.931866
CatBoostClassifier_ADASYN	0.377146	0.819689	0.516600	0.906300
CatBoostClassifier_SMOTE	0.309423	0.831434	0.451002	0.910858
LGBMClassifier_baseline	0.603774	0.342660	0.437197	0.670741
XGBClassifier_RandomOverSampler	0.233923	0.936097	0.374309	0.960021
AdaBoostClassifier_baseline	0.497312	0.255613	0.337668	0.627130
XGBClassifier_SMOTE	0.176471	0.842487	0.291816	0.910949
LGBMClassifier_RandomOverSampler	0.158982	0.964767	0.272981	0.969020
XGBClassifier_ADASYN	0.157757	0.851123	0.266177	0.913663
LGBMClassifier_SMOTE	0.135835	0.867703	0.234898	0.919398
GradientBoostingClassifier_SMOTE	0.128355	0.863903	0.223503	0.916591
XGBClassifier_RandomUnderSampler	0.120325	0.975820	0.214234	0.969231
LGBMClassifier_RandomUnderSampler	0.119296	0.976166	0.212609	0.969213
GradientBoostingClassifier_ADASYN	0.119928	0.871848	0.210852	0.919172
LGBMClassifier_ADASYN	0.117346	0.879793	0.207073	0.922569
CatBoostClassifier_RandomUnderSampler	0.114009	0.976857	0.204188	0.968551
GradientBoostingClassifier_RandomOverSampler	0.094508	0.952332	0.171952	0.952275
GradientBoostingClassifier_RandomUnderSampler	0.086608	0.954750	0.158810	0.951010
AdaBoostClassifier_SMOTE	0.087735	0.796891	0.158068	0.876750
AdaBoostClassifier_ADASYN	0.066439	0.805527	0.122753	0.873126
AdaBoostClassifier_RandomOverSampler	0.057055	0.899827	0.107305	0.910974
AdaBoostClassifier_RandomUnderSampler	0.050403	0.894646	0.095429	0.903190

Tablo 3.2.5.2.4. Veri dengeleme teknikleri ve base model karşılaştırması

3.3. Derin Öğrenme Modülü

Bu bölümde gerçekleştirilen çalışma bünyesinde kullanılan derin öğrenme algoritmaları tanımlanacak ve çalışmada kullanılan modellerinin performans metrikleri ve çıktıları yorumlanacaktır.

3.3.1. Kullanılacak Olan Derin Öğrenme Modellerinin Tespiti

Literatür taramasında detaylıca araştırılmış olan makaleler tablo haline getirilmiş olup (Tablo 3.3.) karşılaştırmak amacıyla en fazla kullanılan algoritmalar renklendirilmiştir. Akademik çalışmalarda en fazla kullanılan algoritmalar **Artificial Neural Network ,**

Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory (LSTM) Networks ve Autoencoders olarak belirlenmiştir ve bu algoritmaların kullanılmasına karar verilmiştir.

Makale / Algoritmalar	Naive Bayes	Convolutional Neural Networks (CNNs)	Recurrent Neural Networks (RNNs)	Long Short-Term Memory (LSTM) Networks	Autoencoders	Deep Neural Network (DNN)
Detection of Credit Card Fraud in E-Commerce Using Data Mining [18]	x	x				
Credit Card Fraud Detection Using Convolutional Neural Networks [19]		x				
Recurrent Neural Networks with Attention for Credit Card Fraud Detection [20]			x	x		
A Deep Learning Approach to Credit Card Fraud Detection [21]					x	x
Deep Learning for Credit Card Fraud Detection [25]						x
A Hybrid Deep Learning Framework for Fraud Detection in Credit Card Transactions [26]				x	x	
A Comparative Study on Credit Card Fraud Detection Using Deep Learning Techniques [27]		x	x		x	x
Autoencoder-Based Fraud Detection in Credit Card Transactions [28]					x	
A Deep Learning Framework for Fraud Detection in Financial Statements [29]						x
Deep Learning-Based Fraud Detection in Bitcoin Transactions [30]				x	x	x
Deep Learning for Credit Card Fraud Detection: A Comparative Analysis of Classifiers [31]			x	x	x	x
Anomaly Detection in Credit Card Transactions Using Autoencoders and Local Outlier Factor [32]					x	
Fraud Detection in Online Transactions Using Autoencoders and Logistic Regression [33]					x	
Credit Card Fraud Detection with Deep Convolutional Neural Networks [34]		x				

Tablo 3.3.1.1. Literatürde en fazla değinilen derin öğrenme algoritmaları

3.3.2. Veri Ön İşleme İşlemlerinin Gerçekleştirilmesi

Bir önceki modülde ön işleme işlemlerinden geçmiş ve test-train setlerine ayrılmış veri setleri ile derin öğrenme algoritmaları ile eğitilmeden önce "**Standard Scaler**" ile ölçeklendirilir. StandardScaler, veri setindeki özelliklerin ortalamasını 0, varyansını ise 1 yaparak veri setini standartlaştıran bir ölçeklendirme yöntemidir. Özellikle makine öğrenmesi modellerinde sıklıkla kullanılan bir veri ön işleme tekniğidir. Veri setini ölçeklendirmedeki temel amaç farklı özelliklerin farklı ölçeklere sahip olması durumunda, modelin yanlışlıkla ölçek büyük olan özelliklere ağırlık vermesini

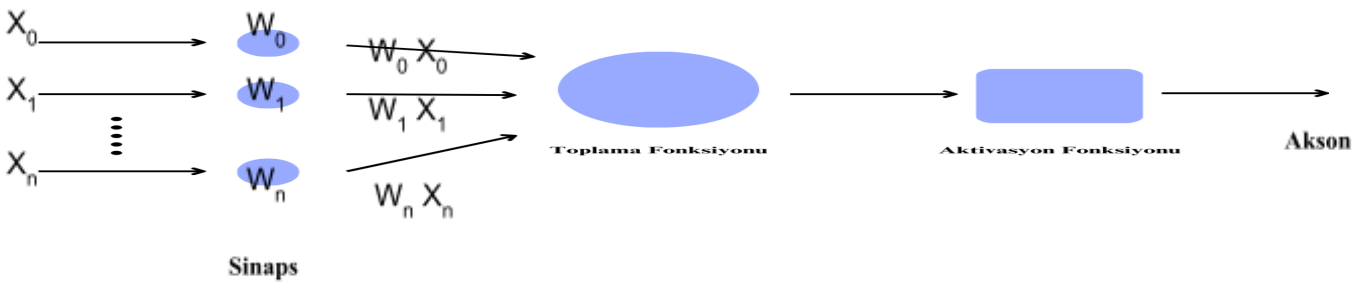
engellemektir. Özelliklerin benzer ölçekte olması, modelin daha dengeli ve tutarlı bir şekilde çalışmasını sağlar. Standard Scaler'ı kullanarak veri setini standartlaştırmak, modelin daha hızlı ve stabil bir şekilde eğitilmesini sağlar. Ölçeklendirilen veri seti algoritmanın ihtiyacına göre (bkz. CNN, RNN, LSTM) yeniden boyutlandırma (reshape) işleminden geçer ve istenilen şekle getirilir.

3.3.3. Derin Öğrenme Modellerinin Eğitilmesi

3.3.3.1. Artificial Neural Network (ANN)

İnsan beyninde bulunan; bilgi transferi gerçekleştirmek, mesaj oluşturmak ve diğer sinir hücreleriyle iletişim kurmak gibi görevleri olan nöronlardan esinlenerek geliştirilen bir teknolojidir. Yapay sinir ağları; kendisine verilen bilgilerden yola çıkar, bilgiyi işleyerek değerlendirmeler yapar ve bu değerlendirmeler ile daha önce kendisine verilmemiş bilgiler hakkında tahminde bulunur.

Yapay sinir ağlarının geliştirilmesi, insanda bulunan sinir hücrelerinin matematiksel modelinin oluşturulması temeline dayanır.



Şekil 3.3.3.1.1. Sinir hücrelerinin matematiksel modeli

Burada kullanılan aktivasyon fonksiyonu sigmoid, hiperbolik tanjant gibi fonksiyonlar kullanılabilir.

3.3.3.2. Convolutional Neural Networks (CNNs)

Matris olarak alınabilecek girdilerin işlenmesi ve analiz edilmesinde kullanılan bir sinir ağı türüdür. Verilen girdilerden istenen kalıpları öğrenme ve çıkarabilme özelliğinden dolayı matris verilerine kolaylıkla çevrilebilecek resim ve video gibi veriler üzerinde özellikle tercih edilir.

CNN algoritması için verilerin convolutional, pooling ve fully connected katmanlarından geçmesi gerekmektedir. Bu katmanlar ve görevleri şu şekilde tanımlanabilir.

Convolutional Layer (Evrişimli katmanlar): matris olarak alınan veri üzerinde belirlenen filtrenin sol üst köşeden başlayarak sağ alt köşeye kadar taraması ve bu taramanın sonucu olarak ise uzamsal özelliklerin çıkarılma işlemidir.

Pooling Layer (Havuzlama katmanı): uzamsal özellikleri azaltarak elde edilen özelliklerden en önemlilerini belirlemek için kullanılan katmandır. Bu katman sayesinde hesaplama karmaşıklığı azaltılarak diğer katmanlar için kolaylık sağlanır.

Fully Connected Layes (Bağlı katmanlar): convolutional ve pooling katmanlarından geçen verilerin tek boyutlu diziye dönüştürülerek sinir ağları ile öğrenmesi sağlanır. Burada her katmandaki her nöron, bir sonraki katmandaki her nörona bağlanır.

3.3.3.3. Recurrent Neural Networks (RNNs)

Zaman serisi ve dizi verileri işlemek için özelleştirilmiş sinir ağlarıdır. Zaman serisi özelinde geçmişten gelen verileri kullanarak gelecekte gelmesi öngörülen verilerin tahmini için kullanılır. Diğer derin öğrenme algoritmalarında girdiler birbirlerinden bağımsız le alınırken, RNN'lerde ise girdiler birbirleri ile ilişkilidir. Yani veri girdileri arasında birbirini takip eden bir ilişki vardır ve bu ilişki sayesinde RNN eğitim süresince, kendi içinde kurduğu döngü benzeri geri besleme mekanizması ile diğer ilişkilerini hatırlar. Bu işlem içerisinde kullandığı aktivasyon fonksiyonu ise şu şekilde ifade edilebilir;

$$a^{\langle t \rangle} = g\left(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\right)$$

W: ağırlık

$a^{\langle t \rangle}$: t anındaki gizli katman

W_{aa}: bir önceki gizli katmanın ağırlığı

W_{ax}: şu anda bulunulan gizli katmanın ağırlığı

g: aktivasyon fonksiyonu

$$y^{\langle t \rangle} = g\left(W_{ya}a^{\langle t \rangle} + b_y\right)$$

$y^{\langle t \rangle}$: çıktı

W_{ya} : çıktı katmanının ağırlık değeri

3.3.3.4. Long Short-Term Memory (LSTM) Networks

RNN algoritmasının özelleşmiş hali olan LSTM algoritması, RNN'lerde meydana gelebilecek bağlam boşluklarının tahmin edilerek giderilmesi amacıyla ortaya çıkmıştır. Bu yapıda önceki zaman aralıklarına ait veriler seçilerek tutulur, geçmiş verilerin unutulmasına izin verilir ve gating mekanizmaları kullanılır. Gating mekanizmasında bulunan giriş, unutma ve çıkış kapıları sayesinde gerekli veriler uzun süreler boyunca yakalanıp korunabilir.

LSTM'ler de diğer sinir ağları gibi ağ içindeki bilgi akışını düzenlemek için aktivasyon fonksiyonlarından yararlanır.

3.3.3.5. Autoencoders

Autoencoderlar çok boyutlu verileri önce hidden space içerisinde sıkıştırıp sonrasında sıkıştırılan veriyi yeniden inşa etmek ve sıkıştırılan veriler içerisinden önemli özellikleri çıkarmak amacıyla geliştirilen bir çeşit sinir ağı mimarisidir.

Autoencoder'lar kodlayıcı (encoder) ve kod çözücü (decoder) olmak üzere iki temel bileşene sahiptir.

Kodlayıcı: giriş verilerini alarak daha düşük boyutlu hidden space denilen bir katmanda verilerin yeni temsillerini oluşturur.

Kod çözücü: hidden space içerisinde bulunan sıkıştırılmış verileri başlangıçtaki boyutlarına getirerek yeniden oluşturur.

Autoencoderların amacı, yapılan yeniden oluşturma işleminde meydana gelebilecek kayıp ve hataları en aza indirmektir.

3.3.4. Derin Öğrenme Modellerinin Performans Metriklerince Karşılaştırılması

3.3.4.1 Örnek Değerlendirme Sonuçları

Bu modülde derin öğrenme modelleri temel veri seti üzerinde eğitildiği için tek bir test senaryosu bulunmaktadır. Aşağıdaki tablolarda (XX, XY) farklı veri setlerinden elde edilen sonuç değerleri örneklenmiştir.

model_name	precision	recall	f1_score	AUC
ANN	0.934579	0.704225	0.803213	0.950273
CNN	0.930435	0.753521	0.832685	0.950215
RNN	0.869231	0.795775	0.830882	0.949980
LSTM	0.929825	0.746479	0.828125	0.939603
Autoencoder	0.104141	0.584507	0.176784	0.949217

model_name	precision	recall	f1_score	AUC
ANN	0.939693	0.695617	0.799440	0.947254
RNN	0.976109	0.580357	0.727921	0.953532
LSTM	0.981851	0.658685	0.788438	0.974791
Autoencoder	0.006663	0.004464	0.005346	0.708898

Şekil 3.3.4.1.1 & Şekil 3.3.4.2. Örnek Derin Öğrenme Sonuç Çıktısı

3.4. Karşılaştırmalı Analiz Modülü

Bu bölümde gerçekleştirilen çalışma bünyesinde kullanılan makine öğrenmesi ve derin öğrenme modellerinin performans çıktıları yorumlanacaktır.

3.4.1. Makine Öğrenmesi ve Derin Öğrenme Modellerinin Karşılaştırılması

Bu kısımda yukarıda ele alınan ve ayrı ayrı çıktı elde edilen makine öğrenmesi ve derin öğrenme modellerinin bir arada değerlendirilmesi söz konusu olmaktadır. Farklı veri setlerine ait örnek bileşik çıktılar aşağıdaki tablolarda verilmiştir.

	model_name	precision	recall	f1_score	AUC
0	XGBClassifier_baseline	0.932203	0.774648	0.846154	0.887277
1	CatBoostClassifier_baseline	0.917355	0.781690	0.844106	0.890786
2	XGBClassifier_RandomOverSampler	0.860294	0.823944	0.841727	0.911860
3	CNN	0.930435	0.753521	0.832685	0.950215
4	LGBMClassifier_RandomOverSampler	0.863636	0.802817	0.832117	0.901303
5	RNN	0.869231	0.795775	0.830882	0.949980
6	LSTM	0.929825	0.746479	0.828125	0.939603
7	XGBClassifier_SMOTE	0.832117	0.802817	0.817204	0.901273
8	ANN	0.934579	0.704225	0.803213	0.950273
9	XGBClassifier_ADASYN	0.783784	0.816901	0.800000	0.908262
10	CatBoostClassifier_RandomOverSampler	0.761290	0.830986	0.794613	0.915275
11	CatBoostClassifier_SMOTE	0.705882	0.845070	0.769231	0.922241
12	CatBoostClassifier_ADASYN	0.670455	0.830986	0.742138	0.915152
13	AdaBoostClassifier_baseline	0.806723	0.676056	0.735632	0.837893
14	LGBMClassifier_SMOTE	0.662857	0.816901	0.731861	0.908104
15	LGBMClassifier_ADASYN	0.638889	0.809859	0.714286	0.904547
16	GradientBoostingClassifier_baseline	0.822430	0.619718	0.706827	0.809747
17	GradientBoostingClassifier_SMOTE	0.207612	0.845070	0.333333	0.919840
18	GradientBoostingClassifier_ADASYN	0.189062	0.852113	0.309463	0.923003
19	GradientBoostingClassifier_RandomOverSampler	0.160526	0.859155	0.270510	0.925823
20	AdaBoostClassifier_SMOTE	0.112558	0.852113	0.198850	0.920443
21	AdaBoostClassifier_ADASYN	0.098919	0.838028	0.176952	0.912636
22	Autoencoder	0.104141	0.584507	0.176784	0.949217
23	LGBMClassifier_baseline	0.117143	0.288732	0.166667	0.642548
24	AdaBoostClassifier_RandomOverSampler	0.080422	0.859155	0.147077	0.921369
25	CatBoostClassifier_RandomUnderSampler	0.057481	0.922535	0.108220	0.948629
26	LGBMClassifier_RandomUnderSampler	0.045150	0.901408	0.085993	0.934776
27	XGBClassifier_RandomUnderSampler	0.039876	0.908451	0.076399	0.935950
28	GradientBoostingClassifier_RandomUnderSampler	0.039888	0.901408	0.076395	0.932576
29	AdaBoostClassifier_RandomUnderSampler	0.027432	0.901408	0.053245	0.924003

Şekil 3.4.1.1. Örnek Derin Öğrenme ve Makine Öğrenmesi Sonuç Çıktısı

	model_name	precision	recall	f1_score	AUC
0	CatBoostClassifier_baseline	0.949558	0.870942	0.908552	0.935441
1	XGBClassifier_baseline	0.970423	0.838880	0.899869	0.919423
2	ANN	0.939693	0.695617	0.799440	0.947254
3	LSTM	0.981851	0.658685	0.788438	0.974791
4	RNN	0.976109	0.580357	0.727921	0.953532
5	LGBMClassifier_baseline	0.657143	0.662744	0.659931	0.831148
6	CatBoostClassifier_RandomOverSampler	0.468237	0.984172	0.634568	0.991364
7	XGBClassifier_RandomOverSampler	0.451108	0.982955	0.618409	0.990704
8	AdaBoostClassifier_baseline	0.854419	0.400162	0.545053	0.700037
9	XGBClassifier_SMOTE	0.343697	0.987013	0.509853	0.992288
10	CatBoostClassifier_SMOTE	0.307683	0.991477	0.469627	0.994297
11	GradientBoostingClassifier_baseline	0.665348	0.341315	0.451180	0.670547
12	XGBClassifier_ADASYN	0.267920	0.989042	0.421626	0.992775
13	LGBMClassifier_RandomOverSampler	0.260509	0.993506	0.412781	0.994931
14	CatBoostClassifier_ADASYN	0.207996	0.990260	0.343783	0.992693
15	LGBMClassifier_SMOTE	0.203781	0.993101	0.338170	0.994043
16	LGBMClassifier_ADASYN	0.164737	0.993101	0.282596	0.993296
17	LGBMClassifier_RandomUnderSampler	0.148148	0.996753	0.257956	0.994673
18	XGBClassifier_RandomUnderSampler	0.123300	0.997159	0.219463	0.993997
19	CatBoostClassifier_RandomUnderSampler	0.114055	0.997565	0.204705	0.993775
20	GradientBoostingClassifier_RandomOverSampler	0.079222	0.943588	0.146171	0.964706
21	GradientBoostingClassifier_SMOTE	0.068493	0.909091	0.127389	0.946555
22	GradientBoostingClassifier_RandomUnderSampler	0.065880	0.994318	0.123572	0.988048
23	AdaBoostClassifier_RandomUnderSampler	0.032476	0.987419	0.062884	0.974698
24	AdaBoostClassifier_RandomOverSampler	0.019423	0.926542	0.038047	0.933040
25	AdaBoostClassifier_SMOTE	0.016685	0.875812	0.032746	0.904548
26	GradientBoostingClassifier_ADASYN	0.016012	0.958198	0.031498	0.941044
27	AdaBoostClassifier_ADASYN	0.005003	0.943182	0.009953	0.850361
28	Autoencoder	0.006663	0.004464	0.005346	0.708898

Şekil 3.4.1.2. Örnek Derin Öğrenme ve Makine Öğrenmesi Sonuç Çıktısı

	model_name	precision	recall	f1_score	AUC
0	XGBClassifier_baseline	0.838736	0.697064	0.761366	0.848181
1	LSTM	0.802947	0.658722	0.723719	0.828938
2	CatBoostClassifier_baseline	0.818792	0.632124	0.713450	0.815696
3	RNN	0.812812	0.617962	0.702119	0.808608
4	ANN	0.837019	0.601382	0.699899	0.800384
5	GradientBoostingClassifier_baseline	0.776934	0.565458	0.654538	0.782304
6	CNN	0.729549	0.563731	0.636009	0.781318
7	CatBoostClassifier_RandomOverSampler	0.444150	0.869430	0.587947	0.931866
8	CatBoostClassifier_ADASYN	0.377146	0.819689	0.516600	0.906300
9	CatBoostClassifier_SMOTE	0.309423	0.831434	0.451002	0.910858
10	LGBMClassifier_baseline	0.603774	0.342660	0.437197	0.670741
11	XGBClassifier_RandomOverSampler	0.233923	0.936097	0.374309	0.960021
12	AdaBoostClassifier_baseline	0.497312	0.255613	0.337668	0.627130
13	XGBClassifier_SMOTE	0.176471	0.842487	0.291816	0.910949
14	LGBMClassifier_RandomOverSampler	0.158982	0.964767	0.272981	0.969020
15	XGBClassifier_ADASYN	0.157757	0.851123	0.266177	0.913663
16	LGBMClassifier_SMOTE	0.135835	0.867703	0.234898	0.919398
17	GradientBoostingClassifier_SMOTE	0.128355	0.863903	0.223503	0.916591
18	XGBClassifier_RandomUnderSampler	0.120325	0.975820	0.214234	0.969231
19	LGBMClassifier_RandomUnderSampler	0.119296	0.976166	0.212609	0.969213
20	GradientBoostingClassifier_ADASYN	0.119928	0.871848	0.210852	0.919172
21	LGBMClassifier_ADASYN	0.117346	0.879793	0.207073	0.922569
22	CatBoostClassifier_RandomUnderSampler	0.114009	0.976857	0.204188	0.968551
23	GradientBoostingClassifier_RandomOverSampler	0.094508	0.952332	0.171952	0.952275
24	GradientBoostingClassifier_RandomUnderSampler	0.086608	0.954750	0.158810	0.951010
25	AdaBoostClassifier_SMOTE	0.087735	0.796891	0.158068	0.876750
26	AdaBoostClassifier_ADASYN	0.066439	0.805527	0.122753	0.873126
27	AdaBoostClassifier_RandomOverSampler	0.057055	0.899827	0.107305	0.910974
28	AdaBoostClassifier_RandomUnderSampler	0.050403	0.894646	0.095429	0.903190

Şekil 3.4.1.3. Örnek Derin Öğrenme ve Makine Öğrenmesi Sonuç Çıktısı

3.4.2. En Başarılı Modellerin Hiperparametrelerinin Optimize Edilmesi

Bu kısım, en iyi performans göstermiş modellerin performansını arttırmak amacıyla hiperparametrelerinin optimize edilmesi üzerine oluşturulmuştur.

```
LGBMClassifier() baseline Best parameters: {'max_depth': 2, 'n_estimators': 5, 'random_state': 0}
LGBMClassifier() baseline Best score: 0.7418117763579826
```

Şekil 3.4.2.1. Örnek Optimizasyon Sonucu

4. SONUÇ

Her projede olduğu gibi yapay zeka temelli dolandırıcılık tespiti çalışmasının da kendine has belirli noktaları ve veri dengesizliği, boyut indirgemesi, performans metrikleri gibi kaçırılmaması gereken önemli detayları bulunmaktadır. Bahsedilen detayların göz önünde bulundurulması ve model eğitimi aşamasından önce bu doğrultuda hareket edilmesi esas alınmıştır.

Bu projede, makine öğrenmesi ve derin öğrenme modellerinin karşılaştırmalı analizinin yapılması sonucunda fraud detection (dolandırıcılık tespiti) alanında kullanılabilecek en iyi modellerin tespiti üzerine çalışılmıştır. Proje içerisinde, farklı veri ön işleme yöntemleri kullanılarak dengesiz veri seti üzerinde çeşitli makine öğrenmesi ve derin öğrenme modelleri eğitilmiş olup eğitilen bu modeller belirlenen başarı metriklerine göre performans değerlendirmesinden geçirilmiştir.

Beş farklı veri seti üzerinde eğitilen ve değerlendirilen, beş makine öğrenmesi beş derin öğrenme olmak üzere toplam on farklı yapay zeka modelinin her bir veri seti üzerinde farklı düzeyde performans gösterdiği gözlemlenmiştir. Bu gözlem sonrasında, dolandırıcılık analizi bazında her projeye uygun mükemmel olarak sınıflandırılabilecek bir algoritmanın olmadığı ve en iyi sınıflandırma algoritmasının bir problem bağlamına ve veri setinin özelliklerine bağlı olarak değişebileceği gözlemlenmiştir. Her algoritmanın avantajları, dezavantajları ve kullanım alanları vardır. Dolayısıyla, en iyi sınıflandırma algoritması tamamen projenin ihtiyaçlarına ve veri setinin özelliklerine bağlıdır.

5. KAYNAKÇA

[1] F. Kaya, "Türkiye'de Kredi Kartı Uygulaması," 2009.

[2] V. Van Vlasselaer *et al.*, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decision Support Systems*, vol. 75, pp. 38–48, Jul. 2015, doi: 10.1016/j.dss.2015.04.013.

[3] D. Olszewski, "Fraud detection using self-organizing map visualizing the user profiles," *Knowledge-Based Systems*, vol. 70, pp. 324–334, Nov. 2014, doi: 10.1016/j.knosys.2014.07.008.

[4] T. Amarasinghe, A. Aponso, and N. Krishnarajah, "Critical Analysis of Machine Learning Based Approaches for Fraud Detection in Financial Transactions," *Proceedings of the 2018 International Conference on Machine Learning Technologies - ICMLT '18*, 2018, doi: 10.1145/3231884.3231894.

[5] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, Nov. 2013, doi: 10.1016/j.eswa.2013.05.021.

[6] Y. Abakarim, M. Lahby, and A. Attioui, "An Efficient Real Time Model For Credit Card Fraud Detection Based On Deep Learning," vol. 7, 2018.

[7] K. Pandey, P. Sachan, and N. G. Ganpatrao, "A Review of Credit Card Fraud Detection Techniques," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, doi: 10.1109/iccmc51019.2021.9418024.

[8] M. Kavitha and M. Suriakala, "Real Time Credit Card Fraud Detection on Huge Imbalanced Data using Meta-Classifiers," 2017.

[9] Thennakoon, Anuruddha, et al. "Real-time credit card fraud detection using machine learning", 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.

[10] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, Feb. 2011, doi: 10.1016/j.dss.2010.08.008.

[11] M. F. Keskenler, D. Dal, and T. Aydın, "Yapay Zeka Destekli ÇOKS Yöntemi ile Kredi Kartı Sahtekarlığının Tespiti," *El-Cezeri Fen ve Mühendislik Dergisi*, May 2021, doi: 10.31202/ecjse.908260.

[12] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," *Procedia Computer Science*, vol. 165, pp. 631–641, 2019, doi: 10.1016/j.procs.2020.01.057.

[13] P. Jiang, J. Zhang, and J. Zou, "Credit Card Fraud Detection Using Autoencoder Neural Network," 2019.

[14] S. Dhawan, S. Charan, R. Gangireddy, S. Kumar, and T. Chakraborty, “Spotting Collective Behaviour of Online Frauds in Customer Reviews,” 2019.

[15] A. YILMAZ and M. SELİMOGLU, “Kredi Kartı Dolandırıcılık Tespitinin Makine Öğrenmesi Yöntemleri ile Tahmin Edilmesi,” *Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, Feb. 2021, doi: 10.20854/bujse.873804.

[16] P. Tiwari, S. Mehta, N. Sakhuja, J. Kumar, and A. Singh, “Credit Card Fraud Detection Using Machine Learning: A Study Technical Report,” 2021.

[17] U. Porwal and S. Mukund, “Credit Card Fraud Detection in e-Commerce: An Outlier Detection Approach,” May 2019.

[18] Y Kırelli, S Arslankaya, and M Zeren, “Detection of Credit Card Fraud in E-Commerce Using Data Mining.,” *Avrupa Bilim ve Teknoloji Dergisi*, pp. 522–529, Oct. 2020.

[19] Shu, L., Zhang, S., Lian, C., & Gao, Y. (2019). “Credit Card Fraud Detection Using Convolutional Neural Networks”. In 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 173-178).

[20] Zhang, Y., Liu, X., Chen, M., & Li, J. (2020). *“Recurrent Neural Networks with Attention for Credit Card Fraud Detection”*. IEEE Access, 8, 80810-80818.

[21] Li, Y., Li, X., Li, W., & Liu, C. (2017). *“A Deep Learning Approach to Credit Card Fraud Detection”*. IEEE Access, 5, 8806-8819.

[22] Han, X., Zhang, Y., & Wang, J. (2018). *“Deep Learning for Credit Card Fraud Detection”*. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C) (pp. 153-160). IEEE.

[23] Al-Kadi, M., Raza, M., & Ghani, A. (2020). *“A Hybrid Deep Learning Framework for Fraud Detection in Credit Card Transactions”*. IEEE Access, 8, 214512-214523.

[24] Farooq, M., Khan, I. A., & Han, X. (2021). *“A Comparative Study on Credit Card Fraud Detection Using Deep Learning Techniques”*. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (AICA) (pp. 1-6). IEEE.

[25] Carcillo, F., Lejeune, A., & Antonioletti, M. (2019). *“Autoencoder-Based Fraud Detection in Credit Card Transactions”*. IEEE Transactions on Neural Networks and Learning Systems, 31(11), 4984-4994.

[26] Anand, S., Talari, S., & Hasan, M. A. (2018). *“A Deep Learning Framework for Fraud Detection in Financial Statements”*. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 8-13). IEEE.

[27] Zhang, L., Liu, H., & Wang, J. (2020). *“Deep Learning-Based Fraud Detection in Bitcoin Transactions”*. IEEE Transactions on Computational Social Systems, 7(5),

1125-1135.

[28] Oliveira, J. L., de Carvalho Junior, A. M., & Duarte, A. M. (2020). *"Deep Learning for Credit Card Fraud Detection: A Comparative Analysis of Classifiers"*. In Proceedings of the 2020 IEEE 19th International Conference on Machine Learning and Applications (ICMLA) (pp. 228-234). IEEE.

[29] Nair, V., Aithal, S., & Rao, N. (2019). "*Anomaly Detection in Credit Card Transactions Using Autoencoders and Local Outlier Factor"*. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0259-0264). IEEE.

[30] Khan, M. A., Bao, Y., Chen, L., & Khan, S. U. (2019). *"Fraud Detection in Online Transactions Using Autoencoders and Logistic Regression"*. In Proceedings of the 2019 IEEE
4th International Conference on Big Data Analytics and Smart Computing (ICBDASC) (pp. 1-5). IEEE.

[31] Gao, C., Zhao, H., & Zhang, Y. (2021). *"Credit Card Fraud Detection with Deep Convolutional Neural Networks"*. Journal of Ambient Intelligence and Humanized Computing, 12, 4055-4067.

[32] Nair, V., Sundaram, A., V, A., & Padmanabhan, A. (2020). *"Anomaly Detection in Financial Transactions Using Deep Learning"*. In 2020 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS) (pp. 199-204). IEEE.

[33] Kim, J., Kim, H., & Kim, K. (2020), *"Fraud Detection in Insurance Claims Using Recurrent Neural Networks"*. Journal of Risk and Financial Management, 13(7), 144.

[34] Cai, Y., Li, X., Liu, Y., & Sun, W. (2021), *"Credit Card Fraud Detection Using Deep Autoencoder Neural Networks"*, Journal of Ambient Intelligence and Humanized Computing, 12(8), 8079-8089.

[35] M. Najafabadi, F. Masoumi, and M. Ghasemzadeh, *"A Deep Learning Approach for Fraud Detection in Healthcare Claims,"* IEEE Access, vol. 8, pp. 149620-149628, 2020.

[36] O. O. Ojewale, A. J. Afolabi, and O. O. Olabiyi, *"Deep Learning for Fraud Detection in Mobile Money Transactions,"* Journal of Information Processing Systems, vol. 16, no. 2, pp. 347-360, 2020.

[37] J. Ryu, H. Kim, and J. Kim, *"Unsupervised Deep Learning for Fraud Detection in Credit Card Transaction Data,"* Symmetry, vol. 11, no. 6, p. 808, 2019.

[38] A. Mirzaei, S. M. Hosseini-Motlagh, and M. Eftekhari-Moghadam, *"Autoencoder-Based Fraud Detection System for Electricity Market,"* IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 1506-1515, 2020.

[39] Kaggle. URL: https://www.kaggle.com (Son Erişim Tarihi: 29.12.2022).

[40] Google Dataset Search. URL: https://datasetsearch.research.google.com/ (Son Erişim Tarihi: 29.12.2022).

[41] UCI Machine Learning Repository. URL: https://archive.ics.uci.edu/ml/index.php (Son Erişim Tarihi: 29.12.2022).

[42] OpenML. URL: https://www.openml.org/ (Son Erişim Tarihi: 29.12.2022).

[43] DataHub. URL: https://datahub.io/ (Son Erişim Tarihi: 29.12.2022).

[44] Papers With Code. URL: https://paperswithcode.com/ (Son Erişim Tarihi: 29.12.2022).

[45] The Publications Office of the European Union. URL: https://data.europa.eu/en (Son Erişim Tarihi: 29.12.2022).

[46]Awesome Public Datasets URL: https://github.com/awesomedata/awesome-public-datasets (Son Erişim Tarihi: 29.12.2022).

[47]Credit Card Fraud Detection URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (Son Erişim Tarihi: 29.12.2022).

[48] A. Dal Pozzolo, O. Caelen, Reid A. J. and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification.", In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015.

[49] A. Dal Pozzolo, A. Boracchi, O. Caelen, L. Borgne, Yann-Ael, S. Waterschoot, G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective", Expert systems with applications,41,10,4915-4928, 2014.

[50] A. Dal Pozzolo, A. Boracchi, Giacomo, O. Caelen, C. Alippi, G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy", IEEE transactions on neural networks and learning systems,29,8,3784-3797,IEEE, 2018.

[51] A. Dal Pozzolo, "Adaptive Machine learning for credit card fraud detection" ULB MLG PhD thesis (supervised by G. Bontempi), 2015.

[52] F. Carcillo, A. Dal Pozzolo, Y. Le Borgne, O. Caelen, Y. Mazzer, G. Bontempi, "Scarff: a scalable framework for streaming credit card fraud detection with Spark," Information fusion,41, 182-194, Elsevier, 2018.

[53] F. Carcillo, Y. Le Borgne, O. Caelen, G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization," International Journal of Data Science and Analytics, 5,4,285-300, Springer International Publishing, 2018.

[54] B. Lebichot, Y. Le Borgne, Liyun He, F. Oblé, G. Bontempi "Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection," INNSBDDL 2019: Recent Advances in Big Data and Deep Learning, pp 78-88, 2019.

[55] Fabrizio Carcillo, Y. Le Borgne, O. Caelen, F. Oblé, G. Bontempi, "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection Information Sciences," 2019.

[56] Y. Le Borgne, G. Bontempi, “Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook,” 2021.

[57] B. Lebichot, G. Paldino, W. Siblini, L. He, F. Oblé, G. Bontempi, “Incremental learning strategies for credit cards fraud detection,” IInternational Journal of Data Science and Analytics, 2021.

[58] M. Al-Shabi, “Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets.” Journal of Advances in Mathematics and Computer Science, 1-16, 10.9734/jamcs/2019/v33i530192, 2019.

[59] T. Sarkar, “XBNet: An extremely boosted neural network,” *Intelligent Systems with Applications*, vol. 15, p. 200097, Sep. 2022, doi: 10.1016/j.iswa.2022.200097.

[60] G. Pang, C. Shen, and A. van den Hengel, “Deep Anomaly Detection with Deviation Networks,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019, doi: 10.1145/3292500.3330871.

[61] Y. Xu, H. Dong, M. Zhou, J. Xing, X. Li, and J. Yu, “Improved Isolation Forest Algorithm for Anomaly Test Data Detection,” *Journal of Computer and Communications*, vol. 09, no. 08, pp. 48–60, 2021, doi: 10.4236/jcc.2021.98004.

[62] D. Nugent, “Privacy-Preserving Credit Card Fraud Detection using Homomorphic Encryption,” 2022.

[63] M. Al-Shabi, “Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets Some of the authors of this publication are also working on these related projects: Difference Equation View project Optical MINs View project,” 2019.

[64]Credit Card Transactions. URL: https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions?select=User0_credit_card_transactions.csv (Son Erişim Tarihi: 29.12.2022).

[65] I. Padhi *et al.*, “Tabular Transformers For Modeling Multivariate Time Series,” 2021.

[66]Sparkov Data Generation. URL: https://github.com/namebrandon/Sparkov_Data_Generation (Son Erişim Tarihi: 29.12.2022).

[67] E. Altman, “Synthesizing Credit Card Transactions,” 2019.

[68] P. Grover *et al.*, “FDB: Fraud Dataset Benchmark,” Aug. 2022.

[69]Amazon Fraud Dataset Benchmark. URL: https://github.com/amazon-science/fraud-dataset-benchmark (Son Erişim Tarihi: 29.12.2022).

[50]IEEE-CIS Fraud Detection. URL: https://www.kaggle.com/competitions/ieee-fraud-detection/data (Son Erişim Tarihi: 29.12.2022).

[51] Synthetic data from a financial payment system. URL: https://www.kaggle.com/datasets/ealaxi/banksim1/code (Son Erişim Tarihi: 29.12.2022).

[52] E. Alonso Lopez-Rojas, S. Axelsson, and E. Alonso, “BankSim: A Bank Payment Simulation for Fraud Detection Research CyberAIMs View project BigData@BTH -Scalable resource-efficient systems for big data analytics View project BANKSIM: A BANK PAYMENTS SIMULATOR FOR FRAUD DETECTION RESEARCH,” 2014.

[53] Lopez-Rojas, Edgar Alonso ; Axelsson, Stefan Banksim: A bank payments simulator for fraud detection research Inproceedings 26th European Modeling and Simulation Symposium, EMSS 2014, Bordeaux, France, pp. 144–152, Dime University of Genoa, 2014, ISBN: 9788897999324.

[54]Synthetic Financial Datasets For Fraud Detection. URL: https://www.kaggle.com/datasets/ealaxi/paysim1 (Son Erişim Tarihi: 29.12.2022).

[55] B. Stojanović *et al.*, “Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications,” *Sensors*, vol. 21, no. 5, p. 1594, Feb. 2021, doi: 10.3390/s21051594.

[56] E. A. Lopez-Rojas , A. Elmir, and S. Axelsson, “PaySim: A financial mobile money simulator for fraud detection,” In: The 28th European Modeling and Simulation Symposium-EMSS, Cyprus, 2016.

[57] İmnternet: Random Forest Model https://www.ibm.com/topics/random-forest Erişim Tarihi: 01.06.2023

[58] İnternet: support vector machine model gösterimi https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 Erişim tarihi: 01.06.2023