T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇEKİŞMELİ ÜRETİCİ AĞLAR İLE SENTETİK VERİ ÜRETİMİNİN KREDİ KARTI SAHTEKARLIĞI TESPİTİNE ETKİSİ

Ensar BAYHAN

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Doç. Dr. M. Elif KARSLIGİL

Ağustos, 2021

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇEKİŞMELİ ÜRETİCİ AĞLAR İLE SENTETİK VERİ ÜRETİMİNİN KREDİ KARTI SAHTEKARLIĞI TESPİTİNE ETKİSİ

Ensar BAYHAN tarafından hazırlanan tez çalışması 04.08.2021 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. M. Elif KARSLIGİL
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. M. Elif KARSLIGİL, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Feza BUZLUCA, Üye
İstanbul Teknik Üniversitesi _____

Doç. Dr. M. Amaç GÜVENSAN, Üye
Yıldız Teknik Üniversitesi _____

Danışmanım Doç. Dr. M. Elif KARSLIGİL sorumluluğunda tarafımca hazırlanan Çekişmeli Üretici Ağlar ile Sentetik Veri Üretiminin Kredi Kartı Sahtekarlığı Tespitine Etkisi başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Ensar BAYHAN

İmza

Aileme
ve arkadaşlarıma

TEŞEKKÜR

Tez çalışmam boyunca her konuda bana destek olan değerli hocam Doç. Dr. Mine Elif KARSLIGİL'e, varlıklarıyla hayatımın her aşamasında yanımda olan anneme, babama ve arkadaşlarıma teşekkür ederim.

Ensar BAYHAN

İÇİNDEKİLER

SİMGE LİSTESİ

σ	Aktivasyon Fonksiyonu
ψ	Çözümleyici Fonksiyon
ϕ	Kodlayıcı Fonksiyon
γ	Ortak Dağılım Fonksiyonu
$\mathbb{P}$	Olasılık Dağılımı
ρ	Pearson Korelasyonu

KISALTMA LİSTESİ

ADASYN	Adaptive Synthetic
ANN	Artificial Neural Network
ANOVA	Analysis Of Variance
BKM	Bankalararası Kart Merkezi
CGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CTGAN	Conditional Tabular Generative Adversarial Network
DNN	Deep Neural Networks
DPGAN	Differentially Private Generative Adversarial Network
DT	Decision Tree
ERT	Extremely Randomized Trees
GAN	Generative Adversarial Networks
GMM	Gaussian Mixture Model
KNN	K-Nearest Neighboor
KVKK	Kişisel Verilerin Korunması Kanunu
LDA	Linear Discriminant Analysis
LR	Logistic Regression
LSGAN	Least Squares Generative Adversarial Network
MAGAN	Margin Adaption Generative Adversarial Network
NB	Naive Bayes
PCA	Principal Component Analysis
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit

RF	Random Forest
RFE	Recursive Feature Elimination
RWGAN	Relaxed Wasserstein Generative Adversarial Network
SAE	Sparse Autoencoder
SGD	Stochastic Gradient Descent
SHAP	Shapley Additive Explanations
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TL	Türk Lirası
VGM	Variational Gaussian Mixture Model
WGAN	Wasserstein Generative Adversarial Network

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Çekişmeli Üretici Ağlar ile Sentetik Veri Üretiminin Kredi Kartı Sahtekarlığı Tespitine Etkisi

Ensar BAYHAN

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. M. Elif KARSLIGİL

Gelişen teknoloji ile birlikte insanların ödeme alışkanlıkları değişmiş, kredi kartları daha güvenilir, hızlı ve pratik olması açısından nakit paranın yerini almaya başlamıştır. Temassız ve çevrimiçi ödemelerin de yaygınlaşması ile birlikte kredi kartları günümüzde, temel ödeme aracı olarak anılmaya başlanmıştır. Kredi kartı kullanımının artması, işlemlerin güvenli bir şekilde yapılmasını zorlaştırmış, kredi kartı sahtekarlarının yeni dolandırıcılık yöntemleri geliştirmesine olanak sağlamıştır. Bankalar sahtekarlık işlemlerinden her sene milyarlarca dolar zarar etmektedir. Oluşan maddi zararın yanında, müşteri ve itibar kaybı gibi manevi zararlara da sebep olmaktadır. Kredi kartı sahtekarlarının ele geçirdikleri kart bilgileriyle işlem yapmasını engelleyecek hızlı ve güvenilir sistemlere ihtiyaç duyulmaktadır.

Bu çalışmada kredi kartı sahtekarlığı problemi incelenmiş, sahtekarlık işlemlerinin tespitinde başarı oranını attırmak için yeni bir yöntem önerilmiştir. Yapılan çalışmada normal ve sahtekarlık harcamalarına ait gerçek veri kümesi kullanılmıştır. İlk olarak hazırlanan veri kümesinin özellikleri belirlenmiştir. Ardından kart sahiplerinin önceki işlemleri kullanılarak yeni özellikler türetilmiş, mevcut özellikler ile birlikte yeni türetilen özelliklere, özellik seçimi uygulanmıştır. Yapılan bu özellik türetimi ve seçimi, kart kullanıcılarının ödeme alışkanlıklarının model tarafından daha iyi öğrenilmesini sağlamıştır. Mevcut özellikler kullanılarak rastgele orman sınıflandırıcısı ile %86.21 başarı elde edilirken, özellik türetimi ve seçimi işlemleri sonucunda %88.24 başarı elde edilmiştir.

Kredi kartı sahtekarlığı için kullanılan veri kümelerinin en önemli ortak problemi sahte

ve yasal işlemlerin aşırı dengesiz olmasıdır. Bu çalışmada, bu soruna çözüm olması amacıyla GAN (Generative Adversarial Networks) ve SMOTE (Synthetic Minority Oversampling Technique) yöntemleri kullanılarak sentetik sahte işlemler üretilmiştir. Öncelikle, özellik türetme ve seçme işlemlerinden sonra elde edilen özelliklerin GAN'a uygun hale getirilmesi amacıyla özkodlayıcı (autoencoder) model kullanılarak özellik çıkarılmıştır. Veri kümesinde bulunan işlemler ile GAN modeli eğitilmiş, ardından çeşitli sayı ve oranlarda sentetik sahte ve yasal işlemler üretilmiştir. GAN, CGAN, WGAN ve CTGAN ile yapılan deneyler sonucunda en başarılı GAN modelinin CTGAN olduğu tespit edilmiştir. CTGAN ile 50.000 adet sentetik veri üretildiğinde %88.75 sınıflandırma başarısı elde edilmiştir.

Kredi kartı sahtekarlığı tespiti yapmaya yönelik pek çok sistemde sahte işlem sayısının yetersizliği en önemli problemdir. Bu nedenle sistem başarısı 500 adet sahte ve 2500 adet yasal işlemden oluşan veri kümesi ile de değerlendirilmiştir. Bu veri kümesi için %80.59 olan sınıflandırma başarısı, 200 adet sentetik sahte işlem üretilerek mevcut örneklere eklendiğinde %82.33'e yükselmiştir. Bu sonuç kredi kartı sahtekarlığı için sentetik veri üretme işleminin sistem başarısını arttırdığını göstermektedir.

Anahtar Kelimeler: Çekişmeli üretici ağlar, kredi kartı sahtekarlığı tespiti, özkodlayıcılar, derin öğrenme, özellik seçimi

ABSTRACT

The Effect of Synthetic Data Generation with Generative Adversarial Networks on Credit Card Fraud Detection

Ensar BAYHAN

Department of Computer Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. M. Elif KARSLIGİL

With the developing technology, people's payment habits have changed, and credit cards have started to replace cash in terms of being more reliable, fast and practical. With the spread of contactless and online payments, credit cards have begun to be referred to as the primary payment method today. The increase in the use of credit cards has made it difficult to conduct transactions securely, allowing credit card fraudsters to develop new fraud methods. Banks lose billions of dollars each year from fraudulent transactions. In addition to the material damage, it also causes moral damages such as loss of customer and reputation. There is a need for fast and reliable systems that will prevent credit card fraudsters from transacting with the card information they have seized.

In this study, the credit card fraud problem has been examined and a new method has been proposed to increase the success rate in detecting fraudulent transactions. In the study, the real dataset of legitimate and fraudulent transactions was used. First, the features of the prepared data set were determined. Then, new features were derived using the previous transactions of the cardholders, and feature selection was applied to the newly derived features along with the existing features. This feature creation and selection enabled the model to better learn the payment habits of card users. While using the existing features, 86.21% success was achieved with the random forest classifier, and 88.24% success was achieved as a result of feature generation and selection processes.

The most common problem with credit card fraud datasets is that fraud and legal transactions are extremely unbalanced. In this study, synthetic transactions were produced by using GAN (Generative Adversarial Networks) and SMOTE (Synthetic Minority Oversampling Technique) methods in order to solve this problem. First of all, features were extracted using the autoencoder model in order to make the features suitable for GAN which obtained after feature creation and selection processes. The GAN model is trained by using real transactions and then synthetic fraud and legitimate transactions in various numbers and rates are generated. As a result of the experiments with GAN, CGAN, WGAN and CTGAN, it was determined that the most successful GAN was CTGAN. When 50,000 synthetic data were generated with CTGAN, 88.75% classification success was achieved.

In credit card fraud detection systems, the insufficiency of the number of fraudulent transactions is the most important problem. For this reason, the success of the system was also evaluated with a dataset consisting of 500 fraud and 2500 legal transactions. The classification success for this dataset, which was 80.59%, increased to 82.33% when 200 synthetic fraud transaction were generated and added to the existing samples. This result shows that the process of generating synthetic data for credit card fraud increases the success of the system.

Keywords: Generative adversarial networks, credit card fraud detection, autoencoders, deep learning, feature selection

1
GİRİŞ

Gelişen teknoloji ile birlikte nakit para yerine, kredi kartlarının kullanılması ve e-ticaret uygulamalarının giderek yaygınlaşması, kredi kartlarını günümüzde temel ödeme aracı haline getirmiştir. Alışverişlerde kolaylık sağlaması, sanal cüzdanların yaygınlaşması ile çevrimiçi ödemelerin de neredeyse tamamı kredi kartları ile yapılmaktadır. Kredi kartları ile yapılan ödemelerin sayısının her geçen gün artması, sahtekarlık işlemi yapan kötü niyetli kişilerin yeni yöntemler geliştirerek daha büyük zararlar verebilmesinin önünü açmıştır.

Kredi kartı sahtekarları, elektronik kart kopyalama cihazları, siber saldırılar veya sosyal mühendislik gibi yöntemler ile kart bilgilerini ele geçirmektedir. Ele geçirilen bu bilgiler kullanılarak kart sahibinin bilgisi ve rızası dışında işlem yapılması kredi kartı sahtekarlığı olarak isimlendirilmektedir. Bu şekilde yapılmış yasal olmayan işlemlere ise sahte işlemler denilmektedir.

Sahte işlemlerin tespiti için uygulanan temel yöntem; kredi kartı kullanıcılarının kendilerine ait olmayan işlemleri bankaya bildirmesidir. Ancak bu yöntem ile bir çok sahte işlem tespit edilememekte, edilse dahi üzerinden belirli bir süre geçtiği için sahtekarların yakalanması zor olmaktadır. Bu sebeple bankalar ve ödeme sistemleri kuruluşları tarafından kredi kartı kullanıcılarının güvenle alışveriş yapabilmeleri adına çeşitli güvenlik uygulamaları geliştirmiştir.

Bankalar, daha önceden tanımlanmış çeşitli kurallar doğrultusunda sahte işlemleri tespit eden sistemler geliştirmişlerdir. Ancak sahtekarlar her geçen gün yeni yöntemler keşfederek bu kuralları atlatmanın bir yolunu bulmaktadır. Bu sebeple yeni geliştirilen sistemler makine öğrenmesi ve derin öğrenme gibi yöntemler kullanarak, sahte işlem tespit oranını yükseltmeyi hedeflemektedir.

Bu çalışmada makine öğrenmesi yöntemleri ile kredi kartı sahtekarlığının tespit edilme başarısını arttırmak için farklı yaklaşımlar denenmiş ve elde edilen sonuçlar değerlendirilmiştir. Kredi kartı sahtekarlığı tespitine yönelik yapılan geçmiş

çalışmalara ait literatür taraması yapılmıştır. Çalışma süresince kullanılan özellik türetme, özellik seçimi ve sentetik veri üretme gibi yöntemler üzerine daha önce yapılmış çalışmalar ve bu çalışmalarda kullanılan veri kümeleri incelenmiştir. Kredi kartı sahtekarlık tespitinde kullanılmak üzere önerilen ve gerçeklenen sistemin detayları açıklanmıştır. Çalışma süresince kullanılan gerçek veri kümesinin özelliklerinin belirlenme süreci ve yapılan deneylerin sonuçlarına yer verilmiştir.

1.1 Literatür Taraması

Kredi kartı sahtekarlığı tespiti sistemlerinde makine öğrenmesi, derin öğrenme, özellik türetme, özellik seçimi ve sentetik veri üretimi üzerine yapılmış çalışmalar incelenmiştir. Ayrıca çalışmalarda kullanılan veri kümelerinin özellikleri de incelenmiştir.

1.1.1 Veri Kümeleri

Kredi kartı sahtekarlığı tespiti çalışmalarına kullanılan veri kümelerinin detayları genellikle paylaşılmamaktadır. Tablo 1.1'de incelenen çalışmalar arasından veri kümelerinin ayrıntılarına yer verenler görülebilmektedir. Ayrıca bu çalışmada kullanılan veri kümesinin detaylarına da tabloda yer verilmiştir. Literatürdeki bir çok çalışmanın aksine, bu çalışmada kullanılan veri kümesi oldukça büyük olup, Türkiye'de 2019 yılında gerçekleştirilmiş ve 30 adet banka tarafından, BKM'ye sağlanan gerçek kredi kartı işlemlerinden oluşmaktadır. Veri kümesinde bir senede gerçekleştirilmiş 618 bin sahte işlem bulunmaktadır. Toplam işlem sayısı ise 8 milyarın üzerindedir.

Tablo 1.1 Literatür çalışmalarında kullanılan veri kümeleri

Veri Kümesi	Tipi	Yasal	Sahte	Toplam	Özellik
Almanya [1]	Kredi Başvurusu	700	300	1000	20
Avustralya [2]	Kredi Kartı	383	307	690	13
Avrupa [3]	Kredi Kartı	284B	492	284B	30
Kang Fu [4]	Kredi Kartı	260M	4B	260M	-
Güney Kore [5]	Kredi Kartı	11M	5B	11M	83
BKM (Bu çalışma)	Kredi Kartı	8,315M	618B	8,315M	60

1.1.2 Sahtekarlık Tespitinde Kullanılan Yöntemler

Kredi kartı sahtekarlığı alanında yapılmış çalışmalara ve kullanılan yöntemlere yer veren incelemelerde (survey) makine öğrenmesi yöntemlerinin sıklıkla kullanıldığı görülmüştür. [6–8]. Ayrıca topluluk (ensemble) öğrenmesi yöntemi kullanılarak

yapılan bir çalışmada [9], torbalama (bagging) yönteminin klasik makine öğrenmesi yöntemlerine göre daha başarılı olduğuna değinilmiştir.

Kredi kartı sahtekarlığı tespiti çalışmalarında kullanılan veri kümeleri, hassas bilgiler içerebildiği için genellikle paylaşılmamaktadır. Bu sebeple sahtekarlık tespiti çalışmalarında ilk karşılaşılan problem veri kümesinde kullanılacak özelliklerin belirlenmesidir. İşleme ait özelliklerin yetersiz kalması sebebiyle gruplama yöntemi kullanılarak yeni özellikler türetilmiştir. [10–12]

Gruplama yöntemi ile özellik türetme yapılan bir çalışmada (APATE) [13] üç farklı zaman aralığı, üç farklı hesaplama türü ve beş farklı gruplama karakteristiği seçilerek toplam 60 yeni özellik eklenmiştir. Eklenen yeni özelliklerin sahte işlemleri tespit etme başarısını %97.83'ten, %98.77'ye çıkardığına değinilmiştir. Bu çalışmayı referans alan başka bir çalışmada ise (HOBA) [14] gruplama türlerini arttırmanın başarıya etkisini gözlemlemek amacıyla; dokuz farklı zaman aralığı, on üç farklı gruplama karakteristiği, dört farklı hesaplama tipi ve dört farklı işlem alışkanlığı ölçütü kullanılarak toplamda 1410 yeni özellik eklenmiştir. APATE çalışmasında türetilen 60 özellik ile HOBA çalışmasındaki 1410 özellik karşılaştırılmış ve fazla sayıda özellik kullanmanın daha başarılı olduğu gözlemlenmiştir. Derin öğrenme yöntemlerinden biri olan RNN yöntemi kullanılarak, 60 özellik ile yapılan sınıflandırmada başarı %35.82 iken, 1410 özellik ile %58.28 başarıya ulaşılmıştır. Ayrıca aynı 1410 özellik ile yapılan SVM ile sınıflandırma sonucunda %40.10 başarı elde edilmiştir. Bu sebeple derin öğrenme yöntemlerinin makine öğrenmesi yöntemlerine göre daha başarılı olduğuna değinilmiştir.

Özellik seçiminin makine öğrenmesine etkisini inceleyen bir çalışmada [15] sarmalama (wrapper) ve filtreleme (filter) yöntemleri kullanılarak özellik seçimi yapılmıştır. Mevcutta bulunan 21 adet özellik sarmalama yöntemi ile 5, filtreleme yöntemi ile 17 adet özelliğe indirgenmiştir. Topluluk (Ensemble) öğrenimi ve makine öğrenmesi yöntemlerinden oluşan 5 farklı model ile deneyler yapılmış ve sarmalama yöntemi ile seçilen beş adet özellik ile başarının, %70.5'ten, %74.6'ya çıktığı tespit edilmiştir.

Günümüzde derin öğrenme yöntemlerinin bir çok alanda kullanılması ile birlikte, sahtekarlık tespiti için de tercih edilen bir yöntem haline gelmiştir. Derin öğrenme ve makine öğrenmesi yöntemlerini karşılaştıran bir çalışmada [5], derin öğrenme ile %90.7 başarı elde edilirken, makine öğrenmesi yöntemleri ile %87.8 başarı elde edilmiştir. Özkodlayıcılar (Autoencoders) ve RBM (Restricted Boltzmann Machine) kullanılarak yapılan bir çalışmada [16] ise yüksek sayıda veri kümesi kullanıldığında, derin öğrenme sistemlerinin sahte işlemleri tespit etmede daha başarılı olduğu öne

sürülmektedir. Bin adet işlemin bulunduğu veri kümesi ile %54.83 başarı elde edilirken, 280 bin işlemin bulunduğu veri kümesi ile yapılan sınıflandırma sonucunda %96.03 başarı elde edilmiştir.

Görüntü verilerinde sıklıkla kullanılan, evrişimsel sinir ağları (CNN) kullanılarak yapılan bir çalışmada [4], kredi kartı işlemlerine ait veri kümesi görüntü verisine dönüştürülerek modelin başarısı ölçülmüştür. Rastgele orman (RF) ile 0.29, destek vektör makinesi (SVM) ile 0.27 ve yapay sinir ağları (ANN) ile 0.26 f1-skoruna ulaşılırken, CNN yöntemi ile f1-skorun 0.31'e yükseldiği görülmüştür. Ayrıca bu çalışmada gruplama yöntemi ile yeni özellikler türetilmiş ve bu yeni özellikler ile sahte işlemleri tespit etme skorunun 0.22'den 0.26'ya yükseldiği gözlemlenmiştir.

1.1.3 Veri Çoğaltma

Kredi kartı sahtekarlığı tespiti problemlerinde, sahte işlemlerin yasal işlemlere oranla çok az olması sıklıkla karşılaşılan bir sorundur. Bu problemi çözmek amacıyla altkümeleme (undersampling) veya üstkümeleme (oversampling) yöntemleri kullanılmaktadır. Günümüzde çekişmeli üretici ağların (GAN) gelişmesiyle birlikte sentetik veri üreterek de dengesiz veri (imbalanced data) problemi çözülebilmektedir. GAN yöntemi ile üstkümeleme yöntemlerinin karşılaştırıldığı bir çalışmada [17], üstkümeleme yöntemlerinden rastgele üstkümeleme ile %16, SMOTE ile %16 ve ADASYN ile %4 başarı elde edilirken, GAN ile %65 ve WGAN (Wasserstein GAN) ile %72 başarı elde edilmiştir. Bu sebeple GAN ile sentetik veri kullanmanın, üstkümelemeye göre daha başarılı olduğuna değinilmiştir.

Yüksek oranda dengesiz veri kümesiyle bulunan bir çalışmada [18] sparse autoencoder (SAE) yöntemi ile özellik çıkarılmıştır. Mevcut özellikler ile %83.47 başarı elde edilirken, SAE ile özellik çıkarıldığında bu başarının %87.36'ya yükseldiğine yer verilmiştir. Çıkarılan özelliklerin mevcut veri kümesinin karakteristiğini daha iyi öğrendiği için, GAN ile üretilen sentetik verilerin orjinal sahte işlemlere daha çok benzediğine değinilmiştir. Bu çalışmada sentetik veri üretmede kullanılan GAN modelinin ayrıştırıcısı (discriminator) ile sınıflandırma yapılmıştır. Ancak GAN'ın ayrıştırıcısı %82.68 başarı ile örnekleri sınıflandırırken, SVM ile %87.36 başarı elde edildiği için, GAN'ın ayrıştırıcısının yeterince iyi eğitilemediğine değinilmiştir.

K-En Yakın Komşu (KNN) yöntemi kullanarak belirli bir sınıfa benzer örnekler ile üstkümeleme yapılmasını sağlayan SMOTE yöntemi ile GAN yönteminin kıyaslandığı başka bir çalışmada [19] ise sentetik veri üretiminin derin öğrenmeye olan katkısı araştırılmıştır. Alınan sonuçlara göre SVM ile %96.62, CNN ile %91.50 başarı elde edilmiştir. Ayrıca SMOTE ile üretilen sentetik örnekler kullanılarak %96.62 başarı elde

edilirken, GAN ile üretilen örnekler ile %99.96 başarı elde edildiği öne sürülmüştür. PATE-GAN adlı yeni bir metot öneren çalışmada [20], sentetik veri üreterek, gerçek veri kümesini gizlemek hedeflenmiştir. Benzer bir amaçla oluşturulmuş DPGAN [21] ile karşılaştırılmıştır. Test kümesi, yeni önerilen PATE-GAN ile %87.37, DPGAN ile %85.78 başarı ile sınıflandırılmış ve PATE-GAN'ın daha başarılı olduğu tespit edilmiştir.

Sentetik veri üretimi için GAN yöntemi çeşitli şekillerde geliştirilerek yeni yöntemler önerilmiştir. Kredi kartı sahtekarlığı veri kümesi ile yapılan bir çalışmada [22] farklı GAN yöntemleri karşılaştırılmıştır. LSGAN (Least Squares GAN), MAGAN (Margin Adaption GAN), WGAN (Wasserstein GAN), RWGAN (Relaxed Wasserstein GAN) yöntemleri kullanılarak üretilen sentetik veriler yapay sinir ağları modelinin eğitiminde kullanılmıştır. Alınan sonuçlara göre 500 adet yeni sahte işlem üreten GAN %73.98, MAGAN %76.58, LSGAN %77.22, WGAN %78.51 ve RWGAN %80.44 başarı ile örnekleri sınıflandırmıştır.

GAN modelleri ile yapılan çalışmalar genellikle görüntü veri kümeleri ile yapılmaktadır. Kredi kartı sahtekarlığı gibi tablo veri kümeleri için CTGAN (Conditional Tabular GAN) [23] isimli yeni bir model geliştirilmiştir. Bu GAN modelinde kategorik olan özellikler üretilen sentetik veri kümesinde korunabilmektedir. Yapılan çalışmada CTGAN ile sentetik veri üretiminden önce, mevcut veri kümesinin özkodlayıcılar ile kodlandığında üretilen verinin orjinaline daha benzer olduğuna yer verilmiştir. Yapılan çalışmada, özkodlayıcılar kullanılmadığında %16.2 başarı ile sınıflandırılırken, özkodlayıcılar ile kodlanan veri kümesinin %51.9 başarı ile sınıflandırdığı görülmüştür.

1.2 Tezin Amacı

Kredi kartı sahtekarlığı, kart sahiplerinin bilgisi ve rızası olmadan, o kişinin çalıntı kredi kartı bilgileri ile alışveriş yaparak gerçekleştirilen bir dolandırıcılık yöntemidir. Bu tez kapsamında, Türkiye'de gerçekleştirilmiş, 30 adet bankaya ait kredi kartı işlemleri kullanılarak, geliştirilen yöntem ile sahte işlemlerin tespit edilmesi amaçlanmıştır. Buna ek olarak;

- Kredi kartı sahtekarlığı sürecinin açıklanması,
- Kredi kartı sahtekarlığı tespiti kapsamında kullanılan veri kümesinin detaylandırılması,
- Kullanılan veri kümesi için belirlenen özelliklere ek olarak türetilen yeni özelliklerin açıklanması,

- Yeni türetilen özellikler arasından en başarılı özelliklerin seçilme sürecinin açıklanması,
- Veri kümesinde az sayıda sahte işlem bulunan diğer çalışmalara referans olması amacıyla, sentetik sahte işlem üretilme sürecinin performansa etkisinin detaylandırılması,
- Kredi kartı sahtekarlığı tespiti modelinin ayrıntılarına yer verilmesi,
- Geliştirilen model ile gelecekte oluşabilecek olası dolandırıcık faaliyetlerinin önüne geçilmesi

amaçlanmıştır.

Kredi kartı sahtekarlığı her sene kişilere ve kurumlara milyarlarca dolar zarar vermektedir. Ayrıca kurumların itibarını zedelediği için manevi olarak da bir çok zararı bulunmaktadır. Bu problemin büyüklüğüne dikkat çekmek ve doğabilecek yeni zararların önüne geçebilmek hedeflenmiştir. Ayrıca bu alanda yapılan diğer çalışmalara referans olması amaçlanmıştır.

1.3 Hipotez

Kredi kartı veri kümelerinin en büyük problemlerinden biri, makine öğrenmesi yöntemlerinde kullanılacak veri kümesinin özelliklerini doğru bir şekilde belirlemektir. Bu tez çalışmasında sahtekarlık tespitinde kullanılan modeli iyileştirmek amacıyla kartların önceki işlemleri çeşitli zaman aralıklarında gruplandırılarak, yeni özellikler türetilmiştir. Mevcut özellikler ile birlikte yeni özellikler arasından en etkili olanlar seçilmiştir. Ayrıca sentetik veri üretme başarısını arttırması amacıyla, özkodlayıcı (autoencoder) model kullanılarak seçilen özellikler kodlanmıştır.

Kredi kartı sahtekarlığı tespitindeki bir diğer problem ise veri kümesinde bulunan sahte ve yasal işlemlerin dağılımının aşırı dengesiz olmasıdır. Bu problemi aşmak için genellikle altkümeleme yöntemi kullanılmaktadır. Ancak bu durumda da kullanılabilecek veri kümesinin sayıca yetersiz kalması olasıdır. Günümüzde görüntü veri kümeleri üzerinde sıklıkla kullanılan çekişmeli üretici ağlar (GAN) ile gerçeklerine oldukça benzeyen sentetik veri kümeleri üretebilmek mümkündür. Bu çalışmada GAN yöntemi ile kredi kartı işlemleri gibi metin tabanlı bir veri kümesinde sentetik sahte işlemler üretilerek, dengesiz veri probleminin çözülmesini hedefleyen bir sistem tasarlanmıştır. Ayrıca aşağıdaki çıkarım ve hipotezler geliştirilmiştir;

- Kredi kartı sahtekarlığında kullanılan veri kümelerine ait özelliklerin tamamı modelin eğitilmesi için gerekli olmayabilir. Tüm özellikler arasından korelasyonu en yüksek olanlar seçilerek özellikler azaltıldığında, başarının artması beklenmektedir.

- Kredi kartı işlemlerine ait özellikler sahtekarlık karakteristiğinin anlaşılması için yeterli olmamaktadır. Kart sahibinin ödeme alışkanlıklarına ait yeni özellikler türetmenin sistemin başarısını arttıracağı öngörülmektedir.

- Çekişmeli üretici ağlar kullanılarak gerçeğine oldukça yakın sahte işlemler üretebilmek mümkündür. Az sayıda işlemi olan kredi kartı veri kümelerinde, sentetik veri üretmenin sınıflandırma başarısını arttıracağı öngürülmektedir.

2
METODOLOJİ

Bu çalışmada yeni özellik türetme ve özellik seçiminin kredi kartı sahtekarlığı tespiti başarısına etkisi incelenmiştir. Ayrıca sentetik veri örnekleri oluşturularak, veri kümesinde bulunan örnek sayısının başarıya olan etkisi değerlendirilmiştir. Bu sebeple özellik seçimi, özellik türetme, özellik çıkarma ve sentetik veri üretme için farklı yöntemler kullanılarak deneyler yapılmıştır.

Bu bölümde, tasarlanan sistemde uygulanan yöntemler tanıtılmıştır. Kullanılan yöntemler sırasıyla, özellik türetme, özellik seçimi, özellik çıkarma, sentetik veri üretimi başlıkları altında incelenmiştir.

2.1 Özellik Türetme

Veri kümesinde bulunan mevcut özellikler kullanılarak, veri karakteristiğinin daha iyi anlaşılabilmesi için yeni özellikler oluşturularak, mevcut özelliklere eklenmesine özellik türetme denilmektedir. Öğrenme sürecinin kolaylaştırılması, insan uzmanların veriyi daha iyi analiz edebilmesi gibi amaçlar için özellik türetme kullanılabilir. Özellik türetme sonucu oluşan özelliklerin bilgilendirici olması beklenmektedir.

Kredi kartı işlemlerinden oluşan veri kümelerinde sahte işlemlerin tekrar edilmesi durumu sıklıkla görülmektedir. Bu sebeple, kart sahiplerinin geçmiş işlemlerinin analiz edilmesi oldukça önemlidir. Geçmiş işlemlerin makine öğrenmesi yöntemleri tarafından daha iyi anlaşılabilmesi için işlem gruplama (aggregation) ile yeni özellikler türetilmesi önerilmiştir [10]. İşlem gruplamayla özellik türetme adımları;

- İşlemin yapıldığı kredi kartı ile son bir gün, bir hafta veya bir ay gibi, farklı zaman aralıklarında gerçekleştirildiği tüm işlemler bir grupta toplanır.
- İşyeri kategorisi, işlem türü gibi, bir gruplama özelliği belirlenir.
- İşlemlerin ortalama tutarı, toplam tutarı, işlem sayısı gibi bir hesaplama türü belirlenir.

- Grupta bulunan işlemler arasından, gruplama özelliği aynı olan işlemler seçilir.
- Seçilen işlemler hesaplama türüne göre hesaplanır.
- Yeni türetilen özelliğin değeri, hesaplama işleminin sonucudur.

şeklinde sıralanabilmektedir. Örneğin, işlemin gerçekleştirildiği kredi kartıyla, aynı işyerinde, son bir haftada yapılmış işlemlerin ortalama tutarı, yeni bir özellik olarak veri kümesine eklenmektedir. Gruplama yapılarak türetilen özellikler, işlemin karakteristiğinin daha iyi anlaşılmasını sağlayarak, öğrenme sürecinin başarısını arttırmaktadır.

2.2 Özellik Seçimi (Feature Selection)

Makine öğrenmesi ve derin öğrenme modellerinde kullanılan veri kümelerinde bulunan özellikler model başarısını belirlemede büyük bir önem taşımaktadır. Veri kümesinde bulunan olası bir gürültü (noise) değeri, veya özelliklerin arasında korelasyon olması gibi problemler, modelin başarısını olumsuz yönde etkileyebilmektedir. Bu sebeple kullanılacak özelliklerin doğru olarak seçilmesi oldukça önemlidir. Ayrıca özellik sayısının daha az olması öğrenme süresini de azaltmaktadır. Özellik seçiminin faydaları aşağıdaki gibi sıralanabilir;

- Makine öğrenmesi algoritmalarının daha hızlı öğrenmesini sağlar.
- Modelin karmaşıklığını azaltır ve anlaşılması daha kolay bir hale getirir.
- Doğru alt özellik kümesi seçildiği takdirde model başarısını arttırır.
- Aşırı öğrenme (overfitting) riskini azaltır.

Özellik seçimi göz ile yapılabilmektedir. Ancak modelin öğrenmesinde etkili, kritik bir özelliğin elenmesi başarıyı düşürebilmektedir. Bu problemin önüne geçmek amacıyla çeşitli yöntemler geliştirilmiştir. Bu yöntemler genellikle, özelliklerin model başarısında olan etkilerini belirleyerek, özellikleri en etkiliden etkisize doğru sıralayarak özellik seçimi sürecini kolaylaştırmaktadır. Özellik seçimi yöntemleri filtreleme, sarmalama ve gömülü yöntemler olarak üç grupta incelenmektedir.

2.2.1 Sarmalama (Wrapper) Yöntemi

Sarmalama, sınıflandırıcı performansına dayalı olarak, özellikleri yinelemeli olarak seçerek, optimal özellik kümesini bulmaya çalışan bir yöntemdir. Sarmalama ile

özellik seçimi işlemine boş bir özellik kümesi belirlenerek başlanmaktadır. Ardından her adımda özellik kümesinde bulunmayan, yeni bir özellik eklenerek, sınıflandırma başarısı ölçülmektedir. Tüm farklı kombinasyonlar ile özellik kümeleri oluşturulduktan sonra, sınıflandırma başarısı en yüksek olan özellik kümesi seçilmektedir. Model, her özellik kümesi kombinasyonu ile eğitilerek, sınıflandırılma başarısı ölçüldüğü için, özellik seçme maliyeti yüksektir.

Sarmalama yöntemi iki farklı şekilde uygulanabilmektedir. İleri (forward) arama yönteminde en küçük özellik kümesinden başlanarak, sırayla eklenen özellikler ile en başarılı özellik kümesi belirlenmektedir. Geri (backward) arama yönteminde ise özellikler sırayla azaltılarak en başarılı küme bulunmaktadır.

Geri aramalı sarmalama yöntemlerinden en yaygınlarından biri yinelemeli özellik elemesi (recursive feature elimination - RFE) [24]'dir. RFE yöntemi, özellikleri başarıya olan etkisine göre büyükten küçüğe doğru sıralamaktadır. RFE ile özellik seçimi işlemine ait yarıkod Algoritma 1'de görülebilmektedir. Özellikler sınıflandırma başarısına etkisine göre sıralandıktan sonra, en başarılıları seçilerek, özellik seçimi yapılabilmektedir. RFE ile özellikler sıralanırken, farklı sınıflandırıcılar kullanılabilmektedir.

Algoritma 1: Yinelemeli Özellik Elemesi Algoritması

```
Tüm özellikleri kullanarak modeli eğit
Modelin sınıflandırma başarısını ölç
for i = S ... 1 iken, her S_i alt kümesi için do
    En önemli i özelliği S_i kümesinde sakla
    S_i özellik kümesini kullanarak modeli eğit
    Modelin sınıflandırma başarısını ölç
    Her özelliğin sıralamasını yeniden hesapla
end
S_i kümesini kullanarak özellikleri başarısına göre sırala
S_i kümesini seçilmek istenilen özellik sayısına göre azalt
```

2.2.2 Filtreleme (Filter) Yöntemi

Filtreleme yöntemine göre özellikler, çeşitli istatistiksel testler ile puanlanarak seçilmektedir. Özellik seçimi, herhangi bir makine öğrenimi algoritmasından bağımsız olarak yapılabilmektedir. Filtreleme yöntemlerinde her bir özelliğe ait bir puan hesaplanmaktadır. Hesaplanan değer kullanılarak, belirlenen bir kurala göre özellik seçimi uygulanmaktadır. Örneğin, "Puanı X değerine eşit ve büyük olan özellikleri seç." veya "Puanına göre özellikleri sırala ve en üstteki K özelliği seç." gibi kurallar belirlenebilmektedir. Özellik puanını hesaplamak aşağıdaki yöntemler kullanılabilmektedir;

- **Pearson's Korelasyonu:** Kovaryans (cov) ve standart sapma (σ) kullanarak, iki değişken X ve Y arasındaki bağımlılığı ölçmektedir. Eşitlik 2.1'de pearson fonksiyonu (ρ) görülmektedir.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.1}$$

- **Varyans Analizi (ANOVA):** Bir veya daha fazla kategorik özellik grubu ortalamaları ve bunlara bağlı olan işlemleri analiz etmek için kullanılan bir istatistiksel test yöntemidir. Eşitlik 2.2'de gözlem değeri (x_i), tüm gözlemlerin ortalaması ($\bar{x}$), gözlem sayısı (n) olmak üzere, varyans fonksiyonuna yer verilmiştir.

$$S^2 = \frac{\sum(x_i - \bar{x}^2)}{n-1} \tag{2.2}$$

- **Ki-Kare (Chi-Square:)** Kategorik özellik gruplarının, frekans dağılımlarını kullanarak, aralarındaki korelasyonu değerlendiren istatistiksel bir testtir. Eşitlik 2.3'de beklenen değer (E), ve gözlemlenen değer (O) olmak üzere ki-kare testi fonksiyonu bulunmaktadır.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.3}$$

Yukarıdaki istatistiksel yöntemler kullanılarak tüm özellikler punlanmaktadır. Puanlarına göre en önemliden, önemsize göre özellikler sıralanmaktadır. Sıralama işleminden sonra istenilen sayıda özellik seçilebilmektedir.

Bir filtreleme yöntemi olan SHAP [25], herhangi bir makine öğrenimi modelinin çıktısını açıklamaya yönelik, oyun teorisinden klasik Shapley değerlerini [26] ve bunların ilgili uzantılarını kullanan bir yöntemdir. Makine öğrenmesi tahminlerini yorumlamak için birleşik bir çerçeve sunulmaktadır. SHAP, yapılan tahminleme sonucunda, her özelliğe belirli bir önem değeri atamaktadır. Özelliklere ait önem değerleri oyun teorisi yöntemleri kullanılarak iyileştirilmektedir. Sonuç olarak her özelliğin tahminleme üzerindeki etkisini belirten puanlar ile birlikte özellik önem sırası belirlenmektedir. Elde edilen bu yeni önem puanlarına SHAP değeri denilmektedir.

SHAP değeri hesaplanırken Eşitlik 2.4'de gösterilen ϕ_i fonksiyonu kullanılmaktadır. Eşitlikte bulunan F tüm özelliklerden oluşan kümeyi, S ise özellik alt kümelerini temsil etmektedir. $f_{S\cup\{i\}}$, SHAP değeri hesaplanan i özelliğinin de içinde bulunduğu

özellik alt kümesi ile modeli eğiten fonksiyonu, f_s ise i özelliği kullanılmadan modeli eğiten fonksiyonu temsil etmektedir. x_s ise S kümesinde bulunan özellik değerlerini temsil etmektedir. Özelliklerin sınıflandırma başarısına etkisi, diğer özelliklere bağımlı olduğundan dolayı, olası tüm alt kümeler ($S \subseteq F \setminus \{i\}$) için değerler hesaplanmaktadır.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_s(x_s) \right] \tag{2.4}$$

2.2.3 Gömülü (Embedded) Yöntemler

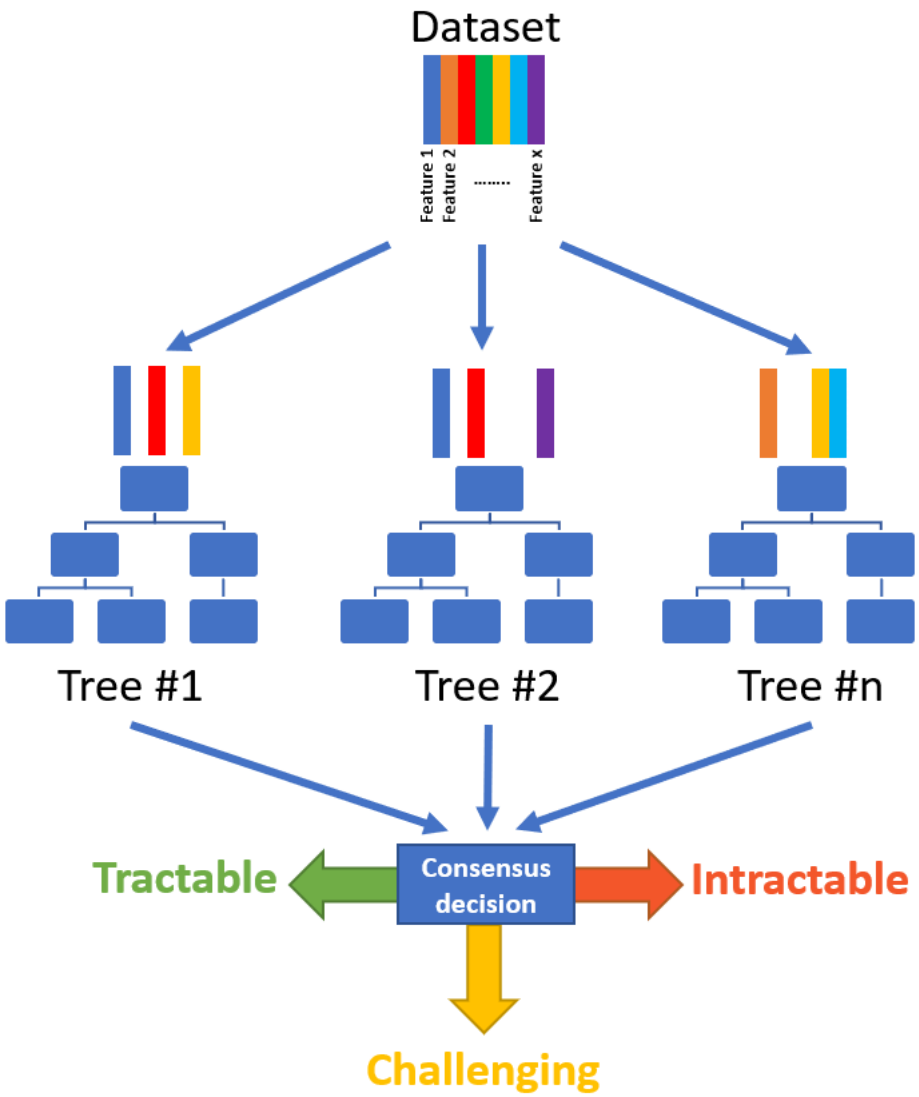
Gömülü özellik seçme yöntemlerinde özellikler, sınıflandırma işlemi sırasında otomatik olarak seçilmektedir. Örneğin, karar ağaçları [27], sınıflandırma yaparken, en önemli özellikleri ağacın kök düğümlerine yerleştirmektedir. Sınıflandırma sonucunda oluşan karar ağacının düğümleri kullanılarak özellikler sıralanabilmektedir. Rastgele orman, son derece rastgele ağaçlar gibi karar ağaçlarından oluşan topluluk öğrenmesi yöntemleri de özellik seçmek için kullanılabilmektedir.

2001 yılında Leo Breiman tarafından geliştirilmiş olan rastgele orman (RF) [28], sınıflandırma işlemi esnasında birden fazla karar ağacı [27] üreterek sınıflandırma başarısını yükseltmeyi hedefleyen bir topluluk öğrenme yöntemidir. Rastgele ormandaki her bir ağaç, bir sınıf tahmini vermektedir ve en çok oyu alan sınıf, modelin tahmini olmaktadır. Rastgele ormanın ardında basit ama güçlü bir kavram olan "kalabalıkların bilgeliği" kavramı bulunmaktadır.

Şekil 2.1'de rastgele orman yöntemine ait bir model bulunmaktadır. Bu modelde görülebileceği üzere ormanda oluşan her ağaç, farklı özellik kümeleri ile oluşabilmektedir. Her karar ağacında yapılan tahminleme sonucu, fikir birliği kararı (consensus decision) ile rastgele ormanın nihai tahmini belirlenmektedir.

Rastgele ormanda bulunan ağaçlarda, bir düğümü parçalara ayırırken en önemli özelliğin aranması yerine, rastgele oluşturulmuş bir alt özellik kümesi arasından en iyi özellik aranmaktadır. Böylece geniş bir çeşitlilikle sonuçlanan, daha iyi bir model elde edilmektedir.

Rastgele ormanın önemli bir kullanım alanı, özelliklerin sınıflandırma üzerindeki önemini belirleyebilmesidir. Belirli bir özelliği kullanan ağaç düğümlerinin, ormandaki tüm ağaçlar arasındaki kirliliği ne kadar azalttığına bakılarak özelliğin skoru oluşturulmaktadır. Her bir özellik için bu skor otomatik olarak eğitilmekte ve



Şekil 2.1 Rastgele orman yöntemi [29]

sonuçları ölçeklendirilirmektedir. Bu yöntem ile elde edilen skorlar kullanılarak, veri kümesine özellik seçimi uygulanabilmektedir.

Son derece rastgele ağaçlar (ERT) [30], bir ormanda bulunan birden fazla, birbiriyle ilişkili karar ağaçlarının sonuçlarını toplayan bir tür topluluk öğrenmesi tekniğidir. ERT, eğitim veri kümesinden çok sayıda budanmamış karar ağacı oluşturarak çalışmaktadır. Regrasyon durumunda karar ağaçların tahmininin ortalaması alınmaktadır. Sınıflandırma durumunda ise çoğunluk oylaması kullanılarak yapılmaktadır.

ERT, karar ağaçlarının oluşturduğu topluluğu inşa ederken daha basit bir algoritma kullanmasına rağmen, genellikle rastgele orman algoritmasıyla aynı veya daha iyi performans elde edebilmektedir. ERT sonucunda oluşturulan karar ağaçlarının kök düğümlerindeki özellikler sınıflandırma için daha etkili iken, yaprak düğümlerinde

bulunan özellikler ise daha az etkili olmaktadır. Bu sebeple ERT'nin düğümlerinin sıralaması kullanılarak özellikler niteliklerine göre sıralanabilmektedir.

2.3 Özellik Çıkarma (Feature Extraction)

Fazla sayıda özellikten oluşan veri kümelerinde aşırı öğrenme (overfitting) sorunu sıklıkla görülmektedir. Ayrıca özellik sayısının fazla olması makine öğrenmesi yöntemlerinin başarısını düşürmekte ve öğrenme süresini uzatmaktadır. Bu tür problemleri önlemek amacıyla özellik çıkarma teknikleri uygulanmaktadır. Özellik çıkarma, veri kümesinde bulunan mevcut özellikler kullanılarak, farklı bir düzlem üzerinde yeni bir özellik kümesi oluşturulmasıdır. Çıkarılan özelliklerin sayısı genellikle mevcut özelliklerin sayısından daha az olmaktadır. Ancak yeni özellik kümesinin sayısının, mevcut özelliklerden fazla olması da mümkündür. Çıkarılan özellikler yeni bir düzlemde oluşturulduğu için, veri kümesinin geriye döndürülerek tekrar eski özelliklerin elde edilmesi mümkün değildir.

Özellik seçimi, sıklıkla özellik çıkarma ile karıştırılmaktadır. Özellik çıkarma sürecinde veri kümesi farklı bir uzaya taşınır ve mevcut özelliklerden farklı yeni bir özellik kümesi oluşur. Özellik seçiminde ise mevcut özelliklerden, öğrenme sürecine katkısı olmayanlar ayrıştırılır. Çıkarma işleminden sonra kalan özellikler, çıkarma işleminden önce bulunan mevcut özellikler ile aynı olup, yalnızca sayısı azaltılmıştır.

2.3.1 Temel Bileşen Analizi (Principal Component Analysis)

Temel bileşen analizi (PCA), 1901'de Karl Pearson tarafından, mekanikteki ana eksen teoreminin bir benzeri olarak geliştirilmiştir [31]. Pearson'dan bağımsız olarak 1930 yılında Harold Hotelling tarafından yeniden geliştirilmiş ve isimlendirilmiştir [32]. PCA, veri kümesinin, en yüksek varyansa göre yeni bir koordinat sistemine dönüştürülmesini sağlayan bir özellik çıkarma yöntemidir. PCA'in temel amacı çok boyutlu veri kümelerinin en önemli özelliklerini tespit ederek daha düşük boyutlu bir uzaya taşımaktır. PCA, özellik çıkarma işlemi için etiket bilgisine ihtiyaç duymamaktadır. Aşağıda PCA ile özellik çıkarma işlem adımları sıralanmıştır.

1. **Standardizasyon:** Tüm değerlerin belirli bir aralıkta olması amacıyla veri kümesine standardizasyon uygulanır. Etiket bilgisi hariç, veri kümesinde bulunan tüm özelliklerin ortalaması hesaplanır. Eşitlik 2.5'de, x başlangıç değeri, μ ortalama ve σ standart sapma fonksiyonu olmak üzere, z

standardizasyon fonksiyonu görülmektedir.

$$z = \frac{x - \mu}{\sigma} \tag{2.5}$$

2. **Kovaryans Matrisi:** Veri kümesinde bulunan ilişkili özellikleri tespit etmek amacıyla kovaryans matrisi hesaplanır. Kovaryans matrisinde bulunan her bir eleman, iki özellik arasındaki ilişkiyi temsil eder. Eşitlik 2.6'de x değeri x_i, y değeri y_i, x değerlerinin ortalaması $\bar{x}$, y değerlerinin ortalaması $\bar{y}$ ve veri sayısı N olmak üzere kovaryans fonksiyonu ($cov_{x,y}$) görülmektedir.

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N - 1} \tag{2.6}$$

3. **Özdeğerler ve Özvektörler:** Veri kümesinin temel bileşenlerini belirlemek amacıyla, kovaryans matrisinin özdeğer ve özvektörü hesaplanır. Kovaryans matrisi A ile temsil edilmek üzere Eşitlik 2.7 ve Eşitlik 2.8'de bulunan denklemleri sağlayan özdeğer (λ) ve özvektör ($\vec{v}$) hesaplanmaktadır. Eşitlikte bulunan I değeri birim matrisi temsil etmektedir.

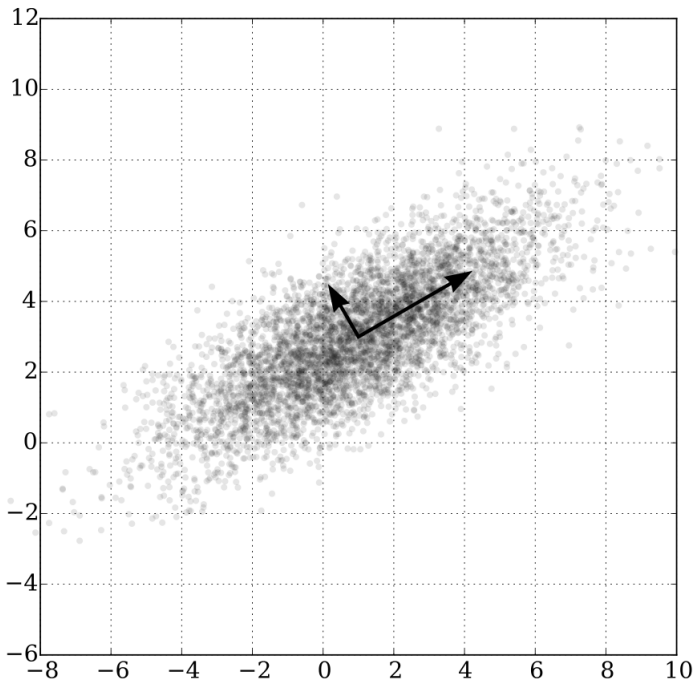
$$A\vec{v} = \lambda\vec{v} \tag{2.7}$$

$$\begin{aligned} A\vec{v} - \lambda\vec{v} &= 0 \\ \vec{v}(A - \lambda I) &= 0 \\ |A - \lambda I| &= 0 \end{aligned} \tag{2.8}$$

4. **En yüksek özdeğerlere sahip özvektörlerin belirlenmesi:** En yüksek özdeğere sahip özellikler, diğerlerine kıyasla veri kümesine ait daha fazla ayrıntı içermektedir. Bu sebeple özdeğeri en yüksek olandan en düşük olana doğru özellikler sıralanır. PCA işlemi sonucunda çıkarılmak istenilen özellik sayısı kadar yeni özellik, sıralanmış olan özvektörlerden seçilerek elde edilmektedir.

Şekil 2.2'de PCA ile özellik çıkarmaya ait bir örnek bulunmaktadır. X ve Y eksenleri ile temsil edilen iki adet özelliğe sahip örnekler, siyah okla gösterilen iki eksen boyunca

yeni bir uzaya çıkarılabilmektedir. Bu iki eksen PCA ile çıkarılan yeni özellikleri temsil etmektedir.



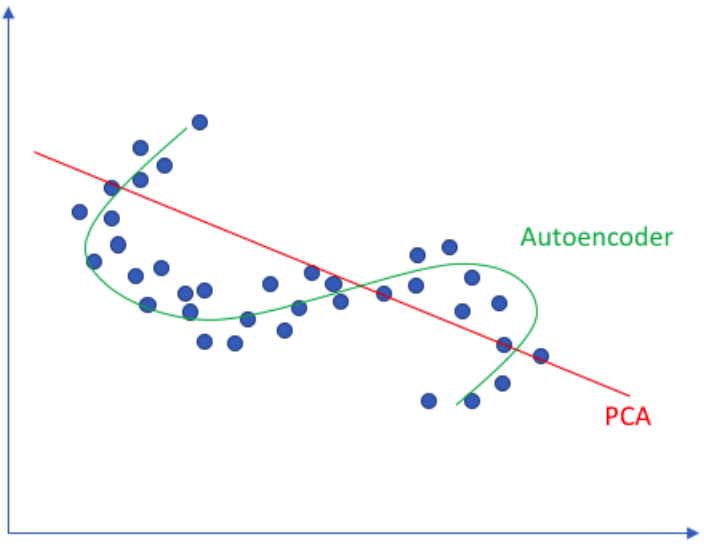
Şekil 2.2 PCA ile özellik çıkarma örneği

2.3.2 Özkodlayıcı (Autoencoder)

Denetimsiz bir öğrenme türü olan özkodlayıcı [33], girdi katmanındaki değerleri çıktı katmanına kopyalayan bir yapay sinir ağı modelidir. Özkodlayıcı, veri kümesini eğitirken kendi etiketlerini kendi ürettiği için öz-denetimli bir öğrenme modeli olarak da kabul edilmektedir. Özkodlayıcılar, girdilerin boyutunu daha küçük bir gösterime indirmek için de kullanılmaktadır.

Yalnızca doğrusal veri kümelerini dönüştürebilirken PCA'e karşın özkodlayıcılar, doğrusal olmayan veri kümelerini de dönüştürelebilmektedir. Şekil 2.3 de linear olmayan bir veri kümesinde özkodlayıcı ve PCA ile çıkarılan özelliklerin karşılaştırılması görülebilmektedir. Şekilde X ve Y ekseni mevcut iki özelliği temsil etmektedir. Mevcut iki özellik kullanılarak PCA ile çıkarılan yeni özellik kırmızı ile, özkodlayıcı ile çıkarılan yeni özellik ise yeşil renk ile gösterilmektedir. Şekilde görüldüğü gibi özkodlayıcı ile çıkarılan yeni özellik, PCA ile çıkarılan özelliğe göre veri kümesini daha iyi temsil etmektedir.

Özkodlayıcı bir sinir ağı, kodlayıcı (encoder) ve çözümleyici (decoder) olarak iki ayrı katmandan oluşmaktadır. Eşitlik 2.9'de ϕ ile gösterilen kodlayıcı fonksiyon, orijinal X verisini darboğazda bulunan gizli bir F uzayına dönüştürür. ψ ile gösterilen



Şekil 2.3 Linear olmayan veri kümesinde özkodlayıcı ve PCA karşılaştırması [34]

çözümleyici fonksiyon ise darboğazdaki gizli F uzayını X çıktısına dönüştürür. Bu durumda çıktı ile girdi birbirine eşit olmaktadır.

$$\begin{aligned} \phi &: X \rightarrow F \\ \psi &: F \rightarrow X \\ \phi, \psi &= arg_{\phi,\psi} min \, \|X - (\psi \circ \phi)X\|^2 \end{aligned} \tag{2.9}$$

Kodlayıcı katman, giriş verisini (x), Eşitlik 2.10'de gösterilen aktivasyon fonksiyonunu ile darboğaz (z) denilen çıktıya dönüştürür. Darboğaz katmanında, genellikle girdi katmanından daha az özellik bulunmaktadır. Bu katmanda oluşan gizli özellikler (latent features) çıktının oluşturulması sırasında yeniden kullanılmaktadır.

$$z = \sigma(Wx + b) \tag{2.10}$$

Çözümleyici katman, kodlayıcı katmanda kodlanmış olan darboğaz çıktısını (z) giriş verisine benzer bir hale (x') getirmektedir. Eşitlik 2.11'te görüleceği üzere, bu katmanda kullanılan aktivasyon fonksiyonu (σ'), kodlayıcı katmandaki aktivasyon fonksiyonuna benzerlik göstermesine rağmen farklı ağırlıklara ve sapma değerlerine

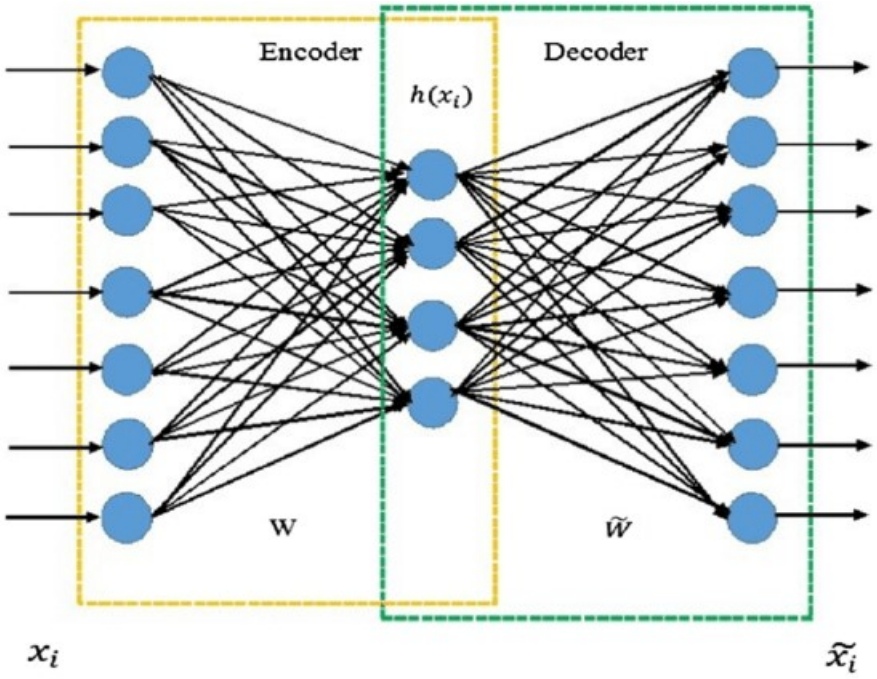
(bias) sahip olmaktadır.

$$x' = \sigma'(W'z + b') \tag{2.11}$$

Eşitlik 2.12'te görülen hata (loss) fonksiyonu (L), oluşturulan sinir ağını, standart geriye yayılım prosedürü ile eğitmek üzere kullanılmaktadır. Bu fonksiyon yardımı ile çıkışta üretilen veri kümesinin, giriş verisine benzerliği arttırılarak, çıkış verisinde oluşabilecek hatalar en aza indirilmektedir.

$$L(x, x') = \left\| x - x' \right\|^2 \tag{2.12}$$

Şekil 2.4'de özkodlayıcıya ait örnek bir sinir ağı modeli bulunmaktadır. Orta kısımda bulunan darboğaz katmanı ($h(x_i)$) kullanılarak özellik çıkarımı yapılabilmektedir. Bu katmanda bulunan düğümlerin sayısı özkodlayıcı kullanılarak çıkarılan yeni özellik kümesinin sayısına eşit olmaktadır. Darboğaz katmanında bulunan düğüm sayısı girdi kümesinde bulunan düğüm sayısından fazla, az veya eşit olabilmektedir.



Şekil 2.4 Özkodlayıcı model örneği [35]

Özkodlayıcının amacı, kodlayıcı ve çözümleyici fonksiyonları kullanarak, veri kümesini yeniden oluşturulabilecek şekilde kodlamak için minimum bilgiye ihtiyaç

duyacağımız şekilde seçmektir. Bu sebeple darboğaz katmanında kullanılan düğüm sayısı çok önemlidir. Darboğaz katmanında çok az düğüm kullanılması halinde veriyi yeniden oluşturma kapasitesi azalacaktır ve orijinalinden daha bulanık veri oluşacaktır. Çok fazla düğüm kullanılması halinde ise veri kümesinde sıkıştırılma yapılamayacağından özkodlayıcı işlemi anlamsız olacaktır.

2.4 Sentetik Veri Üretimi

Denetimli öğrenme modellerinin tahmin doğruluğu, büyük ölçüde eğitim sırasında kullanılan veri kümelerinin büyüklüğüne ve çeşitliliğine bağlıdır. Özellikle derin öğrenme modellerinin doğru bir şekilde eğitilebilmesi için büyük veri kümelerine ihtiyaç duyulmaktadır. Sentetik veri üretimi, veri miktarını arttırmak amacıyla, halihazırda var olan veriler kullanılarak, mevcut veri kümesine benzeyen yeni örnekler oluşturulmasını sağlayan bir tekniktir. Üstkümeleme (oversampling) yöntemine oldukça benzer bir yöntemdir. Sentetik veri üretiminin faydaları aşağıdaki gibi sıralanabilmektedir;

- Model tahminleme başarısını arttırır.
- Modelin eğitilmesi için eğitim kümesini büyütür.
- Aşırı öğrenme (overfitting) sorununu giderir.
- Veri kümesinin çeşitliliğini arttırır.
- Sınıflandırmada sınıf dengesizliklerinin (class imbalance) çözülmesini sağlar.
- Veri toplama ve etiketleme maliyetini azaltır.

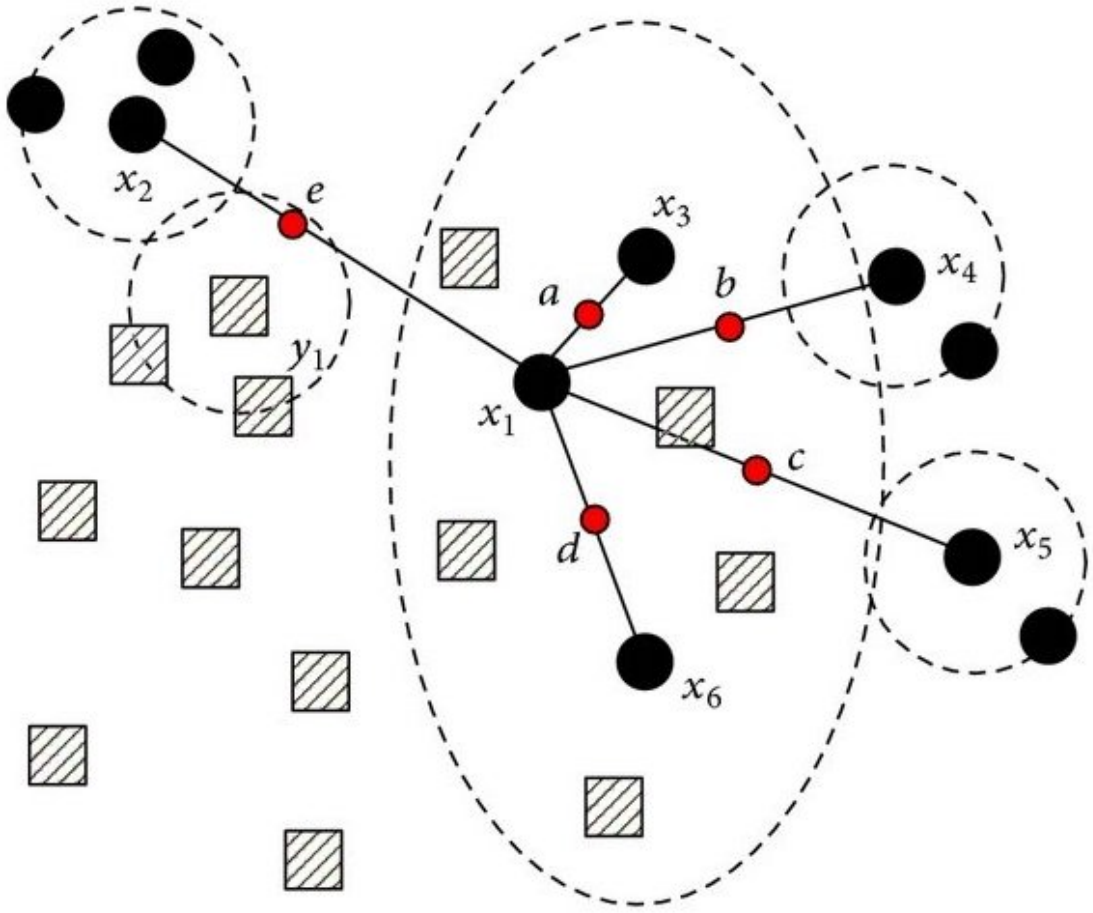
Sentetik veri üretimi için farklı yöntemler bulunmaktadır. Çevirme, döndürme, yeniden boyutlandırma, kırpma, hareket ettirme, Gauss gürültüsü (Gaussian noise) gibi yöntemler ile mevcut veri kümesinde bulunan örnekler üzerinde ufak değişiklikler yaparak sentetik veri üretimi yapılabilmektedir. Ayrıca son yıllarda geliştirilen derin öğrenme yöntemleri sayesinde, mevcut veri kümesindeki örnekler ile bir derin yapay sinir ağı eğitilerek gerçeğine benzer fakat aynı olmayan yeni sentetik veri örnekleri de üretilebilmektedir.

2.4.1 SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE [36], veri kümelerinde bulunan dengesizliği gidermek amacıyla, en yakın komşu (KNN) [27] yöntemini kullanarak azınlık olan sınıfa üstkümeleme uygulayan

bir yöntemdir. SMOTE, özellik uzayındaki, öklid uzaklığına göre yakın olan örnekleri seçerek, bu örnekler arasına bir çizgi çekmektedir. Bu çizgi üzerinde bulunan noktalar kullanılarak yeni örnekler oluşturulmaktadır.

Şekil 2.5'de SMOTE ile veri üretimine ait bir örnek bulunmaktadır. Şekilde kare ile çoğunluk sınıfına ait örnekler, siyah daire ile de azınlık sınıfına ait örnekler gösterilmektedir. Azınlık sınıfıa ait örneklerin arasına çekilen çizgilerin üzerinde bulunan kırmızı daire ile temsil edilen örnekler ise üretilen sentetik örnekleri göstermektedir.

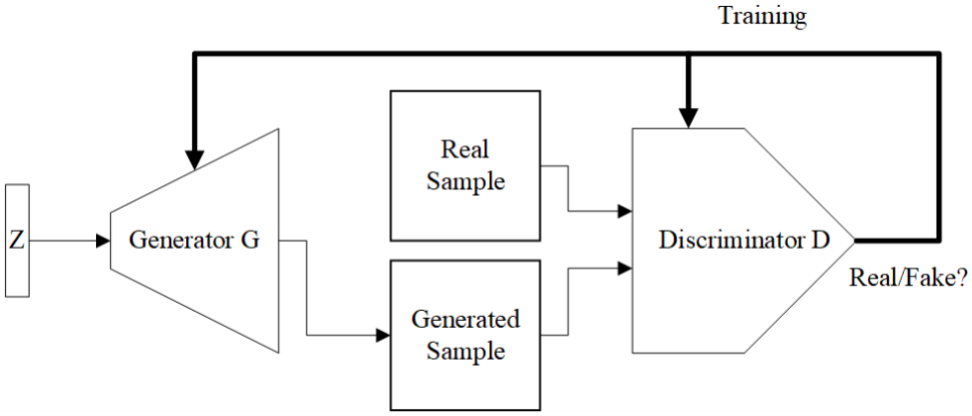


Şekil 2.5 SMOTE ile sentetik veri üretimi örneği [37]

2.4.2 Çekişmeli Üretici Ağlar (Generative Adversarial Networks)

Çekişmeli üretici ağlar (GAN) [38], 2014 yılında Ian Goodfellow ve meslektaşları tarafından icat edilmiştir. GAN, derin sinir ağları (DNN) veya evrişimsel sinir ağları (CNN) gibi derin öğrenme yöntemleri kullanılarak, sentetik veri üretimi sağlayan bir denetimsiz öğrenme yöntemidir. Girdi veri kümesinin kalıplarını kendiliğinden öğrenerek, çıktı verisini üretecek modeli eğitmektedir.

GAN yöntemi, sentetik veri üretimi sürecini iki küçük parçaya bölerek çözmektedir. Üretici (generator) ve ayrıştırıcı (discriminator) adı verilen iki adet yapay sinir ağı bulunmaktadır. Bu iki ağ karşılıklı bir şekilde, kazananı olamayan bir oyun oynamaktadır. Üretici ağ (G) gerçeğine çok benzeyen yeni örnekler üretmektedir. Ayrıştırıcı ağ (D) ise üretici ağın ürettiği örnekler içerisinden gerçeğine benzemeyenleri ayrıştırmaktadır. Karşılıklı olarak gerçekleşen bu üretim ve ayrıştırma işlemlerinin sonucunda gerçeğine çok benzeyen örnekler üretebilecek bir model oluşmaktadır. İki ağdan birisinin başarısız olması durumunda üretilen örneklerin başarısı da azalmaktadır. Şekil 2.6'de bir GAN modelinin yapısına yer verilmiştir.



Şekil 2.6 GAN modeli [39]

Algoritma 2'de GAN ile sentetik veri üretimi sürecine ait yarıkod görülmektedir.

Günümüzde oldukça popüler bir yöntem olan GAN, her geçen gün daha da gelişmektedir. Şekil 2.7'da Goodfellow tarafından yapılan ilk çalışmada üretilen sentetik örnekten, 2019 yılına kadar yapılan çalışmalarda GAN'ın gelişimi görülebilmektedir. Üretilen sentetik görseller ilk çalışmada siyah beyaz iken, sonraki çalışmalarda renkli görüntüler de üretebilir hale gelinmiştir. Görsellerin boyutları ve detayları da zamanla gelişim göstermiştir. Bu gelişimi sağlamak amacıyla farklı çalışmalarda kullanılan, geliştirilmiş GAN modelleri farklı isimler ile anılmaktadır.

GAN'ın gelişmesiyle birlikte farklı kullanım alanları da doğmuştur. Genellikle görüntü veri kümeleriyle kullanılan GAN, günümüzde video, ses, sinyal ve metin veri kümeleri ile yapılan çalışmalarda da kullanılmaktadır.

2.4.3 Wasserstein GAN

Wasserstein GAN (WGAN) [41], 2017 yılında üretilen geliştirilmiş bir GAN yöntemidir. Bu çalışmada amaçlanan klasik GAN'da bulunan hata (loss) fonksiyonunun değiştirilerek, daha dengeli bir sentetik veri üretimi süreci sağlamaktır. Klasik

Algoritma 2: Çekişmeli Üretici Ağlar Algoritması

İlklendirme: m, yığın (batch) sayısı ; n, adım (epoch) sayısı; k, ayrıştırıcı eğitimi adım sayısı

for $1 \ldots n$ **do**

  for $1 \ldots k$ **do**

    m sayısı kadar yeni gürültü örneği oluştur $(z_1, \ldots, z_m)$

    m sayısı kadar yeni sentetik örnek oluştur $(x_1, \ldots, x_m)$

    Ayrıştırıcı modelin ağırlıklarını güncelle:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log D(x_i) + log(1 - D(G(z_i)))]$$

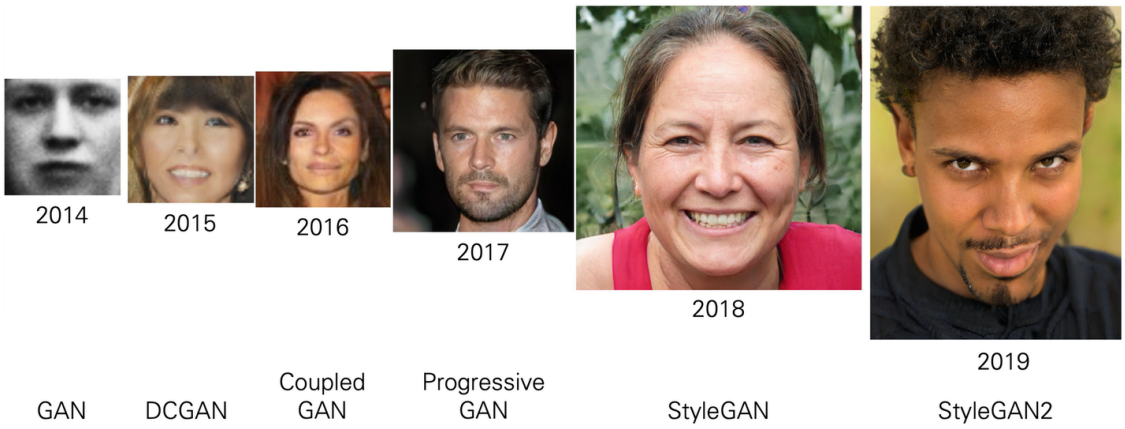
  end

  m sayısı kadar yeni gürültü örneği oluştur (z_i)

  Üretici modelin ağırlıklarını güncelle:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z_i)))$$

end



Şekil 2.7 Tarihsel süreç içerisinde GAN yönteminin gelişimi [40]

GAN'da kullanılan Jensen-Shannon uzaklığı (divergence) yerine Wasserstein mesafesi (distance) kullanılması önerilmektedir.

Jensen-Shannon uzaklığı hesaplanırken, Kullback-Leibler uzaklığı kullanılmaktadır. Eşitlik 2.13'de iki olasılık dağılımının $(\mathbb{P}_r, \mathbb{P}_g)$ arasındaki uzaklığı hesaplayan Kullback-Leibler hata fonksiyonu (KL) gösterilmektedir. İki dağılımın birbirine her noktadaki oranları alınarak, bu oranların logaritmalarının toplamı ile

hesaplanmaktadır

$$KL(\mathbb{P}_r||\mathbb{P}_g) = \int log\left(\frac{\mathbb{P}_r(x)}{\mathbb{P}_g(x)}\right)\mathbb{P}_r(x)d\mu(x) \tag{2.13}$$

Eşitlik 2.14'de iki olasılık dağılımının $(\mathbb{P}_r, \mathbb{P}_g)$ arasındaki uzaklığı hesaplayan Jensen-Shannon hata fonksiyonuna (JS) yer verilmiştir. Eşitlikte görülen $\mathbb{P}_m$, iki olasılık dağılımının ortalaması $(\mathbb{P}_r + \mathbb{P}_g/2)$ ile hesaplanmaktadır.

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r||\mathbb{P}_m) + KL(\mathbb{P}_g||\mathbb{P}_m) \tag{2.14}$$

İki olasılık dağılımı $(\mathbb{P}_r, \mathbb{P}_g)$ arasındaki Wasserstein uzaklığını hesaplayan fonksiyon (W) Eşitlik 2.15'de görülmektedir. Eşitlikte bulunan $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ iki olasılık arasındaki tüm ortak dağılım (joint distribution) fonksiyonlarını (γ) temsil etmektedir. γ, $\mathbb{P}_r$ dağılımını, $\mathbb{P}_g$ dağılımına dönüştürmek için ne kadar kütle (mass) taşınması gerektiğini hesaplamaktadır. Kütlelerin taşınması için gerekli olan optimum ulaşım planının maliyeti Wasserstein uzaklığıdır.

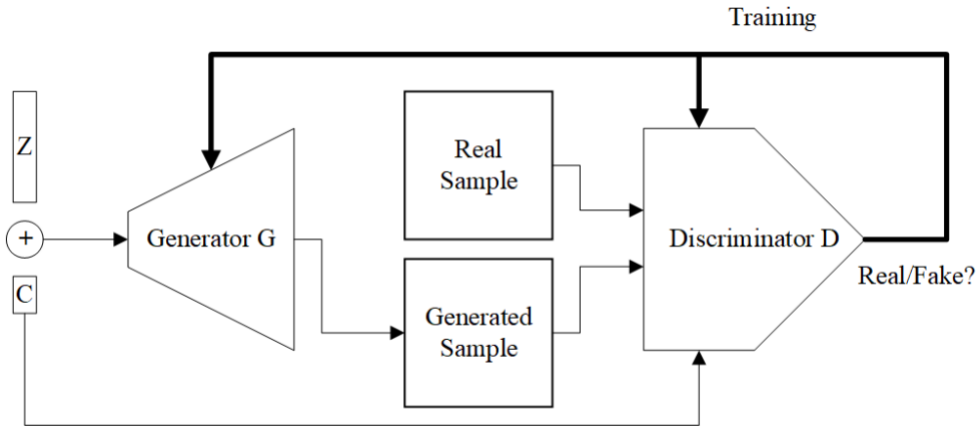
$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x - y||\right] \tag{2.15}$$

Daha dengeli bir eğitim sağladığı ve daha iyi veriler ürettiği için genellikle standart GAN fonksiyonu yerine WGAN tercih edilmektedir.

2.4.4 Koşullu GAN (Conditional GAN)

Klasik GAN modeli eğitimi sonucunda üretilen sentetik veri örnekleri rastgele oluşturulmaktadır. Üretilen sentetik örneklerin belirli bir koşula göre üretilmesini sağlamak amacıyla Koşullu GAN (CGAN) [42] geliştirilmiştir. CGAN üretilen verinin bazı karakteristik özelliklerini kontrol etmeyi amaçlamaktadır.

CGAN, hem üretici ağa, hem de ayrıştırıcı ağa eklenen ekstra bir etiket verisi ile üretilen verinin belirtilen koşula uymasını sağlamaktadır. Şekil 2.8'de CGAN'a ait bir model görülebilmektedir. Bu modelde koşul etiketi (C)'nin hem üretici ağın girdisinde, hem de ayrıştırıcı ağın çıktısında kullanıldığı görülebilmektedir.



Şekil 2.8 Koşullu GAN (CGAN) modeli [39]

Eşitlik 2.16'de (D) ayrıştırıcı model, (G) üretici model olmak üzere, GAN modellerinde kullanılan, G'yi ve D'yi minimize etmeyi amaçlayan değer (value) fonksiyonuna (V) yer verilmiştir.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_{data}(z)}[log(1-D(G(z)))] \quad (2.16)$$

Eşitlik 2.17'de ise koşul değişkeninine (y) bağlı olarak, örnek üretecek şekilde oluşturulmuş CGAN modeline ait değer fonksiyonu (V) bulunmaktadır. Koşul değişkeni (y) herhangi bir yardımcı bilgi olabilmektedir. Sınıf etiketi veya dışarıdan sağlanan başka bir etiket bilgisi koşul değişkeni olarak kullanılabilmektedir.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x|y)] + \mathbb{E}_{z\sim p_{data}(z)}[log(1-D(G(z|y)))] \quad (2.17)$$

2.4.5 Koşullu Tablo GAN (Conditional Tabular GAN)

GAN modelleri genellikle görüntü tabanlı veri kümeleri ile kullanılmaktadır. Görüntü veri kümelerinde bulunan özellikler tamamıyla nümeriktir. Ancak tablo veri kümelerinde hem nümerik hem de kategorik özellikler birlikte kullanılmatadır. Klasik GAN yöntemleri ile üretilen sentetik tablo veri kümelerinde, kategorik olan özellikler bozulmakta ve nümerik hale gelmektedir. Bu zorluğun üstesinden gelmek için geliştirilen koşullu tablo GAN (CTGAN) [23], nümerik özellikler ile birlikte kategorik özellikleri de başarı ile üretebilmektedir.

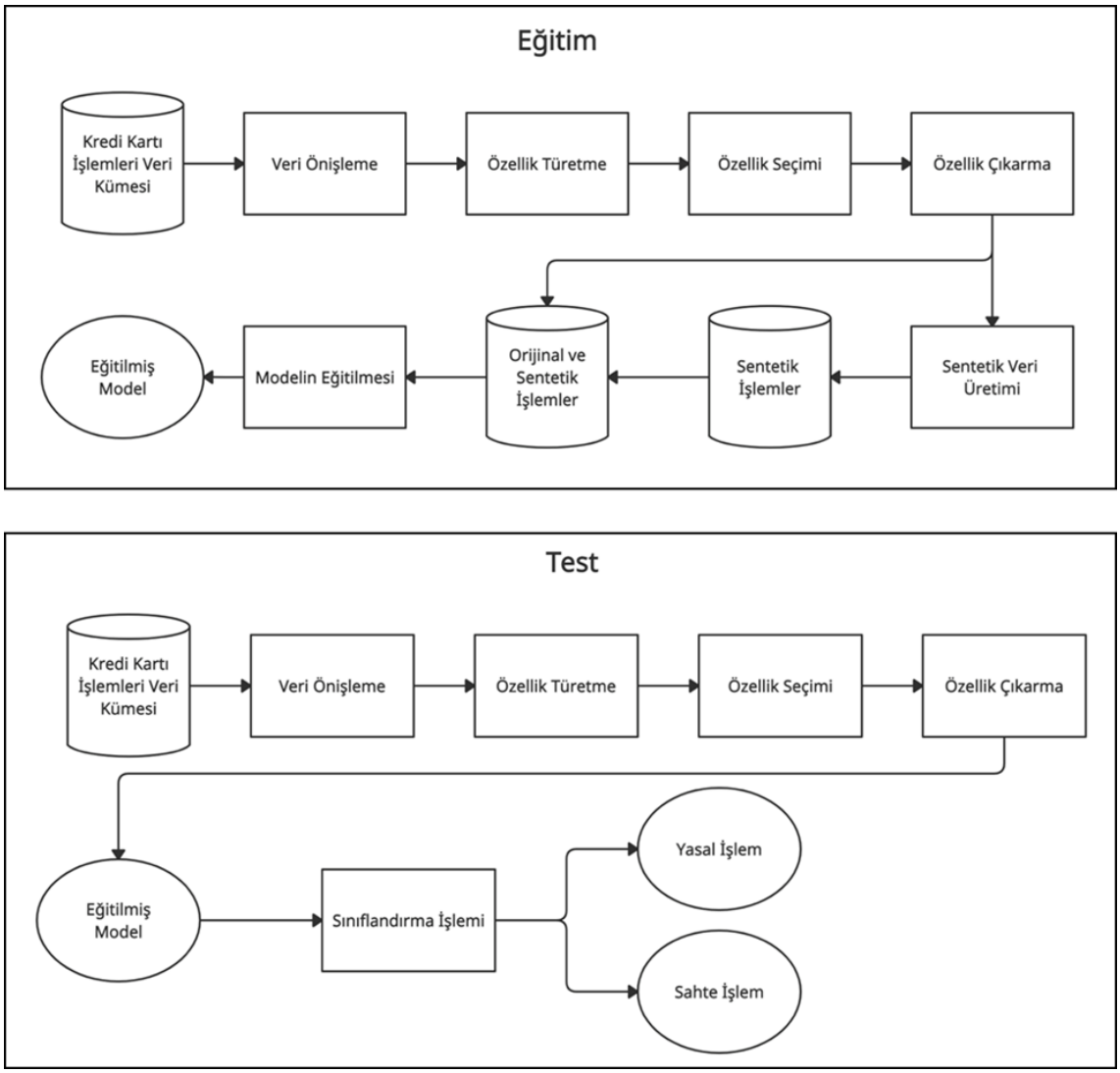
3
SİSTEM TASARIMI

Kredi kartı sahiplerinin yaptığı işlemler genellikle çeşitli yönlerden birbirlerine benzemektedir. Kart sahiplerinin düzenli olarak yaptığı işlemler, o kişinin harcama alışkanlığı olarak isimlendirilmektedir. Kart sahiplerinin harcama alışkanlıkları dışında yaptıkları işlemler riskli olarak kabul edilmektedir. Kredi kartı sahtekarları tarafından gerçekleştirilen sahte işlemler genellikle bu riskli işlem grubunda bulunmaktadır. Bu sebeple kart sahibinin harcama alışkanlığının doğru bir şekilde analiz edilmesi büyük önem taşımaktadır.

Kredi kartı sahtekarlıklarının tespit edilmesi günümüzde çoğunlukla kart sahibinin, kartından kendine ait olmayan bir harcama yapıldığını bildirmesiyle mümkündür. Kart sahibinin kendine ait olmayan harcamayı farketmemesi durumunda, sahtekarlar tarafından kartından birden fazla harcama yapılabilmesi mümkündür. Kart sahiplerinin harcama alışkanlıklarını analiz ederek, olası sahte işlemleri tespit edecek bir sisteme ihtiyaç duyulmaktadır.

Bu çalışmada klasik makine öğrenmesi ve derin öğrenme yöntemleri kullanılarak kredi kartı sahtekarlığı tespiti yapacak bir sistem tasarlanmış ve gerçeklenmiştir. Kart sahiplerinin geriye dönük işlemleri gruplandırılarak, harcama alışkanlıklarının daha iyi analiz edilmesi sağlanmıştır. Gruplandırılan işlemler kullanılarak yeni özellikler türetilmiş ve bu özelliklerden en başarılı olanları seçilmiştir. Özellik türetme ve seçme işlemlerinin sınıflandırma başarısına etkisi gözlemlenmiştir.

Kredi kartı işlemleri veri kümesinde sınıf dağılımı dengesiz olduğu için, GAN yöntemi kullanılarak sentetik veri örnekleri oluşturulmuştur. Yeni oluşturulan sentetik örnekler ile birlikte orijinal kredi kartı işlemleri kullanılarak, sınıflandırma işlemi yapılmıştır. Böylelikle, sentetik veri örnekleri üretiminin kredi kartı sahtekarlığı tespiti başarısına etkisi incelenmiştir. Şekil 3.1'de blok diyagramı şeklinde, bu çalışmada gerçekleştirilen adımlar gösterilmektedir. Bu bölümde, sistem tasarımında gerçekleştirilen adımların ayrıntılarına yer verilmiştir.



Şekil 3.1 Kredi kartı sahtekarlığı tespiti sistemi tasarımı

3.1 Veri Kümesi

Bu çalışmada kullanılan veri kümesi Türkiye'de bulunan 30 adet banka tarafından Bankalararası Kart Merkezi (BKM)'ye sağlanan, kredi kartı işlemlerinden oluşmaktadır. BKM ile yapılan gizlilik sözleşmesi çerçevesinde kimlik bilgileri gizlenmiş, gerçek kişi ve kurumlara ait kredi kartı işlemlerinden oluşmaktadır. Kullanılan veri kümesinde sahte ve yasal işlemler bir arada bulunmaktadır. Veri kümesindeki sahte işlemler, sahte olduğu bankalar tarafından BKM'ye bildirilen işlemlerdir.

Veri kümesinde bulunan işlemlerde, işlemin gerçekleştiği işyeri, işlem sırasında kullanılan kredi kartı, kart sahibi ve yapılan işlemin detaylarına ait bilgileri içermektedir. Kişisel verilerin korunması kanunu (KVKK) sebebiyle kişisel bilgiler (yaş, cinsiyet, meslek gibi) veri kümesinde bulunmamaktadır.

Kullanılan veri kümesinde 2019 yılına ait kredi kartı işlemleri bulunmaktadır. Aynı veri kümesiyle yapılan önceki çalışmada [43], çeşitli zaman aralıkları ile yapılan deneyler sonucunda, bir aylık işlemlerin eğitim, takip eden sonraki ayın ise test kümesi olarak seçilmesinin daha başarılı olduğu görülmüştür. Önceki çalışmalar dikkate alınarak bu çalışmada, 2019 yılına ait 6 aylık kredi kartı işlemleri kullanılarak 3 farklı eğitim ve test kümesi oluşturulmuştur. Ayrıca farklı zaman aralıklarının gözlenebilmesi amacıyla 2017 Mart ve Nisan ayları ile de deneyler yapılarak, sonuçlar karşılaştırılmıştır. Tablo 3.1'de veri kümesinde bulunan işlemlerin sayılarına yer verilmektedir.

Tablo 3.1 Veri kümesinde bulunan işlem sayıları

	Yasal	**Sahte**	**Toplam**
2017 Mart (Eğitim)	258,430	51,686	310,116
2017 Nisan (Test)	235,265	47,053	282,318
2019 Yılı (Toplam)	8,315M	618,000	8,315M
2019 Ocak (Eğitim)	93,620	18,724	112,344
2019 Şubat (Test)	76,760	15,352	91,112
2019 Mart (Eğitim)	73,570	14,714	88,284
2019 Nisan (Test)	69,455	13,891	83,356
2019 Mayıs (Eğitim)	65,700	13,140	78,840
2019 Haziran (Test)	68,845	13,769	82,614

Yurtdışında gerçekleştirilen sahte işlemlerin tespit edilmesi daha zor olduğundan, yasal olarak kabul edilen sahte işlemler bulunma olasılığı yüksektir. Yanlış etiketlenmiş veri örneklerinden kaynaklanabilecek öğrenme hatalarını gidermek amacıyla, yalnızca yurtiçinde gerçekleştirilen işlemler kullanılarak veri kümeleri oluşturulmuştur. Türkiye'de gerçekleştirilmiş farklı para birimleriyle yapılan işlemlere ait tutar bilgisi, o günün kuruna göre Türk Lirası (TL) karşılığına çevirilerek kullanılmaktadır. Veri kümesinde bulunan işlemin para birimi özelliği, işlemin yapıldığı gerçek para birimini temsil etmektedir.

Veri kümesinde sahte ve yasal işlemler dengesiz halde bulunmaktadır. 2019 yılına ait yasal işlemlerin sahte işlemlere oranı %0.00007'dir. Bu dengesizlik modelin başarısını olumsuz etkilemektedir. Bu sebeple modelin eğitiminde kullanılmak üzere oluşturulan veri kümesinde o aya ait tüm sahte işlemler ve sahte işlemlerin beş katı olacak şekilde rastgele yasal işlemler seçilerek, altkümeleme (undersampling) yapılmıştır. Sahte ve gerçek işlemlerin arasındaki bu oran [43] çalışmasında belirlenmiştir.

3.2 Veri Ön İşleme

Makine öğrenmesi modellerinde kullanılan veri kümesinin doğru bir şekilde oluşturulması oldukça önemlidir. Bu çalışmada kullanılan veri kümesindeki

işlemler 30 farklı banka tarafından sağlanmaktadır. Verilerin işlenmesi konusunda bankalar arasında bir standart olmadığı için, veri kümesinde eksik alanlar ve bazı uyumsuzluklar tespit edilmiştir. Geliştirilen modelin başarısını arttırmak amacıyla çeşitli ön işlemeler yapılarak veri kümesi hazırlanmıştır.

3.2.1 Veri Analizi

Kredi kartı işlemlerini içeren veri kümesinde toplamda 49 adet özellik bulunmaktadır. Bu özelliklerin her birisi, uzmanların da desteğiyle incelenmiştir. Çok fazla boş değer içeren, fazla kategoriden oluşan ve tutarsız değerler içeren özellikler göz ile elenmiştir. Bu çalışmada kullanılmak üzere toplam 29 adet özellik belirlenmiştir.

Veri kümesinde bulunan 29 adet özellik ve bu özelliklerin türüne Tablo 3.2'de yer verilmektedir. Bu özelliklerin 27 tanesi kategorik, 1 tanesi nümerik ve 1 tanesi de etiket bilgisine ait değerler içermektedir. Ayrıca kategorik özelliklerin içerdiği kategori sayıları da tabloda verilmiştir.

3.2.2 Veri Temizliği

Farklı bankalar tarafından veri kümesinde bulunan örnekler arasında tutarsızlıklar görülebilmektedir. Örneğin, boş olan değerlerin bazı örneklerde boş harf dizini (Empty String), bazı örneklerde ise null olarak gönderildiği tespit edilmiştir. Aynı değeri temsil eden bazı kategorilerin, farklı gönderimleri (büyük, küçük harf sorunu) mevcuttur. Bu gibi sorunları gidermek amacıyla, sistemde kullanılmadan önce veri kümesi temizlenmiştir.

3.2.3 Sayısal Özelliklerin Normalizasyonu

Veri kümelerinde bulunan sayısal değerlere, makine öğrenmesi yöntemlerinde kullanılmadan önce normalizasyon uygulanması genellikle başarıyı arttırmaktadır. Veri kümesine aykırı olan, çok büyük veya çok küçük değerler öğrenme sürecini olumsuz etkileyebilmektedir. Bu sorunu gidermek amacıyla nümerik değerleri -1 ile 1 arasına indirgeyen min-maks normalizasyonu [44] sıklıkla kullanılan bir yöntemdir.

Bu çalışmada kullanılan orijinal veri kümesinde, işlemin tutar değerini temsil eden bir adet nümerik özellik bulunmaktadır. Ayrıca özellik türetme sonucunda da 15 adet yeni nümerik özellik oluşmaktadır. Tüm nümerik özelliklere min-maks normalizasyonu uygulanarak aykırı değerlerin öğrenme sürecini olumsuz etkilemesinin önüne geçilmiştir.

Tablo 3.2 Veri kümesinde bulunan özellikler

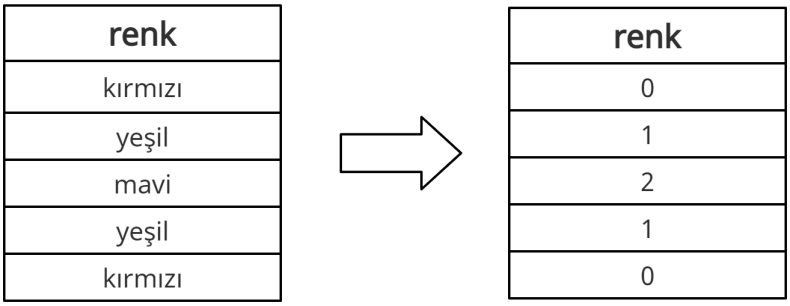
Özellik	Özellik Türü	Kategori Sayısı
İşlem Türü	Kategorik	6
İşlem Tutarı	Nümerik	-
İşlemin Para Birimi	Kategorik	4
Hesap Türü	Kategorik	3
ATM İşlemi mi?	Kategorik	2
Bankalar Arasında Marka Paylaşımı Var mı?	Kategorik	2
Geri Ödenme Durumu	Kategorik	2
Çip Kullanılma Durumu	Kategorik	7
Çip Kullanıldı mı?	Kategorik	2
Kart Sahibi Banka	Kategorik	31
Hizmet Alan Kart Sahibi Banka	Kategorik	2
Hizmet Alan POS Sahibi Banka	Kategorik	2
Taksit Sayısı	Kategorik	16
Kart Türü	Kategorik	3
Kart Ana Türü	Kategorik	6
Kart Alt Türü	Kategorik	8
İşlem Uluslararası mı?	Kategorik	2
İşyeri Kategorisi	Kategorik	308
Ödeme Yöntemi	Kategorik	13
Ödeme Türü	Kategorik	3
PIN girilme durumu	Kategorik	6
POS Kategorisi	Kategorik	5
POS Yetenek Kodu	Kategorik	8
POS Sahibi	Kategorik	27
İşlem Tipi	Kategorik	2
MOTO / E-Ticaret Tipi	Kategorik	11
EMV İşlem Türü	Kategorik	6
Kaynak Sistem	Kategorik	25
İşlem Sahte mi?	Etiket	-

3.2.4 Kategorik Özelliklerin Kullanımı

Veri kümesinde bulunan kategorik değerlerin nümerik olmaması durumunda, kullanılabilecek makine öğrenmesi yöntemleri kısıtlanmaktadır. Örneğin, bu çalışmada kullanılan özkodlayıcılar, yalnızca nümerik veriler ile kullanılabilmektedir. Veri kümesindeki kategorik özelliklerin, kategorik değerlerini kaybetmeden nümerik değerlere dönüştürülmesi mümkündür. Bu dönüşüm için en sık kullanılan yöntemlerden birisi etiket kodlama (label encoding) ve one-hot kodlama (one-hot encoding) yöntemidir.

Etiket kodlama yönteminde her kategoriye karşılık gelen değer, yeni bir nümerik değer ile temsil edilmektedir. Şekil 3.2'de etiket kodlamaya ait bir örnek bulunmaktadır.

Örnekte görüldüğü gibi, her kategori bir nümerik değer ile değiştirilir. Etiket kodlama sonucunda veri kümesindeki özellik sayısı değişmemektedir.



Şekil 3.2 Etiket kodlama örneği

Kategorik değerlerin nümerik olarak temsil edilmesini sağlayan bir diğer yöntem one-hot kodlamadır [45]. One-hot kodlama sırasında her kategori için yeni bir özellik eklenmektedir. Yeni özellikler kullanılarak kategoriler ikili (binary) vektörler ile temsil edilmektedir. Şekil 3.3'de one-hot kodlama sürecine ait bir örnek bulunmaktadır. Şekilde görüldüğü gibi renk özelliği yerine 3 yeni özellik eklenmiştir. Her kategoriye ait özellikler '1' ve '0' değerleriyle temsil edilmektedir.

Şekil 3.3 One-hot kodlama örneği

One-hot kodlama sonrasında, kategorik özelliklerde bulunan farklı kategori sayısına bağlı olarak, veri kümesindeki özellik sayısı artmaktadır. Bu yöntemin fazla sayıda kategorik değere sahip özelliklerden oluşan veri kümelerinde kullanılması, özellik sayısını çok fazla arttıracağından dolayı tavsiye edilmemektedir. Bu çalışmada kullanılan veri kümesindeki kategorik özellikler, fazla sayıda farklı kategori değeri içermediği için one-hot kodlama yöntemi kullanılmıştır. One-hot kodlama sonucunda 432 adet özellik elde edilmiştir.

3.3 Yeni Özellik Türetme

Kredi kartı sahtekarlığı tespitinde, kart sahibinin harcama alışkanlıklarının anlaşılması oldukça önemlidir. Kart sahipleri genellikle benzer harcamalar yapmaktadır. Bu sebeple kredi kartı harcamalarının geriye dönük analiz edilmesi önem taşımaktadır.

Örneğin, aylık olarak ortalama 1.000 TL harcama yapan bir kişinin tek seferde 15.000 TL işlem yapması aykırı (anomaly) bir durum olarak kabul edilmektedir. Bu tarz aykırılıkların daha iyi anlaşılabilmesi için yeni özellikler türetilmesi önerilmektedir.

Kredi kartı işlemlerinde sahte işlemler ve yasal işlemler birbirine oldukça benzerdir. Bu durum sahtekarlık tespitini zorlaştırmaktadır. İşlemlere eklenen yeni özellikler kart sahibinin karakteristiğinin, makine öğrenmesi yöntemleri tarafından daha iyi anlaşılmasını sağlamaktadır.

Bu çalışmada özellik türetilirken, öncelikle uzman desteğine başvurulmuştur. Veri kümesinde bulunan MOTO özelliği, çok fazla kategoriyi aynı anda temsil etmektedir. Bu değerlerin daha iyi anlaşılabilmesi için uzmanlar tarafından belirlenen üç adet özellik eklenmiştir. Bu üç özellik;

1. **Güvenlik Seviyesi:** İşlem sırasında 3 boyutlu doğrulama (3D Secure) kullanıldı mı?
2. **İşlem Çevrimiçi mi?:** İşlemin çevrimiçi veya fiziksel olarak gerçekleştirilme durumu
3. **İşlem Gerçekleştirilme Türü:** İşlemin hangi ortamda gerçekleştirildiği bilgisi

şeklinde sıralanabilmektedir. Bu özellikler işleme ait diğer bilgiler kullanılarak elde edilmiştir.

Özellik türetme için kullanılan bir diğer yöntem ise gruplama yöntemidir. Kart sahibininin gerçekleştirdiği işlemler, belirli zaman aralıklarına göre gruplandırılarak, yeni özellikler türetilmesi mümkündür. Örneğin, aynı kredi kartıyla, aynı iş yerinde, son bir hafta içerisinde yapılmış işlemlerin ortalama tutarı yeni bir özellik olarak veri kümesine eklenebilmektedir.

Gruplama yöntemiyle özellik türetme yapan çalışmalar incelenerek gruplama için kullanılacak özellikler, zaman aralıkları ve hesaplama türleri belirlenmiştir. İşlem gruplama günlük, haftalık ve aylık olacak şekilde üç farklı zaman aralığında gerçekleştirilmiştir. Gruplanan her işlem grubu için, toplam tutar, ortalama tutar, işlem sayısı ve işlem var mı bilgileri hesaplanmaktadır. Aşağıdaki sekiz adet özelliğe ait işlemler belirlenen zaman aralıklarına göre gruplandırılarak, belirlenen hesaplama türleri ile hesaplanmış ve veri kümesine eklenmiştir. Gruplama için kullanılan sekiz özellik;

1. **Tüm işlemler:** Herhangi bir özelliğe göre gruplamadan belirlenen zaman aralığındaki tüm işlemler
2. **İşyeri Kategorisi:** Mevcut işleme ait işyeri kategorisinde yapılan önceki işlemler
3. **MOTO / E-Ticaret Tipi:** Mevcut işleme ait e-ticaret tipinde yapılan önceki işlemler
4. **Ödeme Türü:** Mevcut işleme ait ödeme türüyle yapılan önceki işlemler
5. **Ödeme Yöntemi:** Mevcut işleme ait ödeme yöntemiyle yapılan önceki işlemler
6. **İşlem Çevrimiçi mi?:** Mevcut işleme ait çevrimiçi / fiziksel bilgisiyle yapılan önceki işlemler
7. **Güvenlik Seviyesi:** Mevcut işleme ait güvenlik seviyesinde yapılan önceki işlemler
8. **İşlem Gerçekleştirilme Türü:** Mevcut işleme ait işlem gerçekleştirilme türü ile yapılan önceki işlemler

şeklinde sıralanabilmektedir. Gruplama işlemi için, uzmanlar tarafından belirlenen ve özellik kümesine yeni eklenen özellikler de kullanılmıştır. Gruplama yöntemi kullanılarak toplamda 96 adet yeni özellik türetilmiştir.

Bu çalışmada uzmanlar tarafından belirlenen 3 adet ve gruplama yöntemiyle oluşturulan 96 adet özellik türetilmiştir. Veri kümesinde, yeni türetilen 99 adet özellik ve mevcutta bulunan 28 adet özellik ile birlikte toplamda 127 adet özellik bulunmaktadır.

3.4 Özellik Seçme

Veri kümesinde bulunan özellikler doğru bir şekilde seçilmediğinde, tahminleme başarısını kötü etkileyebilmektedir. Ayrıca veri kümesinde fazla özellik bulunması modelin eğitim süresini de uzatmaktadır. Özellik seçimi doğru bir şekilde gerçekleştirildiğinde, tahminleme başarısını arttırmaktadır. Ayrıca herhangi bir aşırı öğrenme (overfitting) durumunun da önüne geçilebilmektedir.

Bu çalışmada kullanılan veri kümesinde bulunan özelliklerin sayısı, özellik türetme işlemleri sonucu 127'ye yükselmiştir. Türetilen yeni özelliklerin tahminleme başarısına etkisinin ölçülüp, etkisi olmayanların elenmesi gerekmektedir. Özelliklerin başarısının ölçülmesi amacıyla;

1. Yinelemeli Özellik Seçimi (Recursive Feature Selection)
2. Tek Değişkenli Özellik Seçimi (Univariate Feature Selection)
3. Rastgele Orman (Random Forest)
4. Son Derece Rastgele Ağaçlar (Extremely Randomized Trees)
5. SHAP (Shapley Additive Explanations)

yöntemleri kullanılmıştır. Bu yöntemler sonucu özellikler en önemliden, en önemsize doğru sıralanmıştır. Bu yöntemlerden RFE, RF ve SHAP birbirine çok yakın sonuçlar vermektedir. Bu üç yöntem arasından en başarılı olan RFE seçilerek, özellik seçimi bu yönteme göre uygulanmıştır.

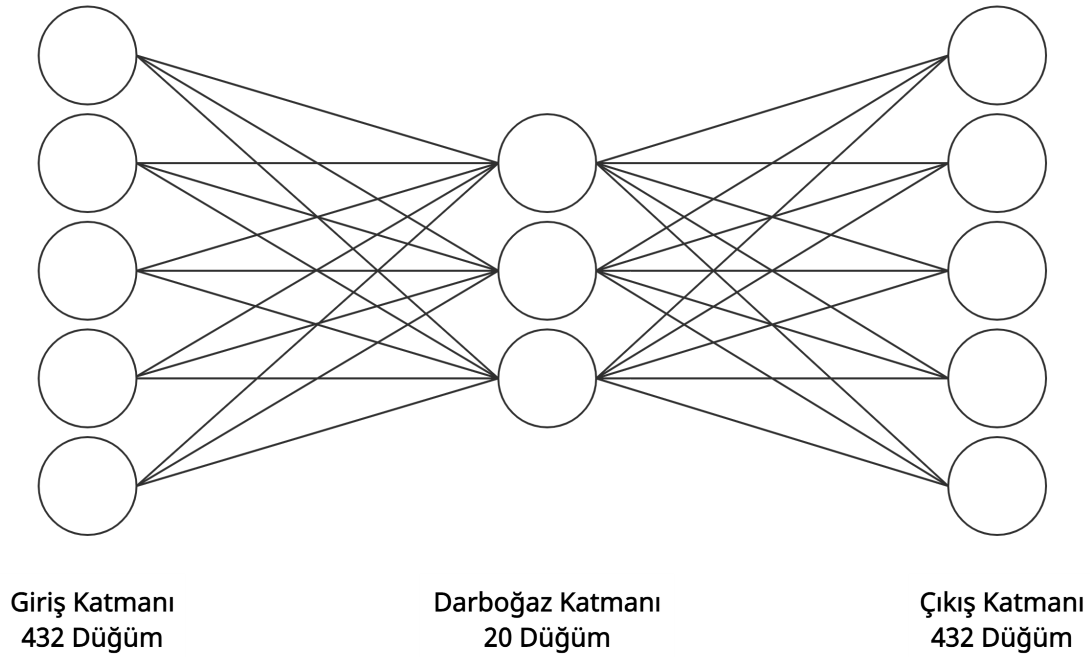
Özellik seçimi işlemi ile 127 adet özellik, sırasıyla en başarılı olan 5, 10, 15, 20, 25, 30, 40, 60, 80 ve 100 adet özelliğe indirgenmiştir. Yapılan testler sonucunda en başarılı olan 30 özellik ile tahminleme başarısının daha iyi olduğu tespit edilmiştir. Çalışmanın ilerleyen kısımlarında seçilen bu 30 adet özellik kullanılmıştır.

3.5 Özellik Çıkarma

Veri kümelerinin karakteristiğinin, makine öğrenmesi yöntemleri tarafından daha iyi anlaşılabilmesi ve az bilgi içeren özelliklerin elenmesi gibi amaçlarla özellik çıkarma yöntemleri kullanılmaktadır. Ayrıca GAN ile sentetik veri üretimi yapılırken, kategorik özellikler korunamayacağı için özellik çıkararak nümerik bir uzaya geçirilmesi üretilen örneklerin başarısını arttırmaktadır.

Bu çalışmada PCA ve özkodlayıcı model kullanılarak özellik çıkarımı yapılmıştır. PCA ile yapılan özellik çıkarımı başarılı olmadığı için tüm deneyler özkodlayıcı model ile özellik çıkarılarak yapılmıştır. Özkodlayıcının darboğaz katmanında oluşan özellikler kullanılarak veri kümesi yeni bir uzayda ifade edilmiştir. One-hot kodlama sonucunda oluşan 432 özellik, 10, 20, 40 ve 80 adet özellik çıkarılarak sınıflandırma deneyleri yapılmıştır. Yapılan deneyler sonucu en başarılı olanın 20 adet özellik ile olduğu tespit edilmiştir. Çalışmanın geri kalanında kullanılan veri kümesinde, özkodlayıcı tarafından çıkarılmış 20 adet nümerik özellik bulunmaktadır. Şekil 3.4'de oluşturulan özkodlayıcı modelin mimarisine yer verilmiştir. Darboğaz katmanında bulunan 20 adet düğüm kullanılarak, özellik çıkarımı yapılmıştır.

Özkodlayıcılar ile özellik çıkarımı sonucu GAN ile sentetik veri üretimi için veri kümesi daha uygun hale getirilmiştir. Veri kümesinde bulunan kategorik özellikler nümerik



Şekil 3.4 Özellik çıkarmak için kullanılan özkodlayıcı modelin mimarisi

hale getirilmiştir. Veri kümesi, karakteristiğini daha iyi ifade eden yeni bir uzaya taşınmıştır. Ayrıca özellik sayısı azaltılarak aşırı öğrenmenin önüne geçilmiştir.

3.6 Sentetik Veri Üretme

Veri kümelerinin dengesiz olması veya veri kümesinde bulunan örneklerin yetersiz olması gibi sebeplerle, tahminleme başarı düşük olmaktadır. Bu sorunun giderilmesi için mevcut örnekler kullanılarak sentetik veri üretimi yapılmaktadır. Üretilen örneklerin gerçeklerine olan benzerliği, sistem başarısı için oldukça önemlidir.

Bu çalışmada veri kümesindeki dengesizliğin giderilmesi amacıyla ilk olarak SMOTE yöntemi ile üstkümeleme yapılmıştır. Üstkümeleme için farklı oranlarda sentetik veri üretilerek deneyler yapılmış, sınıf dağılımının başarıya olan etkisi incelenmiştir.

Sentetik veri üretimi için son zamanlarda kullanılan en başarılı yöntemlerden birisi GAN'dır. Ancak GAN yöntemi genellikle görüntü veri kümeleri ile kullanılmaktadır. Bu çalışmada kredi kartı veri kümesinde GAN ile üretilen örneklerin sınıflandırma performansına etkisi incelenmiştir.

Zaman içerisinde belirli görevleri gerçekleştirmek üzere farklı GAN yöntemleri geliştirilmiştir. Yapılan çalışmada farklı GAN modellerinin başarıya olan etkisini incelemek amacıyla;

- Klasik GAN,
- Belirli bir sınıfa ait örnekleri üretmeyi amaçlayan CGAN,
- Farklı bir hata fonksiyonu kullanarak daha başarılı örnekler üretmeyi amaçlayan WGAN,
- Kategorik özellikleri koruyarak, tablo şeklindeki veri kümelerinde sentetik veri üretmeyi amaçlayan CTGAN

yöntemleri kullanılarak sentetik örnekler üretilmiştir. GAN modelleri eğitilirken özkodlayıcı ile çıkarılmış özellikler kullanılmıştır.

CTGAN yöntemi, veri kümesini kendi içerisinde özkodlayıcı model ile farklı bir uzaya taşıyarak kullanmaktadır. Bu sebeple CTGAN modeli eğitilirken, özkodlayıcı kullanılmadan, doğrudan özellik seçimi sonucu elde edilen 30 adet özellik kullanılarak da deneyler yapılmıştır. Böylelikle CTGAN modelininde özkodlayıcı ile özellik çıkarımının sınıflandırma başarısına etkisi ölçülmüştür.

Üretilen sentetik örnek sayısının tahminlemeye olan etkisini görmek amacıyla farklı sayılarda üretilen sentetik örnekler, mevcut örneklere eklenerek sınıflandırılma yapılmıştır. Üretilen sentetik örneklerden, sadece sahte olan işlemler kullanılarak deneyler yapıldığı gibi, sahte/yasal işlem dengesi korunacak şekilde her iki sınıftan üretilen örnekler kullanılarak da deneyler yapılmıştır. Ayrıca üretilen veri kümesinin kalitesini ölçmek amacıyla, mevcut örnekler kullanılmadan, tamamen sentetik örnekler ile eğitim yapılmıştır.

3.7 Kredi Kartı Sahtekarlığı Tespiti Modeli

Kredi kartı sahtekarlıklarını tespit edebilmek amacıyla geliştirilen sistemde, özellik çıkarma ve sentetik veri üretimi yapılırken makine öğrenmesi ve derin öğrenme yöntemleri bir arada kullanılmıştır. Sınıflandırma için klasik makine öğrenmesi yöntemleri ve derin öğrenme yöntemleri kullanılarak deneyler yapılmıştır.

3.7.1 Klasik Makine Öğrenmesi Yöntemleri ile Sahtekarlık Tespiti

Bu çalışmada önce klasik makine öğrenmesi yöntemlerinin sınıflandırma başarısını ölçmek amacıyla, rastgele orman (RF), karar ağaçları (DT), destek vektör makinesi (SVM), lojistik regrasyon (LR) ve naive bayes (NB) yöntemleri denenmiştir.

Kullanılan tüm makine öğrenmesi yöntemleri ile sınıflandırma yapılırken kullanılan hiperparametreleri belirlemek için ızgara (grid) araması [46] yöntemi kullanılmıştır.

Izgara araması yönteminde parametrelerin alabileceği değerler kümesi oluşturularak, her farklı kombinasyon ile sınıflandırma yapılmıştır. En başarılı sonucu veren parametreler belirlenmiş ve çalışmanın ilerleyen kısımlarındaki deneylerde bu hiperparametreler kullanılmıştır.

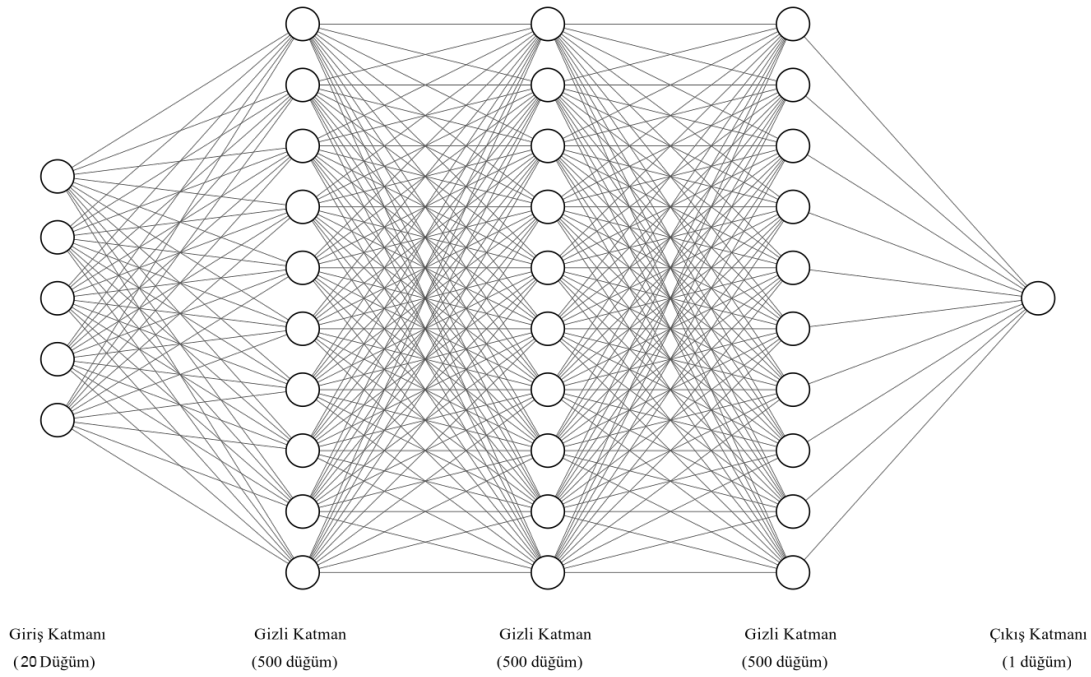
Kredi kartı sahtekarlığı tespiti için en çok tercih edilen yöntem rastgele orman yöntemidir. Çalışmada kullanılan veri kümesiyle yapılan önceki bir çalışmada [43] da en başarılı sınıflandırma yönteminin rastgele orman olduğu tespit edilmiştir. Bu sebeplerle, çalışmadaki farklı deneyleri yapmak amacıyla, ana sınıflandırıcı olarak rastgele orman kullanılmıştır.

3.7.2 Derin Öğrenme Tabanlı Sahtekarlık Tespiti

Derin öğrenme yöntemleri genellikle görüntü veri kümelerinde kullanılmakta ve başarılı sonuçlar vermektedir. Derin öğrenme yöntemlerinin tablo veri kümelerinde de kullanılması mümkündür. Özellikle fazla sayıda örnek içeren veri kümelerinde derin öğrenmenin daha başarılı olduğu iddia edilmektedir. Çalışmada derin öğrenmenin etkisini incelemek amacıyla farklı derin sinir ağları (DNN) mimarileri oluşturulmuş ve deneyler yapılmıştır.

DNN farklı katman ve yapılardan oluşabilmektedir. Bu çalışmada kullanılan DNN modeli Şekil 3.5'de görülmektedir. Giriş katmanı, veri kümesinde bulunan özkodlayıcı ile çıkarılmış özellik sayısı olan 20 adet düğümden oluşmaktadır. Giriş katmanındaki düğüm sayısı eğitim sırasında kullanılan veri kümesinin özellik sayısına göre değişkenlik göstermektedir. Örneğin, one-hot kodlama yapılarak oluşturulmuş veri kümesi ile model eğitilirken 449 adet giriş düğümü kullanılmaktadır. Üç adet 500 düğümden oluşan gizli katman (hidden layer) kullanılmıştır. Bu katmanlarda aktivasyon fonksiyonu olarak ReLU (Rectified Linear Unit) kullanılmıştır. Çıkış katmanından önce aşırı öğrenmeyi (overfit) engellemek amacıyla 0.25 oranında dropout uygulanmıştır. Çıkış katmanı ise veri kümesinin etiket bilgisini belirleyen 1 adet düğümden oluşmaktadır. Çıkış katmanında aktivasyon fonksiyonu olarak sigmoid kullanılmıştır. Ayrıca deneyler sırasında aktivasyon fonksiyonu olarak LeakyReLU yöntemi de uygulanarak başarısı ölçülmüştür.

Eşitlik 3.1'de ReLU'ya ait aktivasyon fonksiyonuna yer verilmiştir. (x) düğümün girdisi olmak üzere, ReLU, yalnızca pozitif değerler üretmektedir. ReLU, işlem hızının yüksek



Şekil 3.5 Çalışmada kullanılan derin öğrenme modeli

olmasından dolayı sıklıkla tercih edilmektedir.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{3.1}$$

Eşitlik 3.2'de LeakyReLU'ya ait aktivasyon fonksiyonu bulunmaktadır. ReLU'nun aksine LeakyReLU, kullandığı 0.01 katsayısı ile öğrenmeyi negatif bölgeler için de devam ettirmektedir.

$$f(x) = \begin{cases} 0.01x, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{3.2}$$

Eşitlik 3.3'de sigmoid fonksiyonuna (S) yer verilmiştir. (x) düğümün girdisi olmak üzere, sigmoid fonksiyon 0 ile 1 arasında bir değer üreterek, çıktının hangi sınıfa ait olduğunun daha iyi anlaşılmasını sağlar. Bu sebeple çıkış katmanında sigmoid

fonksiyonu kullanılmıştır.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

Derin yapay sinir ağı modelini iyileştirme amacıyla farklı iyileştirici (optimizer) fonksiyonlar kullanılarak deneyler yapılmıştır. Derin öğrenme çalışmalarında sıklıkla kullanılan Adam, SGD (Stochastic Gradient Descent) ve RMSprop yöntemleri denenmiştir. Alınan sonuçlara göre en başarılı performansı gösteren, Adam iyileştirici fonksiyonu ile model oluşturulmuştur.

Oluşturulan derin yapay sinir ağı, 128'lik yığınlar (batch) halinde 100 devirde (epoch) eğitilmiştir. Bu devirin üzerindeki sayılarda yapılan deneyler, aşırı öğrenmeye sebep olduğu için eğitim 100 devirde tamamlanmıştır.

4
DENEYSEL SONUÇLAR

Kredi kartı sahtekarlıkları tespiti için önerilen uygulama geliştirilirken her aşamada bir çok farklı deney yapılmıştır. Bu bölümde yapılan deneylerin detaylarına ve sonuçlarına yer verilecektir.

Deney sonuçlarını incelemek amacıyla her deneyde, kesinlik (precision), duyarlılık (recall) ve f1-skor (f1-score) parametreleri kullanılmıştır. Sonuçlar sahte işlemler ve yasal işlemler için ayrı ayrı elde edilmiştir. Kredi kartı sahtekarlığı tespiti sistemi ile asıl hedeflenen sahte işlemleri tespit etmektir. Bu sebeple tahminleme başarısı, sahte işlemlere ait f1-skor kullanılarak ölçülmüştür.

Türkiye'de her gün yüz binlerce kredi kartı işlemi gerçekleştirilmektedir. Sahte işlemleri sınıflandırma başarısındaki %1'lik bir artış dahi kişi ve kurumların binlerce TL'sinin sahtekarlar tarafından ele geçirilmesini önlemektedir. Deney sonuçlarında bulunan, sahte işlemleri doğru tespit etme başarısı incelenirken bu durum mutlaka göz önünde bulundurulmalıdır.

4.1 Test Ortamı

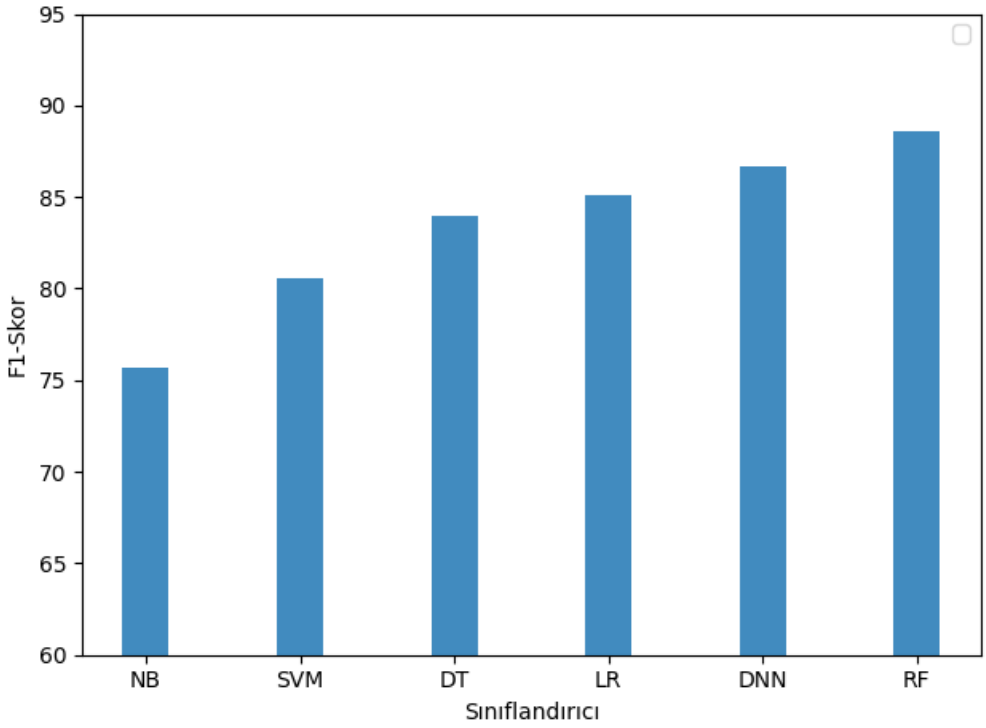
Önerilen kredi kartı sahtekarlığı tespiti sisteminde makine öğrenmesi ve derin öğrenme yöntemleri bir arada kullanılmıştır. Özellikle derin öğrenme yöntemleri yüksek işlemci gücüne ihtiyaç duymaktadır. Bu sebeple çalışma boyunca, yapay zeka çalışmalarında kullanılmak üzere tasarlanan, grafik işlemcisi destekli IBM PowerAI makinesi kullanılmıştır.

Kredi kartı işlemlerine ait veri kümesini kullanmak üzere yeni bir test ortamı oluşturulmuş ve veritabanı buraya aktarılmıştır. Uygulama Python dili ile geliştirilmiştir. Geliştirme aşamasında Tensorflow, Keras, Pytorch, Sklearn kütüphaneleri kullanılmıştır.

4.2 Sınıflandırıcı Seçimi

Kredi kartı veri kümesiyle yapılan çalışmalarda, rastgele ağaç yönteminin genellikle daha başarılı olduğu görülmüştür. Bu çalışmada, farklı makine öğrenmesi ve derin öğrenme yöntemleri karşılaştırılmıştır. Makine öğrenmesi yöntemlerinden naive bayes (NB), destek vektör makinesi (SVM), lojistik regrasyon (LR), karar ağaçları (DT) ve rastgele orman (RF) kullanılımıştır. Derin öğrenme yöntemi olarak ise derin yapay sinir ağları (DNN) kullanılmıştır. 2019 Ocak ayına ait örnekler ile model eğitilerek, 2019 Şubat ayına ait örnekler ile sınıflandırma başarısı test edilmiştir.

Yapılan sınıflandırma deneyleri sonucu Şekil 4.1 ve Tablo 4.1'de görülmektedir. Rastgele ağaç, bu çalışmada'da en başarılı yöntem olarak seçilmiştir. Çalışmanın geri kalan kısmında yapılan deneylerde, rastgele ağaç yöntemi kullanılmıştır. Sınıflandırma işlemi yapılırken özellik seçimi sonucu oluşan 30 adet özellik one-hot kodlama ile kodlanarak 449 özelliğe çıkarılmış ve bu özellikler kullanılmıştır. One-hot kodlama, veri kümesinde bulunan farklı kategorik değerlerin sayısına göre özellik sayısını arttırdığı için, farklı veri kümeleriyle yapılan deneylerde bu sayı değişiklik gösterebilmektedir.



Şekil 4.1 Farklı sınıflandırıcıların karşılaştırılması

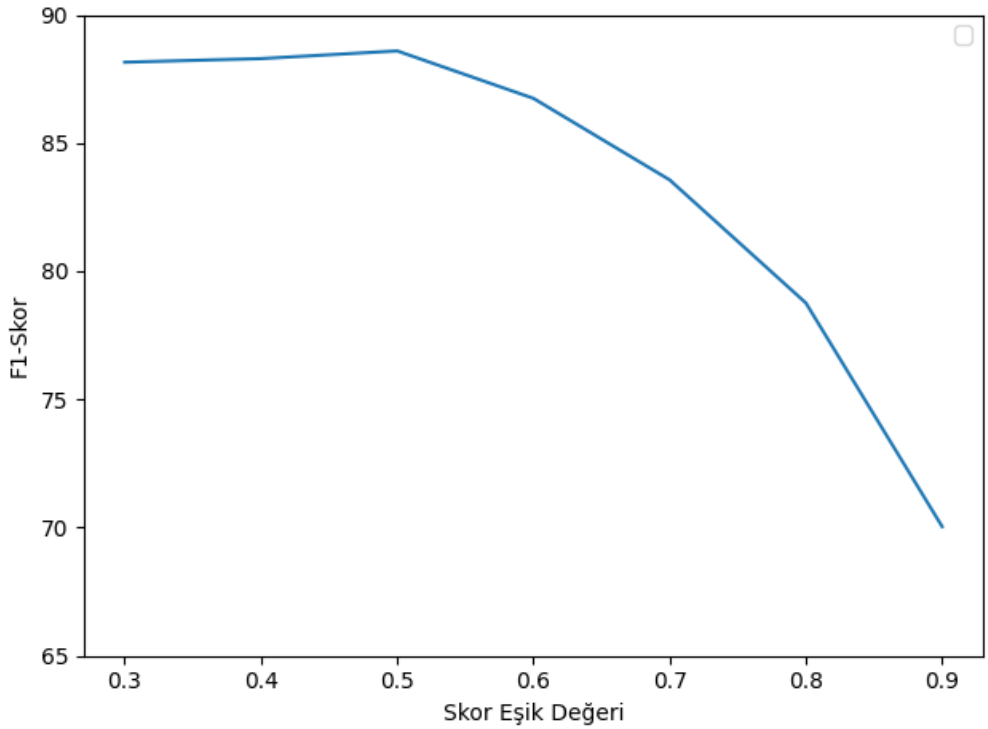
Farklı deneyler sonucu oluşturulan en iyi derin öğrenme modeli olan DNN sonuçları da şekilde görülmektedir. DNN, örnekleri %86.63 başarı ile sınıflandırırken, rastgele orman ise %88.61 başarı ile sınıflandırmıştır. Yani, klasik makine öğrenmesi

Tablo 4.1 Farklı sınıflandırıcıların karşılaştırılması

Sınıflandırıcı	Kesinlik		Duyarlılık		F1-Skor	
	Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
Naive Bayes	97.45%	66.25%	91.02%	88.12%	94.13%	75.64%
Destek Vektör Makinesi	96.05%	80.87%	96.20%	80.26%	96.13%	80.57%
Karar Ağaçları	96.43%	85.91%	97.31%	82.03%	96.87%	83.93%
Lojistik Regrasyon	96.81%	86.21%	97.31%	83.96%	97.06%	85.07%
Derin Sinir Ağları	96.94%	88.78%	97.86%	84.58%	97.40%	86.63%
Rastgele Orman	**97.47%**	**89.98%**	**98.05%**	**87.29%**	**97.76%**	**88.61%**

yöntemlerinden rastgele orman, derin öğrenmeden daha başarılı bir şekilde örnekleri sınıflandırmaktadır.

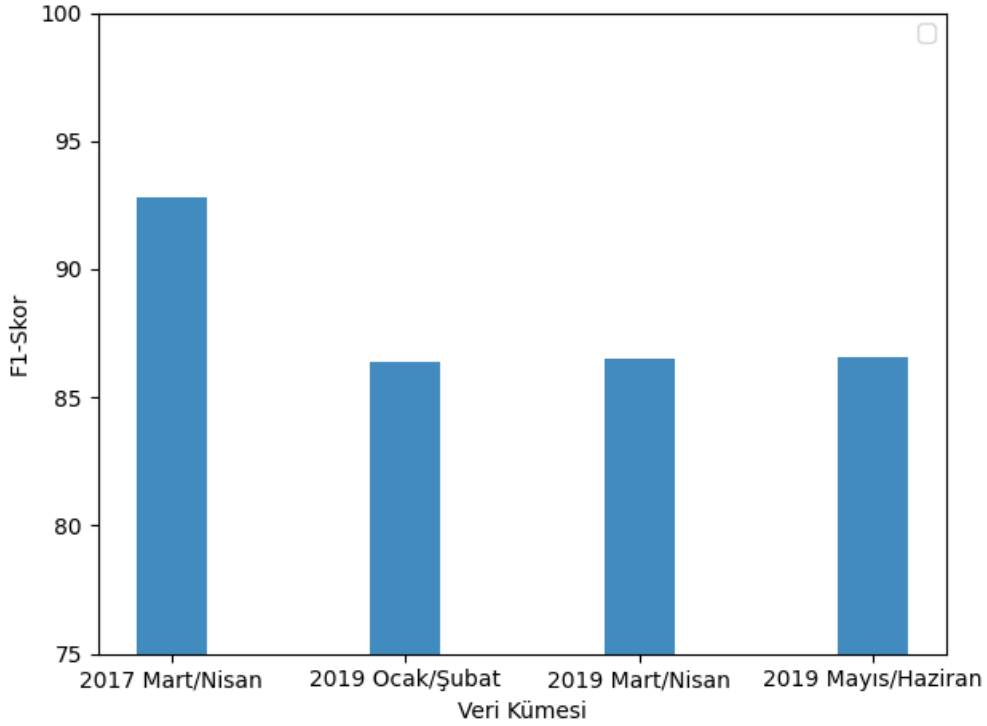
Çalışmada rastgele orman yöntemi kullanılarak işlemlerin sahte olma skorunun elde edilmesi hedeflenmiştir. Bu sebeple rastgele orman ile işlemlerin sahte olma skoru elde edilmiştir. İşlemleri sahte veya yasal olarak sınıflandırabilmek için bir eşik değeri belirlenerek, o eşik değerinin üzerinde olan işlemler sahte olarak sınıflandırılmıştır. Çeşitli eşik değerleri ile yapılan deneyler sonucunda en başarılı sonuç 0.5 değeri ile elde edilmiştir. Yapılan deneylerin sonuçlarına Şekil 4.2'de yer verilmiştir.



Şekil 4.2 Farklı eşik değerlerine göre rastgele orman ile sınıflandırma sonuçları

4.3 Veri Kümesi ve Sahte İşlem Sayısı

Çalışmada farklı aylara ait örnekler için sınıflandırma sonuçlarını karşılaştırabilmek amacıyla, dört farklı veri kümesi kullanılarak deneyler yapılmıştır. Şekil 4.3 ve Tablo 4.2'de bu dört veri kümesi ile yapılan sınıflandırma sonuçları verilmiştir. Sınıflandırma işlemi rastgele orman yöntemi kullanılarak gerçekleştirilmiştir.



Şekil 4.3 Veri kümelerinin karşılaştırılması

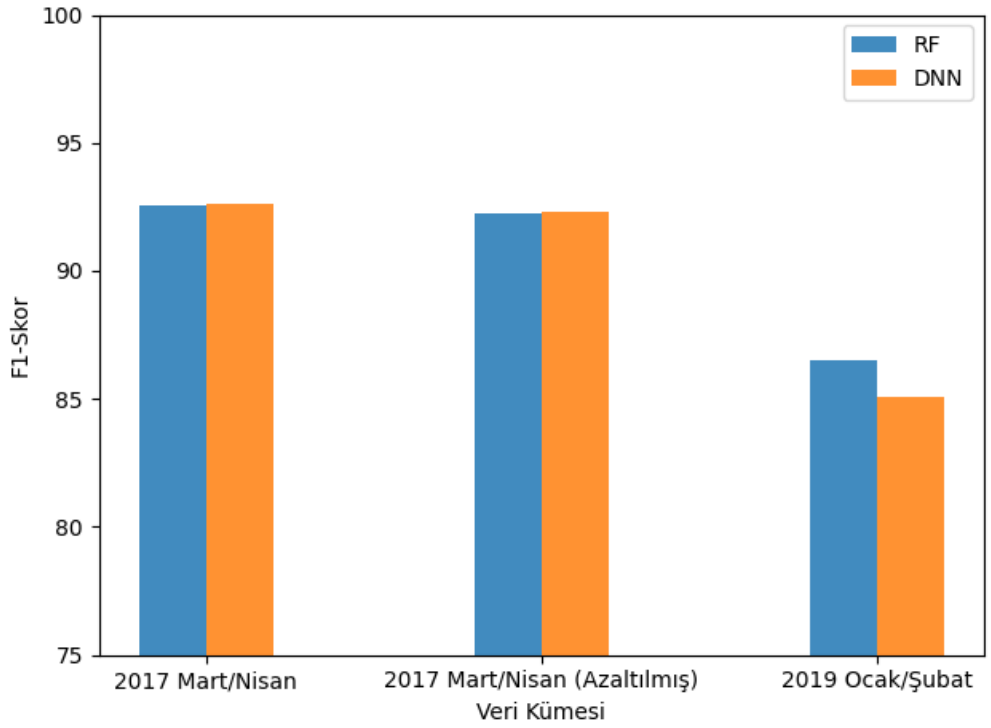
Tablo 4.2 Veri kümelerinin karşılaştırılması

Veri Kümesi	Kesinlik		Duyarlılık		F1-Skor	
	Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
2017 Mart/Nisan	**98.26%**	**93.75%**	**98.78%**	**91.28%**	**98.52%**	**92.50%**
2019 Ocak/Şubat	97.00%	88.20%	97.73%	84.88%	97.36%	86.51%
2019 Mart/Nisan	97.05%	87.49%	97.56%	85.18%	97.30%	86.32%
2019 Mayıs/Haziran	97.28%	89.05%	97.87%	86.31%	97.57%	87.66%

Şekilde görüldüğü gibi daha fazla sayıda işlemin olduğu 2017 veri kümesi daha yüksek bir başarı elde etmiştir. 2017 yılı Mart/Nisan ayına ait örnekler ile %92.50 başarı elde edilirken, 2019 yılı Ocak/Şubat aylarında ise %86.51'lik bir başarı elde edilmiştir. 2019 yılının diğer ayları ile yapılan deneylerde, sınıflandırma işlemi birbirine çok yakın sonuçlar vermiştir.

2017 yılına ait fazla sayıda örnek bulunan veri kümesi ile yapılan sınıflandırma, 2019 Ocak/Şubat veri kümesi ile yapılana göre daha başarılı sonuç vermiştir. Bu başarının

sebebinin veri kümesinde fazla örnek bulunmasından kaynaklı olup olmadığının değerlendirilmesi amacıyla yeni bir deney gerçekleştirilmiştir. 2017 yılı veri kümesi, 2019 Ocak/Şubat veri kümesinde bulunan sahte ve yasal işlemlerle aynı sayıda olacak şekilde rasgele örnekler seçilerek azaltılmıştır. Yapılan deneyin sonucu Şekil 4.4 ve Tablo 4.3'de görülmektedir. 2017 yılındaki veri kümesi azaltılarak yapılan deneyler sonucunda sınıflandırma başarısının değişmediği, dolayısıyla 2017 yılındaki başarının yüksek olmasının sebebinin örnek sayısının fazla olmasından kaynaklı olmadığı görülmüştür.



Şekil 4.4 Sahte işlem sayısına göre sınıflandırma başarıları

Tablo 4.3 Sahte işlem sayısına göre sınıflandırma başarıları

Veri Kümesi	Sınıflandırıcı	Kesinlik		Duyarlılık		F1-Skor	
		Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
2017 Mart/Nisan	RF	98.26%	93.86%	98.80%	91.26%	98.53%	92.54%
2017 Mart/Nisan	**DNN**	**98.36%**	**93.49%**	**98.72%**	**91.79%**	**98.54%**	**92.64%**
2017 Mart/Nisan (Azaltılmış)	RF	98.27%	93.12%	98.65%	91.36%	98.46%	92.23%
2017 Mart/Nisan (Azaltılmış)	DNN	98.34%	93.00%	98.62%	91.67%	98.48%	92.33%
2019 Ocak/Şubat	RF	97.00%	88.20%	97.73%	84.88%	97.36%	86.51%
2019 Ocak/Şubat	DNN	96.75%	86.54%	97.39%	83.66%	97.07%	85.07%

Verinin büyüklüğünden ötürü dört farklı veri kümesi ile tüm deneyleri gerçekleştirmek çok fazla zaman aldığı için, 2019 yılının farklı aylarına ait üç adet veri kümesi ile yapılan deneylerde çok yakın sonuçlar alınması da göz önünde bulundurularak, kalan deneyler 2019 Ocak/Şubat veri kümesi ile yapılmıştır.

4.4 Özellik Türetme ve Özellik Seçimi

İncelenen çalışmalarda yeni özellik türetiminin ve bu özellikler arasından en başarılı olanların seçilmesinin sınıflandırma başarısı için oldukça önemli olduğu tespit edilmiştir. Bu amaçla mevcut veri kümesinde bulunan 28 adet özellik, yeni özellikler türetilerek 127'ye çıkarılmıştır.

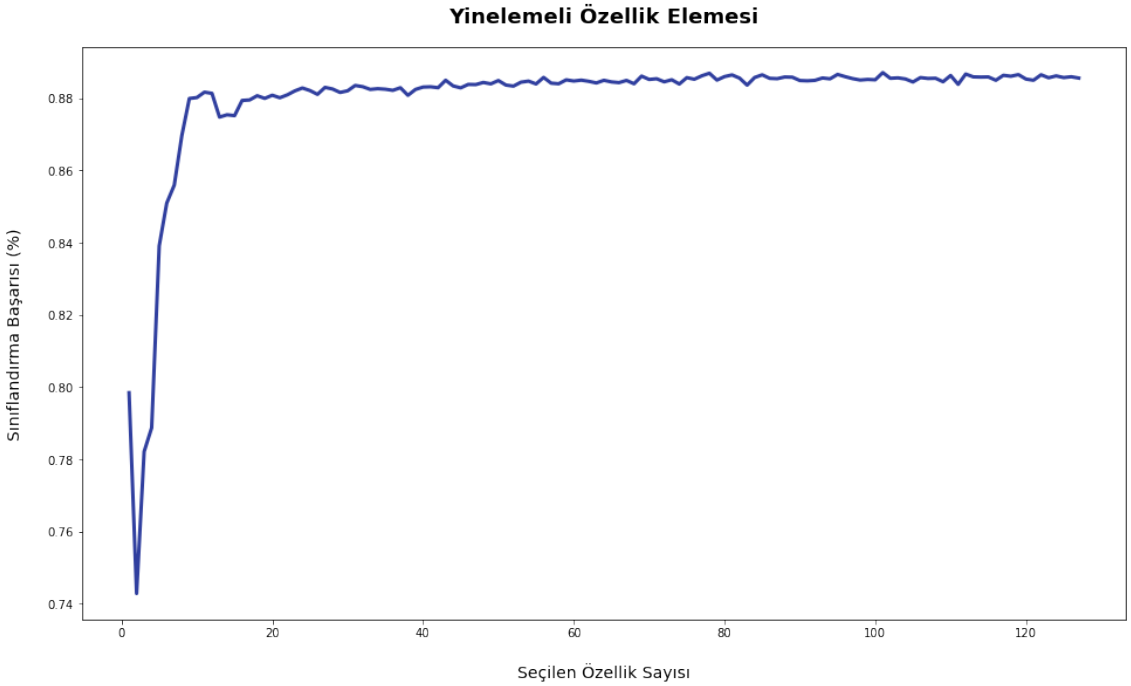
Mevcut özelliklerle birlikte oluşturulan yeni özellikler, RFE yöntemi kullanılarak, sınıflandırmaya olan etkilerine göre sıralanmıştır. Yapılan incelemeler sonucunda, yeni türetilen "Güvenlik Seviyesi (3D Secure)" özelliğinin, işlemleri sınıflandırmada en etkili özellik olduğu tespit edilmiştir. Tablo 4.4'de seçilen özellikler arasından en başarılı olan 10 tanesi görülmektedir. En önemli olduğu tespit edilen 10 özellikten beş tanesinin yeni türetilen özellikler olması, özellik türetme işleminin başarılı olduğunu göstermektedir.

Tablo 4.4 Özellik seçimi sonucunda en başarılı olarak belirlenen 10 özellik

Sıra	Özellik	Özellik Türü	Eklenme Şekli
1	Güvenlik Seviyesi	Kategorik	Türetilmiş
2	İşyeri Kategorisi	Nümerik	Mevcut
3	Aylık İşyeri Kategorisi Sayısı	Nümerik	Türetilmiş
4	Ödeme Yöntemi	Kategorik	Mevcut
5	Kaynak Sistem	Kategorik	Mevcut
6	Tutar	Nümerik	Mevcut
7	Kart Sahibi Banka	Kategorik	Mevcut
8	Aylık İşyeri Kategorisi Tutarı	Nümerik	Türetilmiş
9	Aylık Toplam İşlem Sayısı	Nümerik	Türetilmiş
10	Günlük E-Ticaret Tipi Tutarı	Nümerik	Türetilmiş

RFE kullanılarak sıralanan başarılı özelliklerden 5, 10, 15, 20, 25, 30, 40, 60, 80 ve 100 tanesi kullanılarak sınıflandırma yapılmış ve başarıları ölçülmüştür. Şekil 4.5'de yapılan sınıflandırma sonuçları görülmektedir. Şekil incelendiğinde 10 adet özellik ile sınıflandırma başarısının ciddi oranda yükseldiği görülse de en başarılı sonuçlar 30 özellik ile alınmıştır. Bu sebeple çalışmanın ilerleyen kısımlarında kullanılmak üzere bu 30 adet özellik seçilmiştir.

Özellik türetme ve özellik seçiminin başarılarını görmek amacıyla hem 2017, hem de 2019 yılına ait örneklerle sınıflandırma yapılmıştır. Şekil 4.6 ve Tablo 4.5'de yapılan sınıflandırma sonuçlarına yer verilmiştir. Mavi renk (Mevcut Özellikler), veri kümesinde bulunan özelliklere herhangi bir işlem yapılmadan, mevcutta bulunan 28 adet özelliğin one-hot kodlanarak 481'e arttırılması ile elde edilen sonuçları göstermektedir. Turuncu renk (Özellik Seçimi - Mevcut), mevcutta bulunan 28 özellikten en başarılı 10 tanesi seçildikten sonra, one-hot kodlama kullanılarak elde



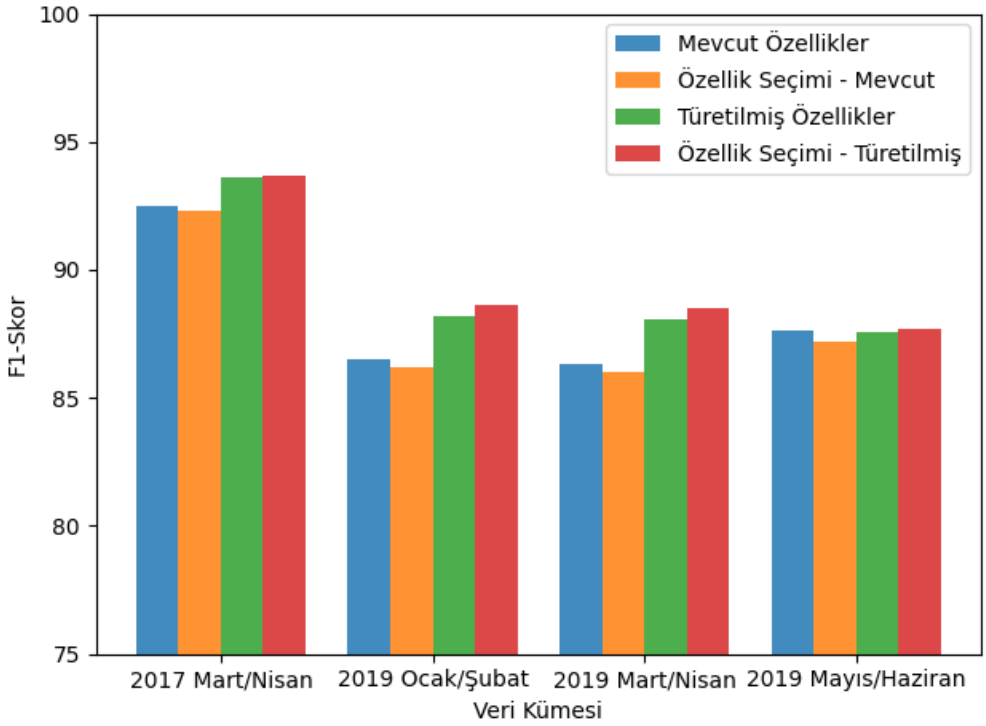
Şekil 4.5 Özellik sayısına göre sınıflandırma başarıları

edilen 395 özellik ile yapılan testlerin sonuçlarını temsil etmektedir.

Yeşil renk (Türetilmiş Özellikler) ile, özellik türetimi sonucu oluşturulan 127 adet özelliğin, one-hot kodlama kullanılarak 610'a çıkarılması ile yapılan deney sonuçları gösterilmiştir. Kırmızı renk (Özellik Seçimi - Türetilmiş) ise, özellik türetme sonucu oluşan 127 özellik arasından seçilen 30 adet özelliğin one-hot kodlama ile 461'e çıkarılması ile yapılan sınıflandırma sonuçlarını temsil etmektedir. One-hot kodlama sonucu oluşan özellik sayıları 2019 Ocak/Şubat ayına ait örneklere göre verilmiş olup, diğer aylarda farklılık gösterebilmektedir.

Tablo 4.5 Özellik seçimi ve özellik türetme işlemleri sonucu sınıflandırma başarıları

Veri Kümesi	Özellik Durumu	Kesinlik		Duyarlılık		F1-Skor	
		Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
2017 Mart/Nisan	Mevcut Özellikler	98.26%	93.75%	98.78%	91.28%	98.52%	92.50%
2017 Mart/Nisan	Özellik Seçimi - Mevcut	98.23%	93.50%	98.73%	91.11%	98.48%	92.29%
2017 Mart/Nisan	Türetilmiş Özellikler	98.53%	94.64%	98.95%	92.64%	98.74%	93.63%
2017 Mart/Nisan	**Özellik Seçimi - Türetilmiş**	**98.55%**	**94.60%**	**98.94%**	**92.73%**	**98.74%**	**93.65%**
2019 Ocak/Şubat	Mevcut Özellikler	97.00%	88.20%	97.73%	84.88%	97.36%	86.51%
2019 Ocak/Şubat	Özellik Seçimi - Mevcut	96.91%	88.00%	97.69%	84.45%	97.30%	86.19%
2019 Ocak/Şubat	Türetilmiş Özellikler	97.36%	89.69%	98.00%	86.73%	97.68%	88.19%
2019 Ocak/Şubat	**Özellik Seçimi - Türetilmiş**	**97.48%**	**89.97%**	**98.05%**	**97.35%**	**97.76%**	**88.64%**
2019 Mart/Nisan	Mevcut Özellikler	97.05%	87.49%	97.56%	85.18%	97.30%	86.32%
2019 Mart/Nisan	Özellik Seçimi - Mevcut	97.04%	87.16%	97.49%	85.17%	97.26%	86.15%
2019 Mart/Nisan	Türetilmiş Özellikler	97.65%	87.91%	97.57%	88.27%	97.61%	88.09%
2019 Mart/Nisan	**Özellik Seçimi - Türetilmiş**	**97.70%**	**88.45%**	**97.68%**	**88.52%**	**97.69%**	**88.48%**
2019 Mayıs/Haziran	Mevcut Özellikler	97.28%	89.05%	97.87%	86.31%	97.57%	87.66%
2019 Mayıs/Haziran	Özellik Seçimi - Mevcut	97.12%	88.97%	97.88%	85.51%	97.50%	87.21%
2019 Mayıs/Haziran	Türetilmiş Özellikler	97.15%	89.64%	98.02%	85.63%	97.58%	87.59%
2019 Mayıs/Haziran	**Özellik Seçimi - Türetilmiş**	**97.27%**	**89.12%**	**97.89%**	**86.28%**	**97.58%**	**87.68%**



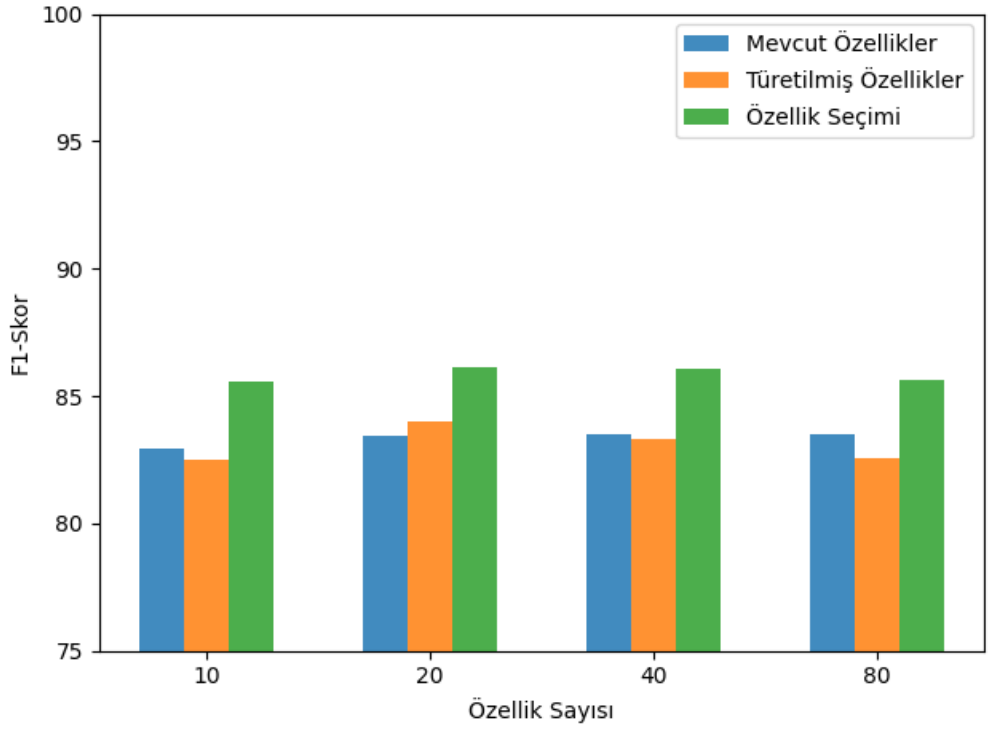
Şekil 4.6 Özellik seçimi ve özellik türetme işlemleri sonucu sınıflandırma başarıları

2019 Ocak/Şubat ayları ile yapılan deney sonuçlarına göre mevcut özellikler ile %86.51 başarı elde edilirken, mevcut özellikler arasından yapılan özellik seçimi sonucu başarı %86.19'e düşmüştür. Mevcut özelliklere yeni türetilmiş özellikler eklendiğinde %88.19 başarı elde edilmiştir. Tüm özelliklere uygulanan özellik seçimi sonucunda başarı %88.64'e yükselmiştir. Tüm aylarla yapılan deneylerde, özellik türetimi ardından uygulanan özellik seçimi sonucunda başarının yükseldiği görülmüştür.

4.5 Özkodlayıcılar

Bir çok çalışmada özkodlayıcılar kullanılarak özellik çıkarmanın sınıflandırma başarısını arttırdığı öne sürülmüştür. Özkodlayıcıların etkisinin gözlemlenmesi amacıyla farklı sayılarda özellik çıkarılarak deneyler yapılmıştır. Yapılan deneylerde 2019 Ocak/Şubat aylarına ait veri kümesi kullanılmıştır. Şekil 4.7 ve Tablo 4.6'de özkodlayıcılar yardımıyla çıkarılan özellik sayılarına göre sınıflandırma başarılarının değişimine yer verilmiştir. Özkodlayıcı ile özellik çıkarımı yapılmadan önce veri kümesi one-hot kodlama ile kodlanmıştır.

Alınan sonuçlara göre özkodlayıcı kullanılarak özellik çıkarıldığında sınıflandırma başarısı düşmüştür. Mevcut özellikler kullanılarak %86.19 başarıyla sınıflandırılan veri kümesi, özkodlayıcılarla 20 adet özellik çıkarıldığında %83.46'e düşmüştür.



Şekil 4.7 Özkodlayıcılar ile çıkarılan özellik sayısına bağlı olarak sınıflandırma başarıları

Tablo 4.6 Özkodlayıcılar ile çıkarılan özellik sayısına bağlı olarak sınıflandırma başarıları

Özellik Sayısı	Özellik Durumu	Kesinlik		Duyarlılık		F1-Skor	
		Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
10	Mevcut Özellikler	95.89%	87.16%	97.66%	79.11%	96.77%	82.94%
10	Türetilmiş Özellikler	95.95%	85.75%	97.36%	79.45%	96.65%	82.48%
10	Özellik Seçimi	96.59%	88.55%	97.85%	82.77%	97.22%	85.56%
20	Mevcut Özellikler	96.05%	87.33%	97.68%	79.91%	96.85%	83.46%
20	Türetilmiş Özellikler	96.30%	86.93%	97.55%	81.28%	96.92%	84.01%
20	**Özellik Seçimi**	**96.70%**	**89.12%**	**97.96%**	**83.33%**	**97.33%**	**86.13%**
40	Mevcut Özellikler	96.05%	87.39%	97.69%	79.93%	96.86%	83.49%
40	Türetilmiş Özellikler	96.14%	86.38%	97.46%	80.45%	96.79%	83.31%
40	Özellik Seçimi	96.69%	88.98%	97.93%	83.28%	97.31%	86.04%
80	Mevcut Özellikler	96.08%	87.29%	97.66%	80.08%	96.86%	83.53%
80	Türetilmiş Özellikler	95.94%	86.01%	97.41%	79.39%	96.67%	82.56%
80	Özellik Seçimi	96.56%	88.84%	97.92%	82.58%	97.24%	85.60%

Ancak özellik türetme ve özellik seçimi uygulandıktan sonra, özkodlayıcılar ile özellik çıkarıldığında, özkodlayıcının başarısının %86.13'e yükseldiği tespit edilmiştir. Her ne kadar kendi içerisinde başarı artsa da, yine de özkodlayıcı kullanılmadan sınıflandırma yapılması daha başarılı olmuştur.

Özkodlayıcıların başarısının özellik sayısına göre değişimini görebilmek amacıyla, 10,

20, 40 ve 80 adet özellik ile sınıflandırma yapılmıştır. Bunlar arasından en başarılı olanı 20 adet özellik ile olmuştur. Sentetik veri üretmek için ihtiyaç duyulan veri kümesi, özkodlayıcı yardımıyla çıkarılan bu 20 adet özellik ile oluşturulmuştur.

4.6 Sentetik Veri Üretimi

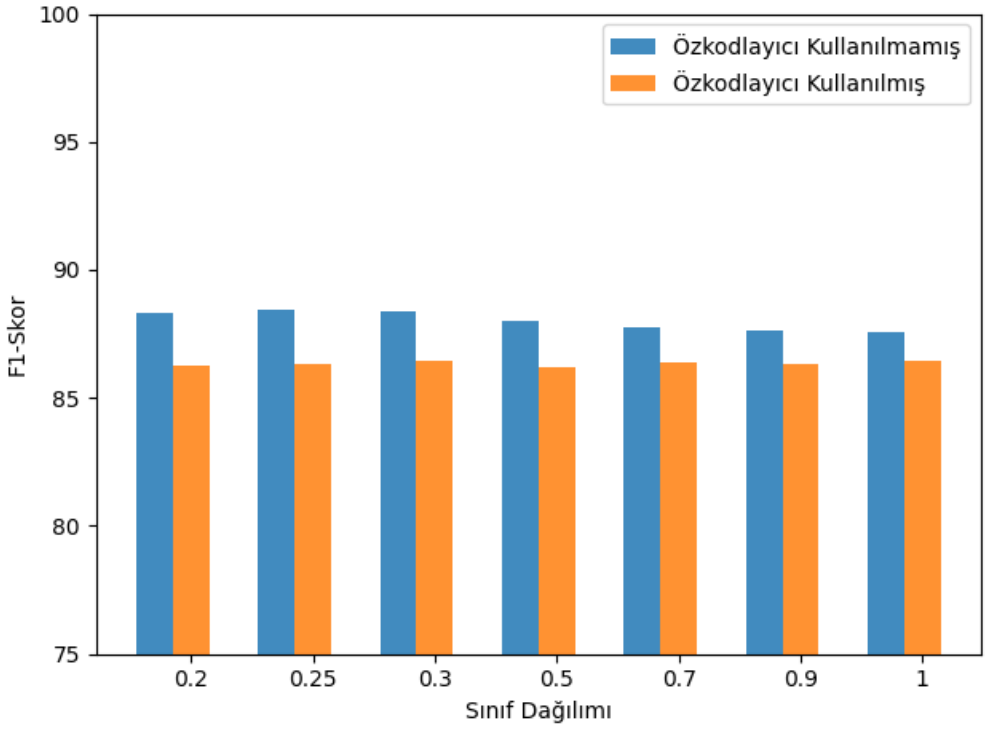
Dengesiz veri kümeleriyle gerçekleştirilen bir çok çalışmada, üstkümeleme ve sentetik veri üretiminin sınıflandırma başarısını arttırdığı ortaya konmuştur. Bu sebeple bu tez çalışmasında veri kümesinde bulunan örnekler kullanılarak, sentetik veri üretilip, başarıya olan etkisi incelenmiştir. Sentetik veri üretimi için SMOTE yöntemi ve farklı GAN yöntemleri ile deneyler yapılmıştır.

4.6.1 Üst Kümeleme ile Sentetik Veri Üretimi

SMOTE yöntemi, azınlık olan sınıfa ait örnekleri üst-örnekleme ile çoğunluk olan sınıfa yaklaştırmaktadır. SMOTE ile farklı oranlarda sahte örnekler arttırılmış ve yapılan sınıflandırma işlemi sonuçları Şekil 4.8 ve Tablo 4.7'de gösterilmiştir. Yapılan deneylerde yasal işlem sayısı sabit tutularak, yalnızca sentetik sahte işlemler üretilmiştir. Şekilde görülen 0.2 oranı veri kümesinin orijinal dağılımını temsil etmektedir. Şekilde mavi renk ile temsil edilen sonuçlar, özkodlayıcı kullanılmadan özellik seçimi sonucu oluşan veri kümesi ile yapılan deneyleri, turuncu renk ile temsil edilen sonuçlar ise, özellik seçimi yapıldıktan sonra özkodlayıcı ile özellik çıkarılarak oluşturulan veri kümesi ile yapılan deneyleri göstermektedir. Sınıflandırma başarısını ölçmek amacıyla rastgele orman yöntemi kullanılmıştır.

Tablo 4.7 SMOTE ile farklı oranlarda üst kümeleme başarıları

Sınıf Dağılımı	Özkodlayıcı	Kesinlik		Duyarlılık		F1-Skor	
		Yasal	Sahte	Yasal	Sahte	Yasal	Sahte
0.2	Yok	97.32%	90.14%	98.10%	86.54%	97.71%	88.30%
0.25	**Yok**	**97.46%**	**89.47%**	**97.94%**	**87.25%**	**97.70%**	**88.35%**
0.3	Yok	97.56%	88.86%	97.79%	87.81%	97.68%	88.33%
0.5	Yok	97.66%	87.98%	97.58%	88.32%	97.62%	88.15%
0.7	Yok	97.72%	87.24%	97.40%	88.67%	97.56%	87.95%
0.9	Yok	97.79%	86.61%	97.24%	89.04%	97.52%	87.81%
1	Yok	97.90%	85.78%	97.02%	89.63%	97.46%	87.66%
0.2	Var	96.79%	88.84%	97.89%	83.76%	97.34%	86.23%
0.25	**Var**	**96.88%**	**88.43%**	**97.79%**	**84.26%**	**97.33%**	**86.30%**
0.3	Var	96.93%	88.23%	97.74%	84.56%	97.34%	86.36%
0.5	Var	97.04%	87.47%	97.56%	85.14%	97.30%	86.29%
0.7	Var	97.10%	86.87%	97.41%	85.48%	97.26%	86.17%
0.9	Var	97.29%	86.49%	97.29%	86.45%	97.29%	86.47%
1	Var	97.41%	85.77%	97.11%	87.10%	97.26%	86.43%



Şekil 4.8 SMOTE ile farklı oranlarda üst kümeleme başarıları

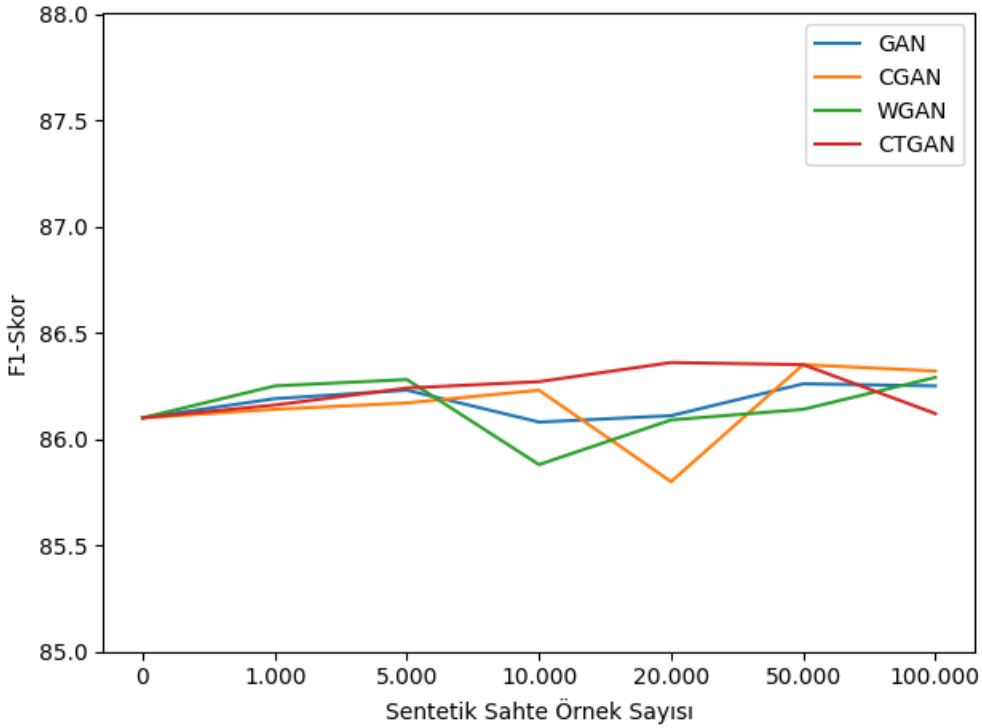
Üst kümeleme yapılmadan %88.30 başarı ile sınıflandırılan örnekler, SMOTE ile sentetik veri üreterek 0.5 oranında %88.15, 0.7 oranında %87.95 ve 1 oranında %87.66 başarı ile sınıflandırılmıştır. Yani üst kümeleme yapmak sınıflandırma başarısını küçük bir oranda düşürmektedir. Bu sonuç, önceki çalışmalarda yapılan deneyler sonucu belirlenen, 0.2 oranının doğru bir şekilde seçildiğini göstermektedir. Özkodlayıcı ile özellik çıkarılarak yapılan sınıflandırma işlemlerinde de başarının artmadığı görülmüştür.

4.6.2 Büyük Veri Kümelerinde Sentetik Veri Üretimi

SMOTE, yeni örnekler oluştururken KNN yöntemini kullanmaktadır. Herhangi bir öğrenme süreci uygulanmamaktadır. Bu sebeple üretilen örneklerin benzerliğinin yeterli olmama ihtimalini değerlendirmek için GAN ile sentetik örnekler üretilmiştir. Bu çalışmada kullanılan tablo veri kümesine uygun GAN modelleri araştırılmış ve bunlardan GAN, CGAN, WGAN ve CTGAN ile deneyler yürütülmüştür. CTGAN dışındaki yöntemler kategorik özelliklerle eğitilemediği için bu yöntemler özkodlayıcı ile çıkarılmış özelliklerden oluşan veri kümeleri ile eğitilmiştir.

GAN, CGAN ve WGAN modelleri, 2019 Ocak/Şubat veri kümesiyle eğitilerek, sentetik sahte işlemler üretilmiştir. Veri kümesinde 2019 Ocak ayına ait 93.620 adet yasal işlem, 18.724 adet sahte işlem bulunmaktadır. Bu işlemlere ek olarak farklı sayılarda

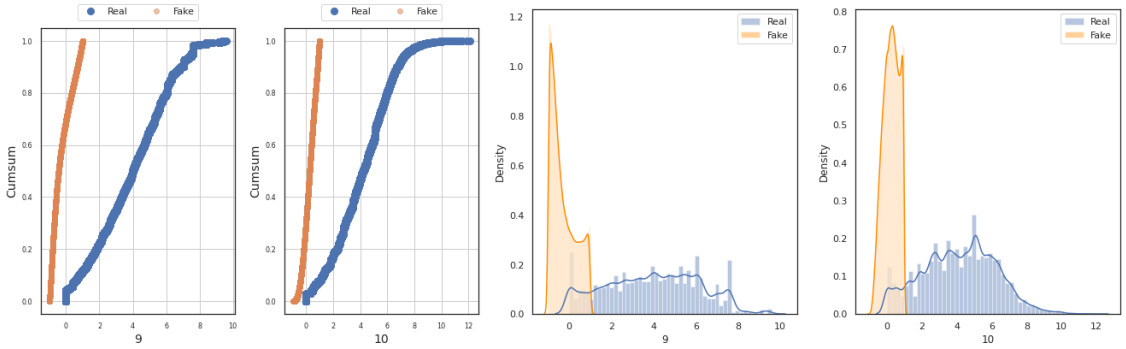
üretilen sahte işlemler, veri kümesine eklenerek rastgele orman modelini eğitmek için kullanılmıştır. Böylelikle sahte işlem sayısını arttırmanın sınıflandırmaya olan etkisi incelenmiştir. Şekil 4.9'da görüldüğü gibi, sınıflandırma başarısında az da olsa bir artış olmuştur.



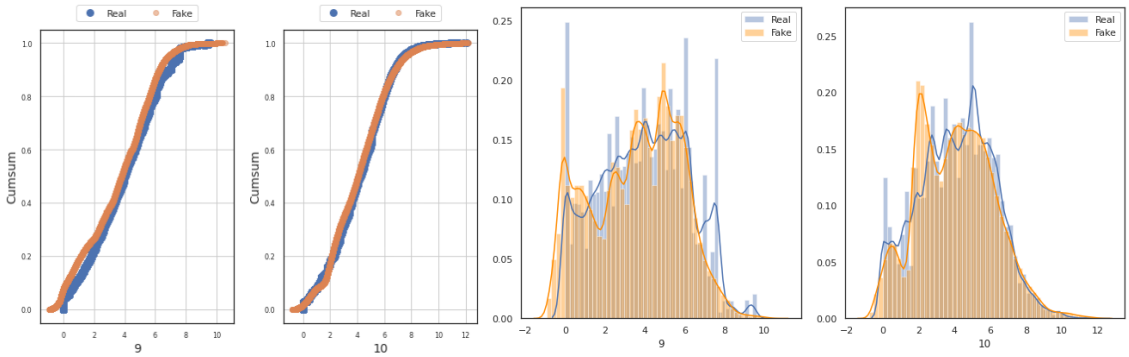
Şekil 4.9 Sentetik sahte işlem sayısına göre sınıflandırma başarıları

GAN, CGAN, WGAN ile yapılan deneyler sonucunda, CTGAN ve GAN yöntemlerinin diğerlerine göre daha başarılı olduğu tespit edilmiştir. Şekil 4.10 ve Şekil 4.11'de GAN ve CTGAN yöntemleri ile oluşturulmuş sentetik örneklerin, gerçeklerine olan benzerliğinin anlaşılması amacıyla oluşturulmuş grafikler görülmektedir. Turuncu ile sentetik, mavi ile gerçek örneklerin dağılımları gösterilmiştir. Özkodlayıcılar sonucu oluşturulan 9. ve 10. özelliklerin toplamalı ve ayrık olarak dağılımlarına yer verilmiştir. Her iki şekil incelendiğinde CTGAN ile üretilen örneklerin gerçeğine çok daha yakın olduğu görülmektedir.

CTGAN'ın sentetik örnekler üretmede daha başarılı olduğu tespit edildikten sonra, bu yöntemle yapılan deneyler çeşitlendirilmiştir. CTGAN kullanılarak, sahte işlemlerle birlikte yasal işlemler de üretilmiştir. Böylelikle sahte/yasal oranı bozulmadan her iki sınıfa ait örnek sayısının arttırılmasının sınıflandırmaya etkisi gözlemlenmiştir. Şekil 4.12'de CTGAN ile yapılan deney sonuçlarına yer verilmiştir. Sonuçlar incelendiğinde rastgele orman yöntemi ile orijinal veri kümesi %88.21 başarı ile sınıflandırılırken, yeni örnekler eklendiğinde %88.75 başarı ile sınıflandırılmıştır. Sahte işlemlerle birlikte yasal işlemlerin eklendiği deneylerde sınıflandırma başarısının daha çok arttığı



Şekil 4.10 GAN ile üretilen işlemlere ait iki özelliğin gerçek örneklerle karşılaştırılması



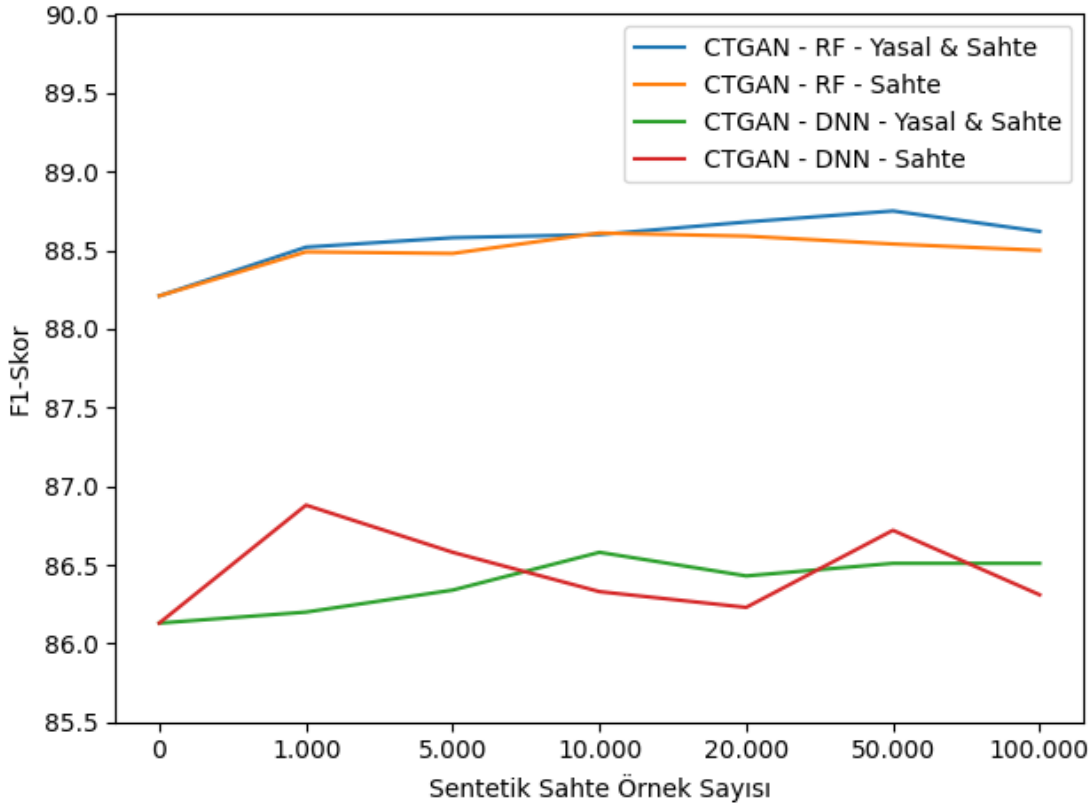
Şekil 4.11 CTGAN ile üretilen işlemlere ait iki özelliğin gerçek örneklerle karşılaştırılması

görülmüştür.

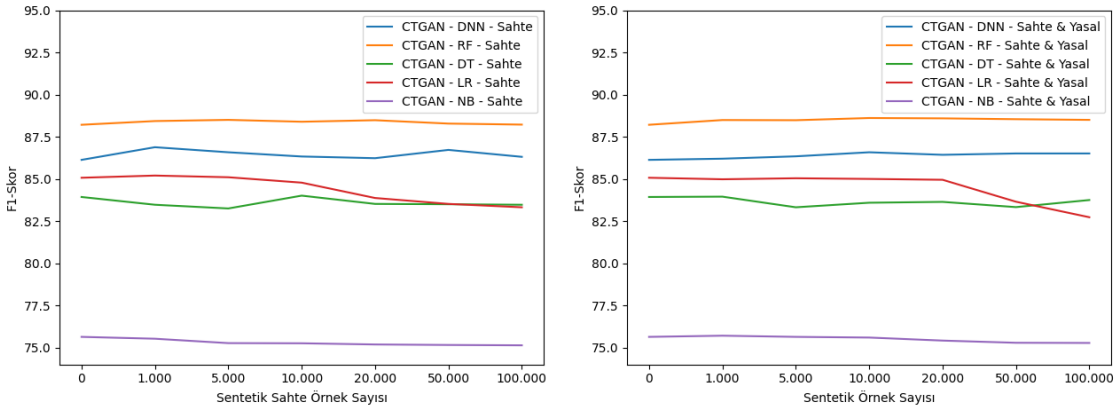
Farklı sınıflandırıcılarla CTGAN başarısının ölçülebilmesi amacıyla, naive bayes (NB), karar ağaçları (DT), lojistik regrasyon (LR), rastgele orman (RF) ve derin sinir ağları (DNN) kullanılarak deneyler gerçekleştirilmiştir. Şekil 4.13'de yalnızca sahte sentetik örneklerle yapılan deney sonuçları (solda) ve hem sahte hem de yasal sentetik örnekler ile yapılan deney sonuçları (sağda) görülmektedir. Alınan sonuçlara göre RF, DNN ve DT ile yeni eklenen sentetik örnekler sınıflandırma başarısını arttırmaktadır. LR ve NB ile yapılan deneylerde ise yeni örnekler eklemek sınıflandırma başarısını düşürmektedir.

4.6.3 Küçük Veri Kümelerinde Sentetik Veri Üretimi

Kredi kartı sahtekarlığı çalışmalarında kullanılan veri kümeleri genellikle az sayıda işlemden oluşmaktadır. İncelenen çalışmalarda en çok kullanılan Avrupa veri kümesinde 500 adet sahte işlem bulunmaktadır. Az sayıda örnekle üretilen sentetik verilerin başarıya olan etkisini ölçümlemek amacıyla, küçük bir veri kümesi oluşturulmuştur. 500 adet sahte ve 2500 adet yasal işlem rastgele seçilerek oluşturulan veri kümesi ile CTGAN modeli eğitilerek sentetik örnekler üretilmiştir. Üretilen



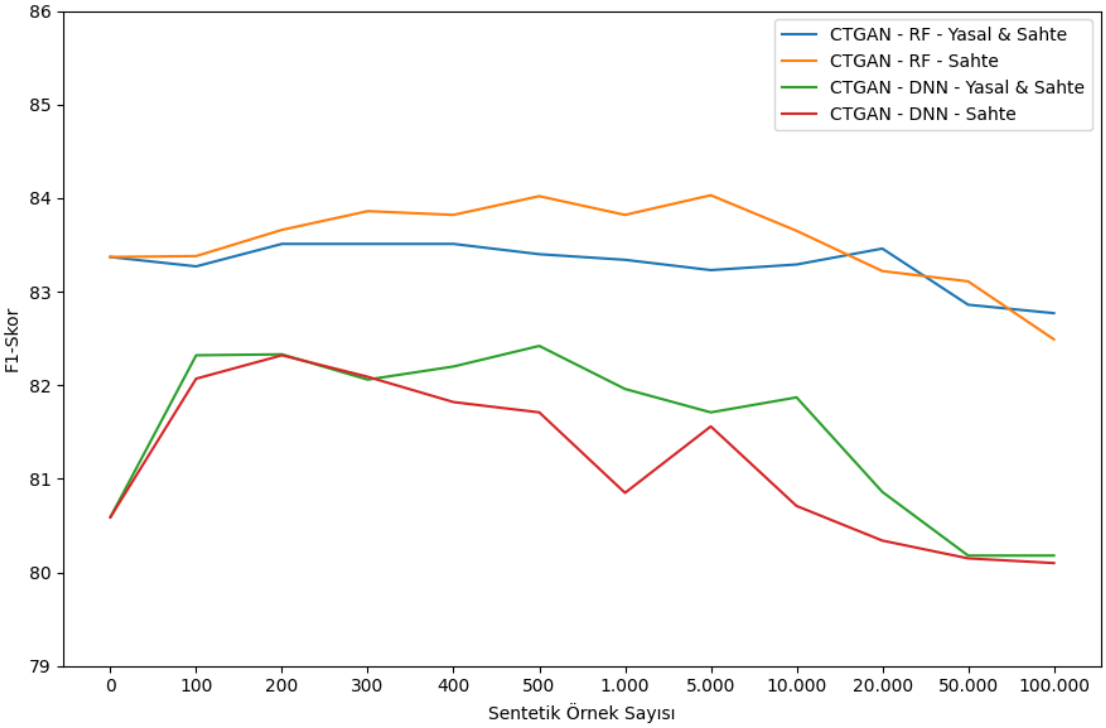
Şekil 4.12 CTGAN kullanılarak üretilen sahte ve yasal işlemlere ait sınıflandırma başarıları



Şekil 4.13 CTGAN kullanılarak üretilen, yalnızca sahte sentetik örneklerle yapılan deney sonuçları (solda) ve hem sahte hem de yasal sentetik örnekler ile yapılan deney sonuçları (sağda)

sentetik örnekler, az sayıda örneğe eklenerek sınıflandırma başarıları ölçülmüştür. Şekil 4.14'de yapılan sınıflandırma işlemlerine ait sonuçlara yer verilmiştir.

Azaltılmış veri kümesi ile sentetik örnek eklemeden yapılan sınıflandırma işleminde rasgele orman ile %83.37, derin öğrenme ile %80.59 başarı elde edilmiştir. 200 adet sentetik sahte örnek üretildiğinde derin öğrenme başarısı %82.33'e yükselmiştir.



Şekil 4.14 Azaltılmış veri kümesiyle yapılan CTGAN sonuçları

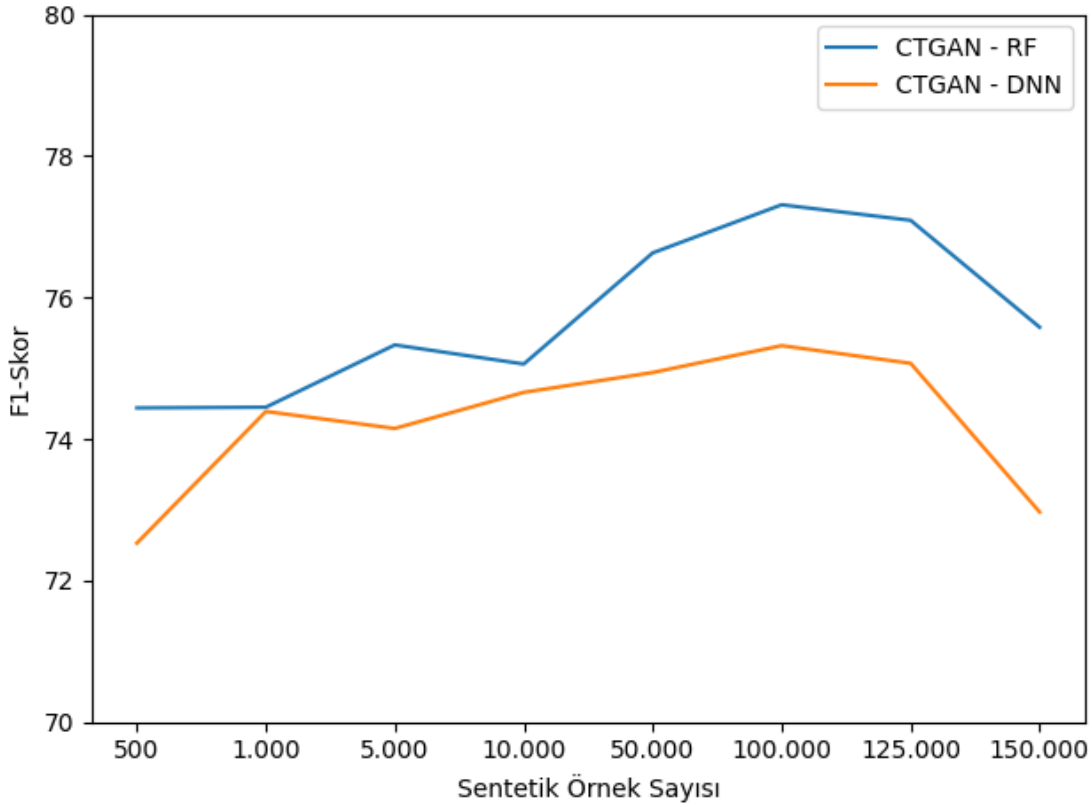
Rastgele orman ile yapılan sınıflandırmada 500 adet sentetik örnek üretilerek %84.02 başarı elde edilmiştir. Üretilen sentetik örnek sayısının 500'den fazla olduğu deneylerde başarı düşmeye başlamıştır. Az sayıda örnekten oluşan veri kümelerinin GAN ile sentetik örnekler üreterek sınıflandırma başarısını arttırabileceği tespit edilmiştir.

4.6.4 Yalnızca Sentetik Veri ile Eğitim

Üretilen örneklerin gerçekten başarılı bir şekilde üretildiklerini ölçmek amacıyla RF ve DNN sınıflandırıcıları, gerçek veri kümesi kullanılmadan, yalnızca sentetik veri örnekleri ile eğitilerek deneyler gerçekleştirilmiştir. Şekil 4.15'de bu deneylere ait sonuçlara yer verilmiştir. Az sayıda örnekle yapılan deneylerde dahi, yüksek bir başarı elde etmiştir. Hem RF hem de DNN'de ile yapılan deneylerde kullanılan örnek sayısı arttıkça sınıflandırma başarısı da artmıştır. Yalnızca sentetik işlemlerle yapılan bu deneylerin sonuçları, üretilen sentetik örneklerin gerçeklerine benzer olduklarını göstermektedir.

4.6.5 Avrupa Veri Kümesi ile Elde Edilen Sonuçlar

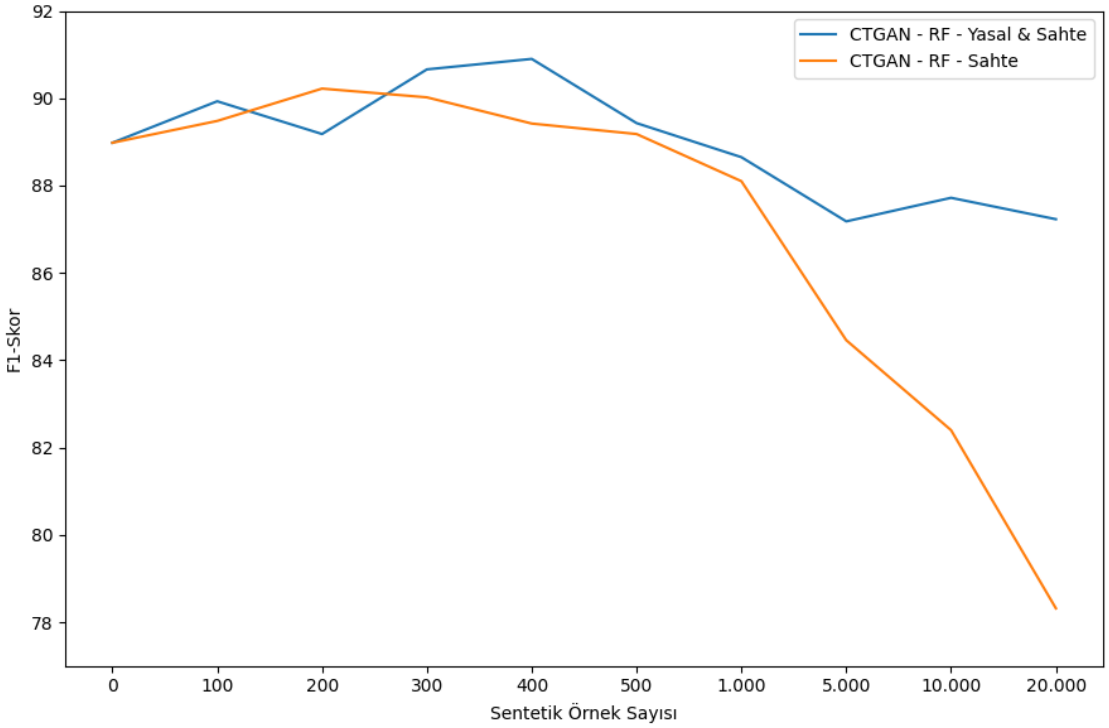
İncelenen çalışmalarda sıklıkla kullanılan Avrupa veri kümesi [3] ile bu çalışmada önerilen yöntemler test edilmiştir. Avrupa veri kümesinde 492 adet sahte, 284.807



Şekil 4.15 Yalnızca sentetik işlem örnekleri kullanılarak yapılan sınıflandırma sonuçları

yasal işlem bulunmaktadır. Çalışmaya uygun olması açısından sahte işlemlerin 5 katı olacak şekilde yasal işlem rastgele seçilerek alt kümeleme yapılmıştır. Elde edilen işlemler eğitim ve test olarak aynı sayıda iki kümeye bölünmüştür. Son olarak elde edilen eğitim ve test kümelerinde 246 sahte, 1230 yasal işlem bulunmaktadır.

Veri kümesinde temel bileşen analizi (PCA) ile elde edilmiş 30 adet özellik bulunmaktadır. Bu özelliklerden biri olan işlem tutarı özelliğine min-max normalizasyonu uygulanmıştır. CTGAN kullanılarak farklı sayılarda elde edilen sahte ve yasal işlemler ile sınıflandırma deneyleri gerçekleştirilmiştir. Rastgele orman sınıflandırıcısıyla gerçekleştirilen deneylerin sonuçlarına Şekil 4.16'de yer verilmiştir. Sentetik veri üretilmeden yapılan sınıflandırma işleminde %88.98 başarı elde edilirken, 500 adet yasal ve sahte işlem üretildiğinde sınıflandırma başarısının %90.90 olduğu gözlemlenmiştir. Avrupa veri kümesiyle yapılan deneyler, bu çalışmada kullanılan küçük veri kümesiyle yapılan deneylere yakın sonuçlar göstermiştir.



Şekil 4.16 Avrupa veri kümesi kullanılarak yapılan sınıflandırma sonuçları

4.7 Deney Sonuçlarının Değerlendirilmesi

Yapılan tüm deneyler sonucunda en başarılı sonuçlar, veri kümesine özellik türetme ve özellik seçimi uygulandığında elde edilmiştir. Özellik türetme ve seçimi yapılmadan önce %86.21 olan sınıflandırma başarısı, işlemleden sonra %88.21'e yükselmiştir. Özkodlayıcı ile özellik çıkarmak sınıflandırma başarısını olumsuz etkilemiştir. En başarılı sonucu veren sınıflandırma algoritmasının rastgele orman olduğu tespit edilmiştir.

GAN, CGAN, WGAN ve CTGAN yöntemlerinden en başarılı olanın CTGAN olduğu tespit edilmiştir. CTGAN ile üretilen 20.000 adet sentetik sahte işlem üretilerek veri kümesine eklendiğinde sınıflandırma başarısı %88.64'e yükselmektedir. Üretilen 50.000 adet yasal ve sahte işlemlerden oluşan sentetik örnekler ile sınıflandırma başarısı %88.75'e yükselmektedir.

4.8 Kredi Kartı Sahtekarlık Tespiti Sisteminin Kullanımı

Kredi kartı sahtekarlık tespiti sistemine ait deneyleri gerçekleştirmek amacıyla 160 çekirdekli 3.60 Ghz işlemciden ve 1 TB RAM'den oluşan IBM PowerAI bilgisayarı kullanılmıştır. Mevcut donanımlar ile bir aylık kredi kartı işlemleri (150.000 adet) kullanılarak rastgele orman modelinin eğitilmesi 30 saniyenin altında tamamlanırken,

derin öğrenme modelinin eğitilmesi ise 5 dk'nın altında tamamlanmaktadır. Eğitim süresi uzun olmasına rağmen, önceden eğitilmiş modeller kullanılarak yeni işlemler (100.000 adet) 5 saniyeden daha kısa bir sürede sınıflandırılabilmektedir. Bu süre, işlemin gerçekleştirildiği sırada (runtime) sahte olup olmadığının kontrol edilebilmesi için yeterli bir süredir. Bu çalışma sonucunda, sınıflandırıcı modelin belli aralıklarla eğitilerek, gerçekleştirilen her kredi kartı işleminin işlem sırasında sınıflandırılması ve sınıflandırma sonucuna göre işlemin onaylanması veya reddedilmesi önerilmiştir. Yapılan çalışmada ayda bir kez eğitmenin yeterli olabileceği görülmüştür.

Bankacılık işlemlerinde genellikle skor bazlı sistemler kullanıldığı için bu çalışmada gerçeklenen rastgele orman yöntemi ile her işleme ait sahte olma skoru çıkarılmıştır. Bu skor, işlemlerin sahte işlemlere ne kadar benzediğini belirtmektedir. Böylelikle bankalar tarafından belirlenecek farklı eşik değerleri ve parametreler ile işlemler engellenebilir veya kart sahiplerine ulaşılarak onay alınabilir. Örneğin, belirli bir tutarın üzerindeki işlemler için eşik değeri düşürülebilir veya sahte işlem oranının düşük olduğu işyeri kategorilerinden gelen işlemler için yüksek eşik değerleri uygulanabilir. Bu parametreler sistemi kullanacak bankaların stratejilerine göre belirlenebilir.

5
SONUÇ VE ÖNERİLER

Bu çalışmada kredi kartı sahtekarlıklarını tespit etmek amacıyla makine öğrenmesi ve derin öğrenme yöntemlerini bir arada kullanan yeni bir sistem önerilmiştir. Türkiye'de gerçekleştirilen banka işlemlerine ait gerçek bir veri kümesi kullanılarak sistem gerçeklenmiş ve yapılan deneylerin sonuçları değerlendirilmiştir.

Önerilen sistemde kart sahiplerinin harcama alışkanlıklarının daha iyi anlaşılması amacıyla karta ait önceki işlemler gruplandırılarak yeni özellikler türetilmiştir. Mevcut özelliklerle birlikte, türetilen özellikler arasından en başarılı olanlar seçilmiştir. Yapılan özellik türetme ve özellik seçme işlemlerinin sınıflandırma başarısını arttırdığı tespit edilmiştir. Mevcut özellikler kullanılarak rastgele orman sınıflandırıcısı ile %86.21 başarı elde edilirken, özellik türetimi ve seçimi sonucunda %88.64 başarı elde edilmiştir. Klasik yöntemlerden olan rastgele ormanın, derin öğrenme yöntemi olan derin sinir ağlarına göre daha başarılı sınıflandırma yaptığı tespit edilmiştir. Özkodlayıcılar kullanılarak özellik çıkarıldığında sınıflandırma başarısının olumsuz etkilendiği gözlemlenmiştir.

Kredi kartı sahtekarlığı tespiti çalışmalarında, genellikle veri kümelerinin sınırlı sayıda olması sebebiyle sorunlar yaşanmaktadır. Ayrıca veri kümesinde bulunan örneklerin sınıf dağılımlarının aşırı dengesiz olması da çalışmaların başarısını düşürmektedir. Bu çalışmada farklı veri üretme yöntemleri ile sentetik kredi kartı işlemleri üretiminin performansa etkisi değerlendirilmiş, CTGAN yöntemi ile sentetik veri üretiminin başarıyı arttırdığı gözlemlenmiştir. Hem sahte, hem yasal işlemler üretilerek, mevcut veri kümelerine eklendiğinde sınıflandırmaya olan etkileri incelenmiştir. Çalışmada kullanılan veri kümesi, yeterli sayıda işlem içerdiği için, yeni işlemler ekleyince sınıflandırma başarısında ciddi bir artış sağlanamamıştır. Ancak veri kümesinde bulunan örnek sayısı azaltılarak, yalnızca 500 sahte ve 2500 yasal işlem ile yapılan deneylerde başarının daha çok arttığı görülmüştür. Sentetik veri üretilmeden %80.59 sınıflandırma başarısı elde edilirken, 200 adet sentetik sahte işlem veri kümesine eklendiğinde başarı %82.33'e yükselmiştir.

Gerçek örnekler kullanılmadan, yalnızca sentetik işlemler kullanılarak yapılan deneylerde, sınıflandırma başarısının, gerçek işlemlere oldukça yakın sonuç verdiği tespit edilmiştir. 500 adet sentetik örnek kullanılarak yapılan deneylerde rastgele orman %74.44, derin yapay sinir ağları %74.53 başarı elde etmiştir. Bu da üretilen işlemlerin gerçek işlemlere benzediğini göstermektedir.

Kredi kartı işlemlerinin sahte olduğu, ancak kart sahibinin, işlemi kendisinin gerçekleştirmediğini bankaya bildirmesiyle anlaşılmaktadır. Bu sebeple yasal olarak kabul edilen işlemlerin, gerçekte sahte olma olasılığı bulunmaktadır. Sahte işlemlerin yasal olarak sınıflandırılması öğrenme sürecini olumsuz etkilemektedir. Gelecek çalışmalarda zayıf denetim (weak supervision) ve denetimsiz öğrenme (unsupervised learning) yöntemleri kullanılarak, yasal olarak bildirilmiş sahte işlemler tespit edilmeye çalışılacaktır.

KAYNAKÇA

[1] H. Hofmann, *UCI Machine Learning Repository, Statlog (German Credit Data) Data Set*. [Online]. Available: `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)` (visited on 05/29/2021).

[2] D. Dua, C. Graff, *UCI Machine Learning Repository, Statlog (Australian Credit Approval) Data Set*. [Online]. Available: `http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)` (visited on 05/29/2021).

[3] A. Dal Pozzolo, O. Caelen, R. A. Johnson, G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*, IEEE, 2015, pp. 159–166.

[4] K. Fu, D. Cheng, Y. Tu, L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*, Springer, 2016, pp. 483–490.

[5] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S.-k. Nam, Y. Song, J.-a. Yoon, J.-i. Kim, "Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning," *Expert Systems with Applications*, vol. 128, pp. 214–224, 2019.

[6] J. West, M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Computers & security*, vol. 57, pp. 47–66, 2016.

[7] Z. Zojaji, R. E. Atani, A. H. Monadjemi, *et al.*, "A survey of credit card fraud detection techniques: Data and technique oriented perspective," *arXiv preprint arXiv:1611.06439*, 2016.

[8] A. Abdallah, M. A. Maarof, A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[9] M. Zareapoor, P. Shamsolmoali, *et al.*, "Application of credit card fraud detection: Based on bagging ensemble classifier," *Procedia computer science*, vol. 48, no. 2015, pp. 679–685, 2015.

[10] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[11] A. C. Bahnsen, D. Aouada, A. Stojanovic, B. Ottersten, "Detecting credit card fraud using periodic features," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, IEEE, 2015, pp. 208–213.

[12] Y. Lucas, P.-E. Portier, L. Laporte, L. He-Guelton, O. Caelen, M. Granitzer, S. Calabretto, "Towards automated feature engineering for credit card fraud detection using multi-perspective hmms," *Future Generation Computer Systems*, vol. 102, pp. 393–402, 2020.

[13] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, "Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decision Support Systems*, vol. 75, pp. 38–48, 2015.

[14] X. Zhang, Y. Han, W. Xu, Q. Wang, "Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Information Sciences*, 2019.

[15] A. Singh, A. Jain, "Adaptive credit card fraud detection techniques based on feature selection method," in *Advances in computer communication and computational sciences*, Springer, 2019, pp. 167–178.

[16] A. Pumsirirat, L. Yan, "Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine," *International Journal of advanced computer science and applications*, vol. 9, no. 1, pp. 18–25, 2018.

[17] A. K. Gangwar, V. Ravi, "Wip: Generative adversarial network for oversampling data in credit card fraud detection," in *International Conference on Information Systems Security*, Springer, 2019, pp. 123–134.

[18] J. Chen, Y. Shen, R. Ali, "Credit card fraud detection using sparse autoencoder and generative adversarial network," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, 2018, pp. 1054–1059.

[19] J. Hwang, K. Kim, "An efficient domain-adaptation method using gan for fraud detection,"

[20] J. Jordon, J. Yoon, M. Van Der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2018.

[21] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[22] A. Sethia, R. Patel, P. Raut, "Data augmentation using generative models for credit card fraud detection," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, 2018, pp. 1–6.

[23] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019.

[24] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[25] S. Lundberg, S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[26] L. S. Shapley, *Notes on the N-person Game–II: The Value of an N-person Game*. Rand Corporation, 1951.

[27] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] S. De Cesco, J. B. Davis, P. E. Brennan, "Targetdb: A target information aggregation tool and tractability predictor," *PloS one*, vol. 15, no. 9, e0232644, 2020.

[30] P. Geurts, D. Ernst, L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[31] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[32] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in statistics*, Springer, 1992, pp. 162–190.

[33] G. E. Hinton, R. S. Zemel, "Autoencoders, minimum description length, and helmholtz free energy," *Advances in neural information processing systems*, vol. 6, pp. 3–10, 1994.

[34] H. Nugroho, M. Susanty, A. Irawan, M. Koyimatu, A. Yunita, "Fully convolutional variational autoencoder for feature extraction of fire detection system," *Jurnal Ilmu Komputer dan Informasi*, vol. 13, no. 1, pp. 9–15, 2020.

[35] H. Ahmed, M. L. D. Wong, A. K. Nandi, "Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features," *Mechanical Systems and Signal Processing*, vol. 99, pp. 459–477, 2018.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[37] F. Hu, H. Li, "A novel boundary oversampling algorithm based on neighborhood rough set model: Nrsboundary-smote," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[38] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[39] C. Öngün, A. Temizel, "Paired 3d model generation with conditional generative adversarial networks," in *European Conference on Computer Vision*, Springer, 2018, pp. 473–487.

[40] P. Salehi, A. Chalechale, M. Taghizadeh, "Generative adversarial networks (gans): An overview of theoretical model, evaluation metrics, and recent developments," *arXiv preprint arXiv:2005.13178*, 2020.

[41] M. Arjovsky, S. Chintala, L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[42] M. Mirza, S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[43] B. Can, A. G. Yavuz, E. M. Karsligil, M. A. Guvensan, "A closer look ınto the characteristics of fraudulent card transactions," *IEEE Access*, vol. 8, pp. 166 095–166 109, 2020. DOI: 10.1109/ACCESS.2020.3022315.

[44] S. Patro, K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[45] S. Garavaglia, A. Sharma, "A smart guide to dummy variables: Four applications and a macro," in *Proceedings of the northeast SAS users group conference*, vol. 43, 1998.

[46] P. Lerman, "Fitting segmented regression models by grid search," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 1, pp. 77–84, 1980.

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. E. Bayhan, M. E. Karsligil, A. G. Yavuz, and M. A. Güvensan. "Özellik Seçiminin Kredi Kartı Sahtekarlığı Tespiti Başarısına Etkisi" In 2021 29th Signal Processing and Communications Applications Conference (SIU), IEEE, 2021.