ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME KULLANARAK IP ÜZERİNDEN SES HİZMETİ VEREN ŞEBEKELERDE SAHTEKARLIĞA YÖNELİK ÇAĞRILARIN TESPİTİ

Murat BAĞ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2019

Her hakkı saklıdır

TEZ ONAYI

Murat BAĞ tarafından hazırlanan **"Derin Öğrenme Kullanarak IP Üzerinden Ses Hizmeti Veren Şebekelerde Sahtekârlığa Yönelik Çağrıların Tespiti"** adlı tez çalışması 03/09/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Ahmet AKBULUT

Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı

Jüri Üyeleri :

Başkan : Dr. Öğr. Üyesi Alparslan Çağrı YAPICI

Başkent Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Hakkı Alparslan ILGIN

Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı

Üye : Dr. Öğr. Üyesi Ahmet AKBULUT

Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Özlem YILDIRIM
Enstitü Müdür V.

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

03/09/2019

Murat BAĞ

ÖZET

Yüksek Lisans Tezi

Derin Öğrenme Kullanarak IP Üzerinden Ses Hizmeti Veren Şebekelerde Sahtekârlığa Yönelik Çağrıların Tespiti

Murat BAĞ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği

Danışman: Dr. Öğr. Üyesi Ahmet AKBULUT

Telekomünikasyon operatörleri altyapısı kullanılarak yapılan sahtekârlık/kötü niyetli kullanım, bu endüstride gelir kayıplarına ve kullanıcı mağduriyetlerine sebep olmaktadır. Bu problem karşısında telekomünikasyon operatörlerince kullanılan sahtekârlık yönetim sistemleri servis özelinde ve mevcut sabit kural kümeleri çerçevesinde çalışmaktadır. Gerçek zamanlı verilerin kullanılmaması nedeniyle, kayıplar gerçekleştikten sonra sahtekarlık çağrıları tespit edilerek yenilerinin oluşması engellenmektedir.

Tez çalışması kapsamında; abonelerin çağrı detay kayıtları, gerçek zamanlı sinyalleşme izleri, demografik verileri ve ödeme verilerinin analizleri sonrası sahtekârlık tespitinde etkili parametreler belirlenerek iki ayrı yapay sinir ağı tasarlanmıştır. Bu ağlardan biri çevrimdışı tespit yaparken diğeri de gerçek zamanlı tespit yapmaktadır.

Tasarlanan ağlar; performanslarının ölçülmesi amacıyla geniş bir veri seti ile test edilmiştir. Ölçülen performansların, farklı bir makine öğrenme yöntemi ile karşılaştırılması için k-en yakın komşuluk (ing. k-nearest neighborhood, KNN) algoritması kullanılmıştır. Çevrimdışı sahtekarlık çağrı tespiti yapan ağda; diğer ağ performanslarına göre daha başarılı sonuçlar alınmıştır. Gerçek zamanlı tespit yapan ağda ise; çevrimdışı tespit performansına göre sahtekarlık çağrısı tespit oranları daha düşük olmasına rağmen, sahtekarlık nedeniyle oluşan kayıpların önemli ölçüde engellenebileceği görülmüştür.

Ağustos 2019, 79 sayfa

ANAHTAR KELİMELER: Derin Öğrenme, Çağrı Sahtekârlığı, Ses Şebekesi, IP Üzerinden Ses, Sahtekarlık, Gerçek Zamanlı Sahtekarlık Tespiti

ABSTRACT

Master Thesis

Fraud Detection in VOIP Networks Using Deep Learning

Murat BAĞ

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Ahmet AKBULUT

Fraud / malicious use of telecommunication services are causing significant losses for service providers. In the face of this problem, fraud management systems used by telecom operators operate in a service specific manner and within the framework of existing fixed rule sets. Due to the lack of real-time data usage, fraud calls are identified after losses occur. New rules are identified to prevent new fraud calls.

Within the scope of the thesis; subscriber call detail records, signaling traces, user demographic and payment data are used to train two different artificial neural networks for fraud detection.

In order to measure the performance of the designed networks, the performance of another network using k-nearest neighborhood (KNN) algorithm was trained and compared with the designed networks. Training the network with the data set created with call detail records, subscriber demographic and payment data; the network was able to detect fraud calls with the highest success rate. Although, the real-time fraudulent call detection performance was lower than offline detection performance, it could detect fraudulent calls before the connection has been established and prevent losses.

August 2019, 79 pages

KEY WORDS: Deep Learning, FMS, Fraud Call, Voice Network, VOIP, Fraud, Real-time Fraud Detection

TEŞEKKÜR

Yüksek Lisans tez çalışmam sürecinde bilgi ve tecrübesini esirgemeyen tez danışmanım değerli Dr. Öğr. Üyesi Ahmet Akbulut'a (Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı) ve çalışmalarımın esnasında her zaman destek olan eşime teşekkürlerimi sunarım.

Murat BAĞ

Ankara, Ağustos 2019

İÇİNDEKİLER

TEZ ONAY SAYFASI
ETİK ..... **i**
ÖZET ..... **i**
ABSTRACT ..... **ii**
TEŞEKKÜR ..... **iii**
KISALTMALAR DİZİNİ ..... **vii**
ŞEKİLLER DİZİNİ ..... **viii**
ÇİZELGELER DİZİNİ ..... **ix**
1. GİRİŞ **1**
2. PROBLEME GENEL BAKIŞ **3**
2.1 Temel Kavramlar **3**
2.1.1 Arabağlantı **3**
2.1.2 ÇDK **3**
2.1.3 Hususi şube santrali (HŞS) **3**
2.1.4 Sinyalleşme **4**
2.1.5 Yeni nesil şebekeler (YNŞ) **5**
2.2 Sahtekarlık Yöntemleri **5**
2.2.1 Çağrı transferi yöntemi **5**
2.2.2 Yanlış numara denetimi **7**
2.2.3 Gelir paylaşımı dolandırıcılığı **7**
2.2.4 Çoklu çağrı transferi sahtekarlığı **7**
2.2.5 Çağrı yönlendirme dolandırıcılığı **8**
3. DERİN ÖĞRENME **9**
3.1 Yapay Sinir Ağları **10**
3.1.1 İleri beslemeli YSA **11**
3.1.2 Geri beslemeli YSA **11**
3.2 En yakın k komşu algoritması **12**
3.3 Öğrenme Yöntemleri **13**
3.3.1 Gözetimli öğrenme **13**
3.3.2 Gözetimsiz öğrenme **13**
3.4 Aktivasyon fonksiyonları **14**
3.4.1 Doğrusal aktivasyon fonksiyonu **14**
3.4.2 Adım aktivasyon fonksiyonu **14**
3.4.3 Sigmoid aktivasyon fonksiyonu **15**
3.4.4 Tanjant hiperbolik aktivasyon fonksiyonu **15**
3.4.5 Eşik değer aktivasyon fonksiyonu **16**
3.4.6 Gauss aktivasyon fonksiyonu **16**
3.5 Hata Fonksiyonu **17**
3.5.1 İkinci dereceden hata fonksiyonu **17**
3.5.2 Çapraz entropi hata fonksiyonu **18**

3.6 Dereceli Azalma 19
3.6.1 Olasılıksal Dereceli Azalma 22
3.6.2 Dereceli azalma optimizasyon algoritmaları 23
3.7 Geri Yayılım Algoritması 24
3.8 K Katmanlı Çapraz Doğrulama 24
3.9 Eğitim Performansını Yükseltme Yöntemleri 25
3.9.1 Seyreltme düzenleme yöntemi 26
3.9.2 Örgü arama yöntemi ile parametre ayarlama 27
4. KAYNAK ÖZETLERİ 28
5. MATERYAL ve YÖNTEM 30
5.1 Çalışmada Kullanılan Veri Seti 30
5.1.1 Arama detay kayıtları kullanılarak hazırlanan veri seti 30
5.1.2 OBP sinyalleşmesi davet mesajları içerikleri kullanılarak hazırlanan veri seti . 31
5.2 Örnek Mimari Tasarımlar 33
5.3 Kullanılan Programlama Dili 35
5.4 Kullanılan Kütüphaneler 35
5.4.1 TensorFlow 35
5.4.2 Keras 35
5.4.3 Numpy 36
5.4.4 Scikit-learn 36
5.4.5 Pandas 36
5.5 Veri Setinin Düzenlenmesi 36
5.5.1 Sınıf dengesizliğinin kontrol edilmesi 37
5.5.2 Girdi ve Çıktı Kümelerinin Oluşturulması 37
5.5.3 Kategorik Değişkenlerin Ele Alınması 38
5.5.4 Eğitim ve test setlerinin belirlenmesi 39
5.5.5 Ölçeklendirme 39
5.6 Ağ Modeli Oluşturma 39
5.6.1 Giriş katmanını ve ilk gizli katmanın oluşturması 40
5.6.2 İkinci ve üçüncü gizli katmanların oluşturulması 40
5.6.3 Çıktı katmanının oluşturulması 40
6. DENEYSEL ÇALIŞMA 41
6.1 Ağ Parametrelerinin Seçimi 41
6.2 Ağ Performansı Değerlendirme Metrikleri 42
6.2.1 Doğruluk 43
6.2.2 Kesinlik 44
6.2.3 Hassaslık 44
6.2.4 Belirlilik 44
6.2.5 F1 skoru 44
6.3 ÇDK Bilgileri Kullanılarak Sahtekarlık Tespiti için Ağ Seçimi 45
6.3.1 Model değerlendirme 47
6.4 OBP Davet Mesajı kullanılarak Sahtekarlık Tespiti için Ağ Seçimi 50
6.4.1 Model Değerlendirme 51
6.5 Seçilen Ağ Performanslarını Yükseltme Çalışmaları 52
6.5.1 Model 1 53

6.5.2 Model SIP1 53
6.6 KEYK Algoritması ile Tasarım 54
6.6.1 K değerinin belirlenmesi 55
6.7 İşlemci İhtiyacı Az Olan İki Ağ Tasarımı 56
6.7.1 Agresif ve Tolerans veri setleri ile eğitilecek ağların seçimi 57
6.7.2 Model değerlendirme 59
6.7.3 Seçilen ağ performanslarını yükseltme çalışmaları 60
6.7.4 Agresif ve Tolerans ağlarının birleştirilmesi 61
6.8 Ağ performanslarının karşılaştırılması 62
6.8.1 Model 1 - KEYK karşılaştırması 62
6.8.2 Model SIP1 - KEYK karşılaştırması 64
6.8.3 Model 1 – Model SIP1 performans karşılaştırması 66
6.8.4 Model 1 – Karma 1 ağ performans karşılaştırması 69
7. SONUÇ 72
KAYNAKLAR 75
ÖZGEÇMİŞ 79

KISALTMALAR DİZİNİ

3GPP	3rd Generation Partnership Project
3NOP	3. Nesil Ortaklık Projesi
ATSE	Avrupa Telekomünikasyon Standartları Enstitüsü
CDR	Call Detail Record
ÇDK	Çağrı Detay Kayıtları
DN	Doğru Negatif
DP	Doğru Pozitif
ETSI	European Telecommunications Standard Institute
GİÜ	Grafik İşlemci Ünitesi
HŞS	Hususi Şube Santrali
IÇOAS	IP Çoklu Ortam Alt Sistemi
IPDK	IP Detay Kaydı
IPDR	IP Detail Record
IMS	IP Multimedia Subsystem
KEYK	K-En Yakın Komşuluk
KNN	K-Nearest Neighborhood
MİB	Merkezi İşlemci Birimi
MSE	Mean Squared Error
OBP	Oturum Başlatma Protokolü
PBX	Private Branch Exchange
DDÜ	Doğrultulmuş Doğrusal Ünite
SIP	Session Initiation Protocol
SYS	Sahtekarlık Yönetim Sistemi
YN	Yanlış Negatif
YP	Yanlış Pozitif
YNŞ	Yeni Nesil Şebekeler

ŞEKİLLER DİZİNİ

Şekil 2.1 Çağrı başlatmak için hedef santrale iletilen örnek OBP mesajı ve içeriği 4
Şekil 2.2 Örnek çağrı transferi yöntemi ile sahtekarlık 6
Şekil 2.3 Örnek Çoklu Çağrı transferi dolandırıcılığı Senaryosu 8
Şekil 3.1 Derin öğrenmenin yapay zeka ve makine öğrenmesindeki yeri 9
Şekil 3.2 Örnek YSA nöron yapısı 10
Şekil 3.3 Örnek ileri beslemeli yapay sinir ağı modeli 11
Şekil 3.4 Örnek geri beslemeli yapay sinir ağı 12
Şekil 3.5 Doğrusal aktivasyon fonksiyonu 14
Şekil 3.6 Adım aktivasyon fonksiyonu 15
Şekil 3.7 Sigmoid aktivasyon fonksiyonu 15
Şekil 3.8 Tanjant hiperbolik aktivasyon fonksiyonu 16
Şekil 3.9 Eşik değer aktivasyon fonksiyonu 16
Şekil 3.10 Gauss aktivasyon fonksiyonu 17
Şekil 3.11 İkinci dereceden hata fonksiyonunun hata değerine göre eğitim hızı değişimi .. 18
Şekil 3.12 Çapraz entropi hata fonksiyonunun hata değerine göre eğitim hızı değişimi 19
Şekil 3.13 Örnek hata fonksiyonu grafiği 20
Şekil 3.14 Dereceli azalma yöntemi ile en iyi hata değerine ulaşma yolu 21
Şekil 3.15 dereceli azalma yöntemi ile yerel minimuma ulaşma örneği 21
Şekil 3.16 k katmanlı çapraz doğrulamada k=10 için eğitim ve test setlerinin ayrılması 24
Şekil 3.17 Aşırı uyum - normal - yetersiz uyum örnek grafikleri 26
Şekil 3.18 Örnek seyreltme düzenleme yöntemi 27
Şekil 5.1 ÇDK verileri kullanılarak sahtekarlık çağrılarının tespit edilmesi için örnek mimari 33
Şekil 5.2 OBP davet mesajları kullanılarak sahtekarlık çağrılarının tespit edilmesi için örnek mimari 34
Şekil 5.3 Çağrı dağılımları 37
Şekil 6.1 Örnek döngü sayısı hata fonksiyonu grafiği 42
Şekil 6.2 Hata matrisi tahmin dağılımına göre tanımlamalar 43
Şekil 6.3 Birinci veri seti ile denenen modellerin hata fonksiyonlarının döngü sayısına göre değişim grafiği 49
Şekil 6.4 İkinci veri seti ile denenen modellerin hata fonksiyonlarının döngü sayısına göre değişim grafiği 52
Şekil 6.5 Örnek k değerlerinin KEYK modellemesi üzerine etkisi 55

ÇİZELGELER DİZİNİ

Çizelge 5.1 Birinci veri setinde kullanılan alanlar ve açıklamaları 30
Çizelge 5.2 Birinci veri seti örnek değerler 31
Çizelge 5.3 İkinci veri setinde kullanılan alanlar ve açıklamaları 32
Çizelge 5.4 İkinci veri seti örnek değerler 32
Çizelge 5.5 Çağrı detay kayıtlarından tekrarlayan verilerin çıkarılması ile üretilen veri seti kategorik değişkenleri 38
Çizelge 6.1 Kombinasyonları denenen parametreler ve değerler 41
Çizelge 6.2 Örgü arama modeli testlerinde performansı yüksek ÇDK bilgisi ile eğitilmiş ağlar 45
Çizelge 6.3 Birinci veri seti ile denenen model performansları 47
Çizelge 6.4 Örgü arama modeli testlerinde yüksek performans gösteren modeller ve sonuçları 50
Çizelge 6.5 İkinci veri seti ile denenen model performansları 51
Çizelge 6.6 Model 1 deneme sonuçları 53
Çizelge 6.7 Model SIP1 deneme sonuçları 54
Çizelge 6.8 KEYK algoritmasının birinci veri seti ile eğitimi sonrası performans metrik değerleri 55
Çizelge 6.9 KEYK algoritmasının ikinci veri seti ile eğitimi sonrası performans metrik değerleri 56
Çizelge 6.10 Agresif ve Tolerans modelleri için kombinasyonları denenen farklı parametre ve değerler 57
Çizelge 6.11 Örgü arama modeli testlerinde yüksek performans gösteren modeller ve test sonuçları 58
Çizelge 6.12 Agresif ve Tolerans İkinci veri setleri ile denenen model performansları 59
Çizelge 6.13 Agresif 1 model deneme sonuçları 60
Çizelge 6.14 Tolerans 1 model deneme sonuçları 61
Çizelge 6.15 Ağ ağırlıklarına göre performans metrikleri 62
Çizelge 6.16 Model 1 - KEYK karşılaştırmalı test sonuçları 62
Çizelge 6.17 Model 1 ve KEYK algoritmasının tahmin farklılıkları ve gerçek değerler tablosu 63
Çizelge 6.18 Model SIP1 - KEYK karşılaştırmalı test sonuçları 64
Çizelge 6.19 Model SIP1 ve KEYK algoritmasının tahmin farklılıkları ve gerçek değerler tablosu 64
Çizelge 6.20 Eşik değere göre model başarı parametre değerleri 66
Çizelge 6.21 Model 1 ve Model SIP1 tahmin farklılıkları ve gerçek değerler tablosu 67
Çizelge 6.22 Eşik değere göre model başarı parametre değerleri 69
Çizelge 6.23 Model 1 ve Karma 1 tahmin farklılıkları ve gerçek değerler tablosu 70

1. GİRİŞ

Telekomünikasyon operatörleri altyapısı kullanılarak yapılan sahtekârlık/kötü niyetli kullanım, telekomünikasyon bu endüstride gelir kayıplarına ve kullanıcı mağduriyetlerine sebep olmaktadır. Sahtekarlık aramaları nedeniyle 2017 yılı içerisinde küresel bazda telekomünikasyon sektörü kayıplarının 29 miyar dolar civarında olduğu tahmin edilmektedir (Anonymous 2018). Bu problem karşısında telekomünikasyon operatörlerince kullanılan sahtekârlık yönetim sistemleri (SYS) servis özelinde ve mevcut sabit kural kümeleri çerçevesinde çalışmaktadır. Gerçek zamanlı verilerin kullanılmaması nedeniyle, kayıplar gerçekleştikten sonra sahtekarlık çağrıları tespit edilerek yenilerinin oluşması engellenmektedir.

Telekomünikasyon sektöründe kullanılan sahtekârlık tespit sistemleri; operatörlere sahtekârlıkların tespit edilmesi, önleyici ve düzenleyici aksiyonlar alınması imkânı sağlamaktadır. Abonelerin çağrı detay kayıtları (ÇDK), demografik verileri ve ödeme verileri kullanılarak sahtekârlık kuralları belirlenmektedir (Phua vd. 2004). Sistemler, bu kurallara göre analiz yaparak çıktılar üretmektedir. Üretilen çıktılarda hatalı tespit olasılığı da bulunmaktadır.

Klasik sistemler, otomatize edilmiş bir tespit yöntemi kullanarak önceden belirlenmiş kurallara göre uyarı ve önlem çıktıları oluştururlar. Bu sistemlerde analiz, ancak çağrının sonlandırılması sonrası ÇDK incelenmesi ile yapılabilmektedir. Fakat klasik yöntemlerdeki en büyük engel, sabit kural kümelerine bağımlılık nedeniyle yeni sahtekârlık yöntemlerinin tespitlerinin kural kümesi güncellemesi olmadan mümkün olmamasıdır.

Tez çalışması kapsamında; abonelerin ÇDK, gerçek zamanlı sinyalleşme izleri, demografik verileri ve ödeme verilerinin analizleri sonrası sahtekârlık tespitinde etkili parametreler belirlenerek iki ayrı yapay sinir ağı tasarlanmıştır:

- İlk sistemde sahtekarlık tespitinde kullanılabilecek; gerçek zamanlı sinyalleşme verileri, demografik veriler ve ödeme verilerinden elde edilebilen parametrelere göre tasarım yapılmıştır. Bu sitem ile yapılan çalışmada başarı oranının yüksek olması beklenmektedir.

- İkinci sistemde sahtekârlık tespitinde kullanılabilecek abonelerin ÇDK, demografik verileri ve ödeme verilerinden elde edilebilen parametrelere göre tasarım yapılacaktır. Bu sistem ile yapılan çalışmada, başarılı tespit oranının daha düşük olması beklenmesine rağmen, zarar oluşmadan tespit yapılarak kayıpların minimize edilmesi hedeflenmektedir.

Tasarlanan ağların başarımlarını ölçmek için k-en yakın komşuluk (KEYK) algoritması, aynı eğitim ve test kümeleri ile çalıştırılmıştır. Tüm sistemlerin başarımları karşılaştırmalı olarak değerlendirilmiştir.

2. PROBLEME GENEL BAKIŞ

2.1 Temel Kavramlar

2.1.1 Arabağlantı

Arabağlantı genel tanımıyla; iki farklı servis sağlayıcı şebekesi arasındaki trafiğin sağlanması amacıyla, iki şebekenin birbiri ile irtibatlandırılması olarak ifade edilir. Arabağlantı yapılmaksızın farklı şebeke abonelerinin haberleşmesi mümkün değildir. Servis sağlayıcılar kendi aralarında arabağlantı sözleşmesi imzalayarak, trafiklerini belirledikleri ücretlendirme kriterlerine göre, diğer servis sağlayıcı şebekesine gönderebilirler. Bu sayede, bir operatör aboneleri, farklı bir operatör aboneliğinde çağrı ve görüşme yapabilir.

2.1.2 ÇDK

Bir ÇDK; telefon santrali tarafından üretilen ve bu santral üzerinden geçen arama ya da başka telekomünikasyon işleminin (ör. metin mesajı) ayrıntılarını belgeleyen veri kaydıdır. Kayıt; aramanın süresi, arama zamanı, tamamlanma durumu, kaynak numarası ve hedef numarası gibi çeşitli özelliklerini içerir. ÇDK, bir telekomünikasyon işleminin belirli bir örneğini tanımlayan veri alanlarını içerir ancak bu işlemin içeriğini (ör. ses kaydı) barındırmaz.

Bu kayıtların asıl amacı abone faturalandırmasının yapılmasıdır. Asıl işlevinin yanında; kullanım istatistikleri raporlanması, trafik yoğunluklarının belirlenmesi ya da sahtekarlık tespiti amacıyla da kullanılmaktadır.

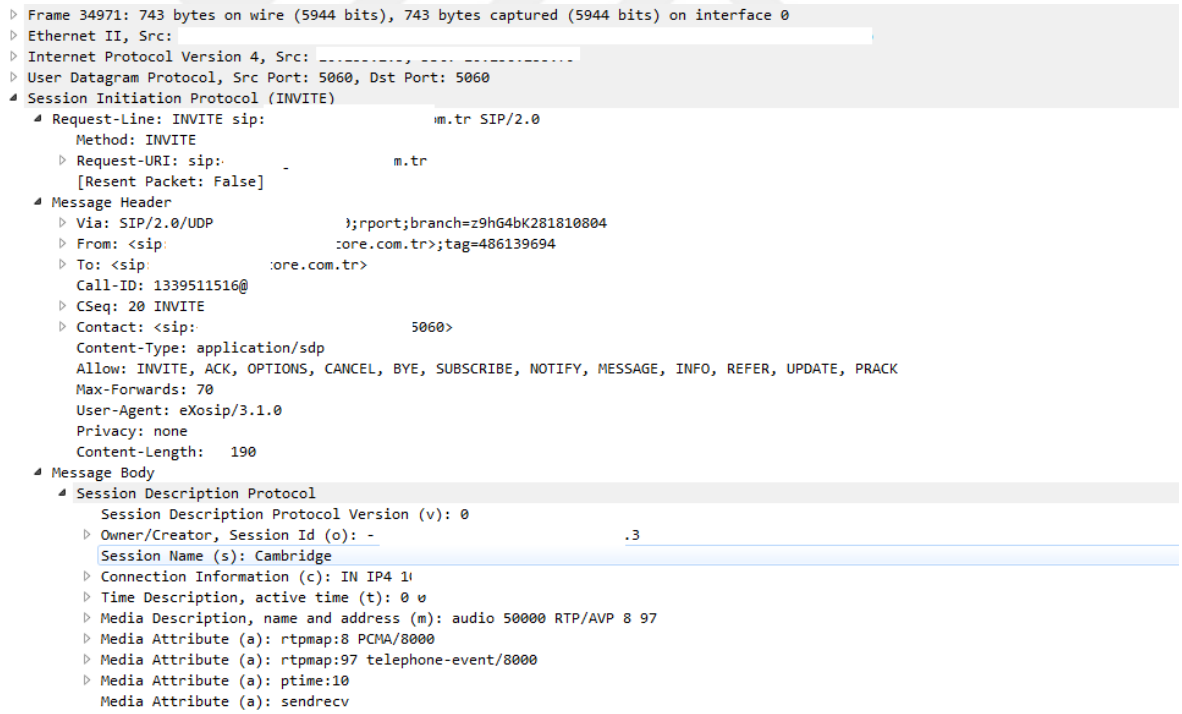
2.1.3 Hususi şube santrali (HŞS)

HŞS (ing. private branch exchange, PBX); özel bir organizasyona hizmet veren ve kurumdaki çok sayıda telefon arasında iletişim sağlayan bir telefon santrali veya anahtarlama sistemidir. Ayrıca, servis sağlayıcı santrali ile iletişim kurarak içeriden yapılan çağrıların ulusal şebekeye çıkmasını sağlar.

2.1.4 Sinyalleşme

Sinyalleşme; bir telekomünikasyon devresinin kurulması, kontrolü ve ağın yönetimi ile ilgili bilgi alışverişinin yapılmasını sağlar. Sinyalleşme sistemleri, dünya çapında otomatik ve standartlaştırılmış telefon hizmetlerinin sağlanmasında temel yapı taşıdır. İletilen mesajın standart olması farklı ulusal sistemler arasında arayüz sağlar.

IP üzerinden haberleşmede kullanılan protokollerden biri olan OBP sinyalleşmesi ile bir veya birden fazla kullanıcının dahil olduğu oturumlar (oturum açma, kapama, güncelleme) yönetilebilir. Şekil 2.1'de OBP ile çağrı kurulumu sırasında, hedef santrale gönderilen ilk mesaj görülmektedir. Mesaj içeriğindeki bazı alanlar kişisel verilerin korunması amacıyla gizlenmiştir.



Şekil 2.1 Çağrı başlatmak için hedef santrale iletilen örnek OBP mesajı ve içeriği

2.1.5 Yeni nesil şebekeler (YNŞ)

Telekomünikasyon çekirdek şebekeleri ve erişim şebekelerindeki mimari değişikliklerin temeli olan YNŞ (ing. next generation networks, NGN); bir servis sağlayıcının tüm hizmetlerinin IP altyapısı üzerinden, paketler halinde iletmesi fikri üzerine kuruludur. Bu nedenle uçtan uca IP terimi, eskiden telefon merkezli ağların YNŞ'e doğru dönüşümünü tanımlamak için de kullanılmaktadır.

Ses uygulamaları için YNŞ üzerinde santral görevi gören ve anahtarlama işlevini yerine getiren cihazlar yazılımsal santrallerdir. Bu cihazların en önemli özelliği, mevcut devre anahtarlamalı ağlar ile iletişim kurma, o ağlardan gelen ve o ağlara giden çağrıları yönetebilmesidir.

IP Çoklu Ortam Alt Sistemi (IÇOAS), Avrupa Telekomünikasyon Standartları Enstitüsü (ATSE) ve 3. Nesil Ortaklık Projesi (3NOP) tarafından tanımlanan bir internet medya hizmetleri için standartlaştırılmış bir YNŞ mimarisidir.

2.2 Sahtekarlık Yöntemleri

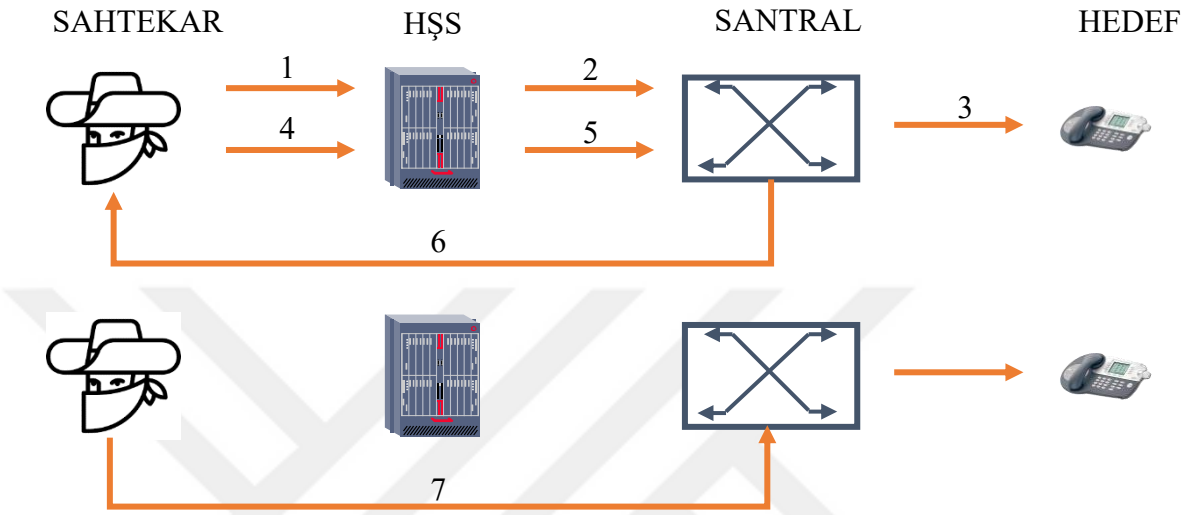
Telekom servis sağlayıcıları, sahtekarlığa açık ve koruyucu programlar olmaması durumunda savunmasızdır. Kötü niyetli kullanıcılar; tespit edilmesi, izlenmesi ve önlenmesi zor bir şekilde çağrıları manipüle edebilirler. Abonelerin çoğunlukla, servis sağlayıcının güvenlik tedbirleri dışında hesabını koruyacak uygulamaları bulunmamaktadır. Sahtekarlık sonrası, aboneler yüksek fatura ile karşı karşıya kalabilmekte, servis sağlayıcılar ise maddi zarar ve müşteri memnuniyetsizlikleri ile karşılaşmaktadırlar (Anonymous 2018).

Sahtekarlık aramaları, farklı yöntemler ve farklı amaçlar ile yapılabilmektedir.

2.2.1 Çağrı transferi yöntemi

Bu senaryoda, sahtekar bir HŞS hattına girer ve HŞS'nin servislerini ücretsiz uzun mesafe aramaları yapmak için kullanır. Sahtekar; ele geçirdiği HŞS'ye, gelen çağrıları kendi hattına

transfer etmesi talimatı vererek kendi telefon servisine abone olan kullanıcıların ücretsiz görüşmesini sağlar (Anonymous 2018).



Şekil 2.2 Örnek çağrı transferi yöntemi ile sahtekarlık

Şekil 2.2'de çağrı transferi yöntemi ile sahtekarlık adım adım belirtilmiş açıklamaları aşağıya eklenmiştir.

1. HŞS hat üzerinden çağrı yapılması.
2. Çağrı başlatma isteğinin santrale gelmesi.
3. Santralin kendi abonesinden gelen çağrıyı hedefe iletmesi.
4. Sahtekar HŞS'nin santrale bu çağrıyı kendi hattına aktarma isteği göndermesini istemesi.
5. HŞS'nin isteği göndermesi.
6. Çağrı transferi için santral ile sahtekar hattı arasında mesajlaşılma.
7. Transferin tamamlanması ve ücretlendirmenin HŞS'ye çıkarak sahtekarın istediği hedef ile ücretsiz görüşmesi.

2.2.2 Yanlış numara denetimi

Aranan numaranın serviste olmayan bir numara olması durumunda, arayan kısa bir anons ile hatalı numara çevirdiği yönünde bilgilendirilir. Çağrı cevaplanmadığı için arayan ve aranan taraf arasında bir bağlantı kurulmaz. Çağrı kurulmadığı için de ücretlendirme olmaz.

Sahtekarlık durumunda, aranan numara hatası alındığında arayan tarafa anons basılır fakat çağrı başlatan servis sağlayıcıya çağrı başarılı olarak kurulmuş gibi gösterilir. Bu gibi durumda servis sağlayıcı hem itibar kaybı hem de maddi kayıp yaşar (Anonymous 2018).

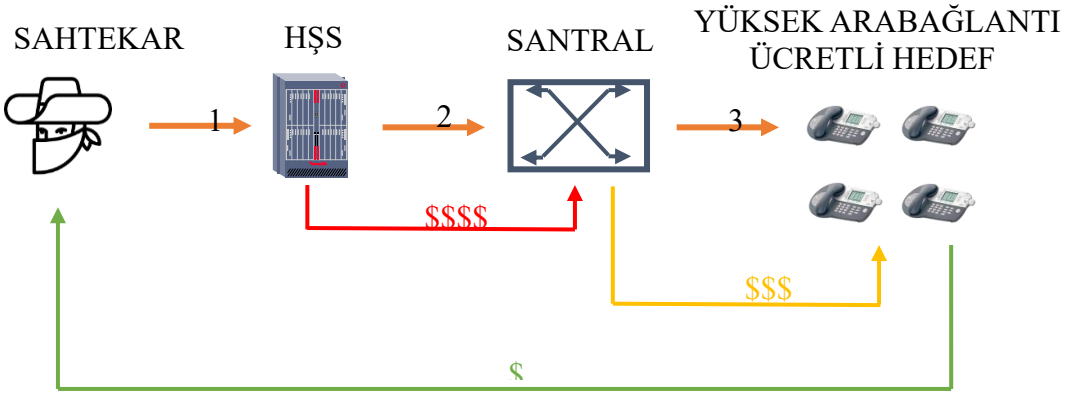
2.2.3 Gelir paylaşımı dolandırıcılığı

Bu tip sahtekarlıklar, servis sağlayıcının arabağlantı anlaşmalarının kötüye kullanılması ile oluşur. Dolandırıcının amacı, yüksek arabağlantı ücreti olan bir hedef numaraya çağrı yapılmak ve bunun karşılığında servis sağlayıcıya ödenmesi gereken ücreti ödememektir. Bu tip sahtekarlıklarda, ulusal ya da uluslararası numaralarda görülebilir ve karşı operatör ile anlaşılarak kar paylaşılır. Teknik olarak yasal olan bu tip durumlar, çağrı transferi ya da diğer sahtekarlık yöntemleri ile yapılabilir.

Bu dolandırıcılığın tespiti için, yüksek arabağlantı ücreti olan hedeflere yapılan çağrıların anlık artışlarının takip edilmesi gerekmektedir (Anonymous 2018).

2.2.4 Çoklu çağrı transferi sahtekarlığı

Çoklu çağrı transferleri sahtekarlığı; gelir paylaşımı dolandırıcılığı kazancını iki katına çıkarmak için geliştirilmiş, tespit edilmesi daha karmaşık olan bir tekniktir.



Şekil 2.3 Örnek çoklu çağrı transferi dolandırıcılığı senaryosu (Anonymous 2018)

Dolandırıcı çağrı transferi dolandırıcılığı yönteminde olduğu gibi çağrı kurar fakat yönlendirmeyi kendi numarasına değil başka bir yüksek ücretli numaraya yapar. Bu işlemi şekil 2.3'deki adımları takip ederek, aynı anda çok sayıda çağrı kurmak için tekrar tekrar uygular. Bu dolandırıcılık yöntemi gelir paylaşımı dolandırıcılığına benzer şekilde anlık çağrı sayısındaki artış ile yapılabilir (Anonymous 2018).

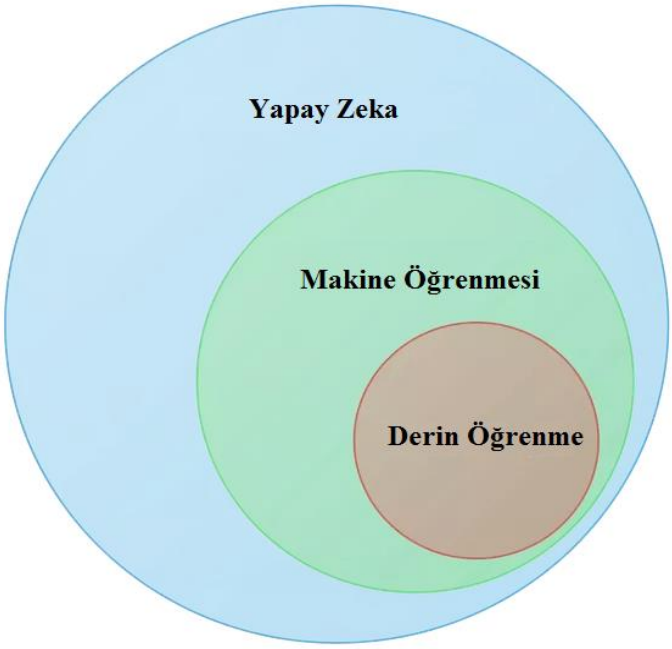
2.2.5 Çağrı yönlendirme dolandırıcılığı

Bu dolandırıcılık yöntemi, çağrı transferi dolandırıcılığının gelişmiş bir şeklidir. Sahtekar; abone santraline bağlanarak gelen tüm çağrıları kar paylaşımının olduğu yüksek arabağlantı ücreti olan bir numaraya yönlendirir. Daha sonra yönlendirdiği numaraya pek çok çağrı yaparak kar elde eder (Anonymous 2018).

3. DERİN ÖĞRENME

Derin öğrenme; çok katmanlı ileri beslemeli sinir ağlarının eğitim sürecidir ve makine öğrenmesi metotlarının bir parçasıdır (Anonymous 2018). Birden fazla gizli katman olması nedeniyle bu öğrenme şekli için derin ifadesi kullanılmıştır. Makine öğrenme yöntemleri günlük hayatta pek çok alanda uygulanmaktadır. Örnek olarak; yüz tanıma (Mohammed vd 2011, Guo vd. 2000), otonom araçlar (Chen vd. 2015, Sallab vd. 2017), siyah beyaz eski resimlerin renklendirilmesi (Liu vd. 2017), hastalık teşhisi (Bruijne 2016, FIÇICI 2016) ve sınıflandırma (Harrington 2012) gibi alanlarda kullanılmaktadır.

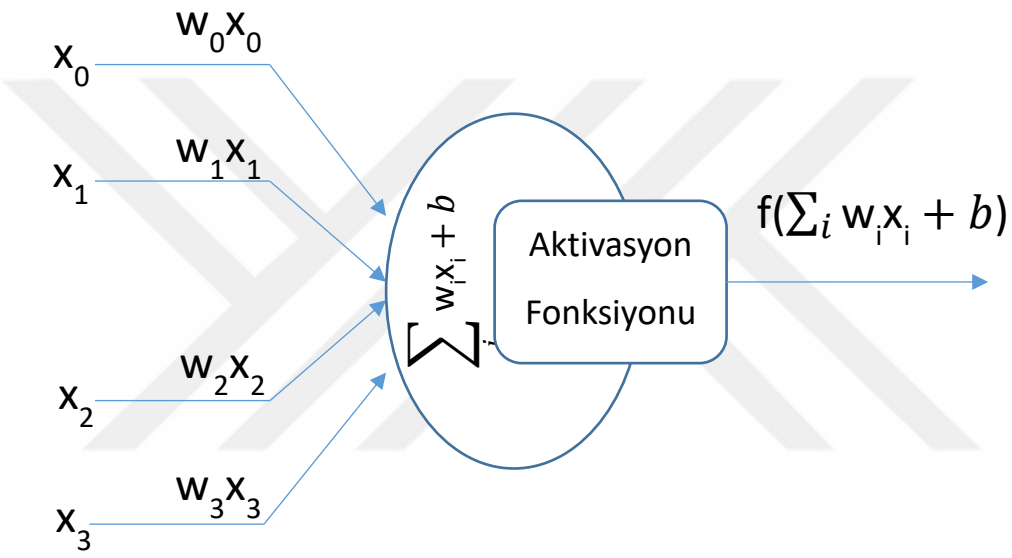
Derin öğrenme modelleri, biyolojik sinir sisteminin bilgi işleme ve iletişim kalıpları ile benzerlikler gösterir. Örnek olarak sinirsel kodlama, çeşitli uyaranlar ve bunlara verilen sinirsel tepkilerin ilişkilerini tanımlamaya çalışır. Modern derin öğrenme modellerinin çoğu, bir yapay sinir ağına (YSA) dayandırılmıştır (Anonymous 2018).



Şekil 3.1 Derin öğrenmenin yapay zeka ve makine öğrenmesindeki yeri

3.1 Yapay Sinir Ağları

YSA; insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama, genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır (Öztemel 2003).

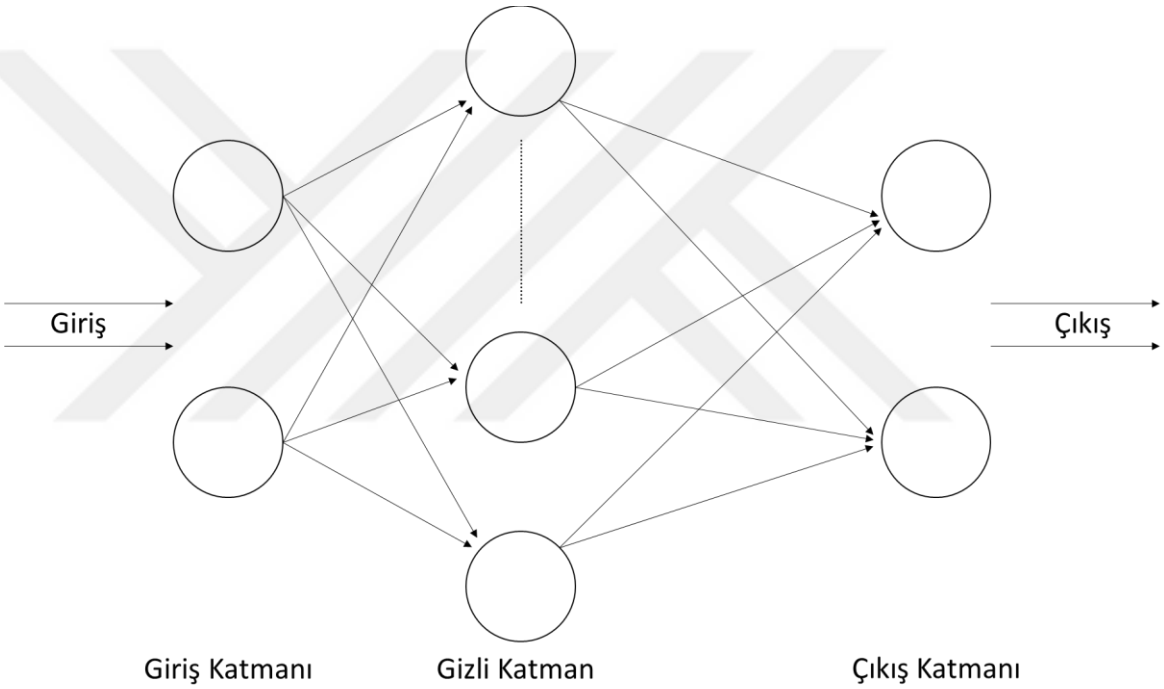


Şekil 3.2 Örnek YSA nöron yapısı

YSA'nın; mimarı yapı, öğrenme algoritması ve aktivasyon fonksiyonu olmak üzere 3 temel bileşeni bulunmaktadır. Şekil 3.2'de YSA içerisinde yer alan bir nöronun çalışma prensibi görülmektedir.

3.1.1 İleri beslemeli YSA

İleri beslemeli yapay sinir ağları, çok katmanlı yapay sinir ağlarının bir türüdür. İleri beslemeli YSA'da nöronlar genellikle katmanlara ayrılmışlardır. Sinyaller giriş katmanından başlayarak tek yönlü olarak çıkış katmanına iletilir. Aynı katmandaki nöronlar arasında iletim olmaz. Yani süreç; girişten çıkışa doğru, doğrusal olmayan statik bir işlevle gerçekleştirilir ve sonlandırılır (Öztemel 2003).



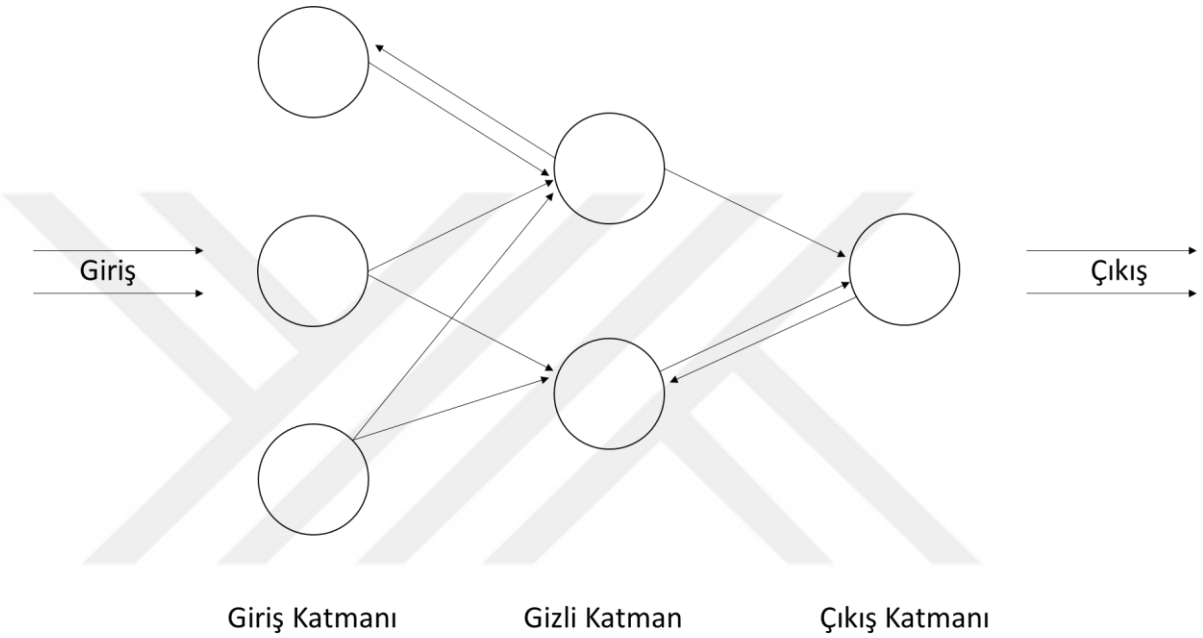
Şekil 3.3 Örnek ileri beslemeli yapay sinir ağı modeli

İleri beslemeli yapay sinir ağlarında temelde; giriş, gizli ve çıkış olmak üzere 3 çeşit katman bulunur. Gizli katmanın kaç seviye olacağı tamamen probleme göre belirlenmektedir. Her katman ve seviyede 1 veya daha çok sayıda nöron bulunabilir (şekil 3.3).

3.1.2 Geri beslemeli YSA

Geri beslemeli YSA'da, en az bir nöronun çıkışı kendisine ya da diğer nöronlara giriş olarak verilir ve genellikle geri besleme, bir geciktirme elemanı üzerinden yapılır (Ataseven 2013).

Geri besleme, bir katmandaki nöronlar arasında olduğu gibi katmanlar arasındaki nöronlar arasında da olabilir. Bu yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir (Kabalcı 2013). Geri beslemeli YSA'larda amaç, eğitim esnasında hatayı en aza indirerek ilerlemektir.



Şekil 3.4 Örnek geri beslemeli yapay sinir ağı

Geri beslemeli YSA'ların dinamik hafızaları vardır ve hem anlık çıkışı hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundurlar. Şekil 3.4'te örnek akış gösterilmektedir. Geri beslemeli ağlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır (Uslu 2016).

3.2 En yakın k komşu algoritması

KEYK algoritması, yapay sinir ağları ve vektör destek makineleri gibi daha karmaşık sınıflandırıcılar için genellikle bir referans noktası olarak kullanılan sağlam ve çok yönlü bir sınıflandırıcıdır (Zakka 2016). Basit yapısına rağmen, KEYK daha güçlü karmaşık sınıflandırıcılardan daha iyi performans gösterebilir ve ekonomik tahmin, veri sıkıştırma ve

genetik gibi çeşitli uygulamalarda kullanılmaktadır (Anonymous 2015). KEYK sınıflandırıcı, ayrıca parametrik olmayan ve aşamalı bir öğrenme algoritmasıdır.

Sınıflandırma ayarlarında KEYK, tahminlerini k sayısı kadar benzer örneğin değerlendirilmesi ile çoğunluk kararına göre yapar. Bu çoğunluk kararı Öklid uzaklığına göre eşitlik 3.1'de ifade edilmektedir (Zakka 2016).

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \cdots + (x_n - x_n')^2} \quad (3.1)$$

Her bir sınıf için koşullu olasılık değerlerini tahmin eder.

Bu çalışma kapsamında ağ eğitimleri sonrası modellerin test edilebilmesi amacıyla KEYK algoritması eğitilerek ağ başarıları diğer modeller ile karşılaştırılacaktır.

3.3 Öğrenme Yöntemleri

3.3.1 Gözetimli öğrenme

Gözetimli öğrenme; eldeki girdi-çıktı verileri ve çıktıları bilinmeyen girdi verileri ile tahmin yapılmasına olanak veren öğrenme algoritmaları oluşturur. En yaygın kullanımı sınıflandırma problemleri içindir. Gözetimli öğrenmede kullanılan veri seti, girdiler ve bunlarla eşleşmiş beklenen çıktı değerlerini içerir. Girdiler etiketlenerek yapay zeka sistemine eğitim verisi olarak gönderilir (Öztemel 2003). Yeterli miktarda eğitim verisi sonrası sistem, eğitim tamamlandığı zaman etiketlenmiş verileri ayırt edebilir ve sınıflandırabilir duruma gelmiş olur.

3.3.2 Gözetimsiz öğrenme

Gözetimsiz öğrenme, bir yapay zeka algoritmasının sınıflandırılmamış ya da etiketlenmemiş bilgiler ile eğitimi ve algoritmanın bu verileri kullanarak rehbersiz olarak kullanmasıdır. Gözetimsiz öğrenmede, bir yapay zeka sistemi, kategorileri verilmemiş olsa bile, sırasız bilgileri benzerlik ve farklılıklara göre gruplandırabilir. Gözetimsiz öğrenme ile eğitilebilen ağlar genellikle üretken öğrenme modelleri ile ilişkilidir. Etiketlenmemiş ve

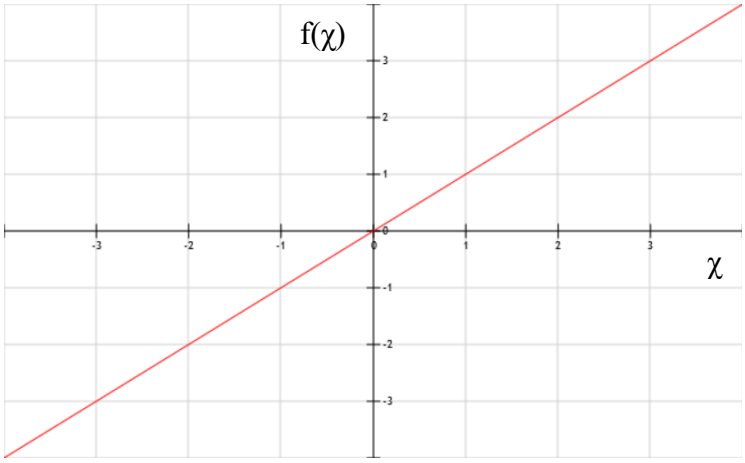
sınıflandırılmamış veriler yapay zeka sistemine sunulur ve sistem algoritmaları önceden eğitilmeden bu veri üzerinde çalışır (Öztemel 2003). Çıktı algoritmalara bağlıdır. Bir sistemi gözetimsiz öğrenmeye tabi tutmak yapay zekayı test etmenin bir yolu olarak da kullanılır.

3.4 Aktivasyon fonksiyonları

Aktivasyon fonksiyonu, katmanlar arasında çıktıların sonraki katmana iletimlerinde rol alır. Giren ve çıkan katmanlar arasındaki eşleşmeyi sağlar. Aktivasyon fonksiyonları, ağ performansında önemli etkiye sahiptir. Ağın doğrusal olmayan yapıyı öğrenmesini sağlayan bileşenidir.

3.4.1 Doğrusal aktivasyon fonksiyonu

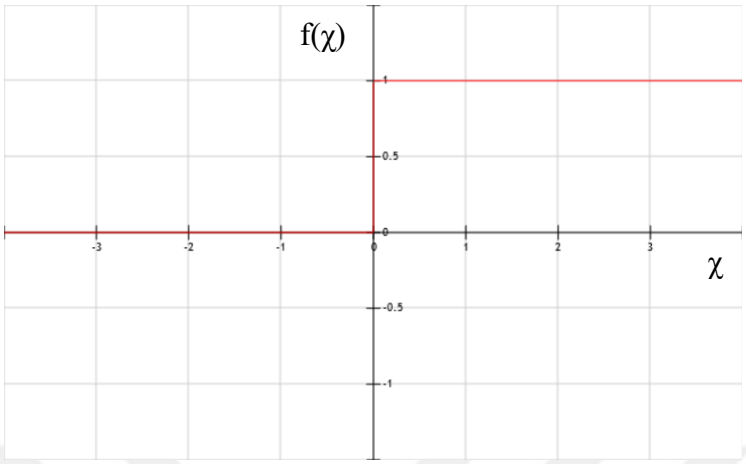
Doğrusal aktivasyon fonksiyonu genelde doğrusal problemlerin çözümünde kullanılır. Girdi ile çıktı arasında doğrusal bir ilişki vardır ve bu aktivasyon fonksiyonu genelde giriş katmanında kullanılır.



Şekil 3.5 Doğrusal aktivasyon fonksiyonu

3.4.2 Adım aktivasyon fonksiyonu

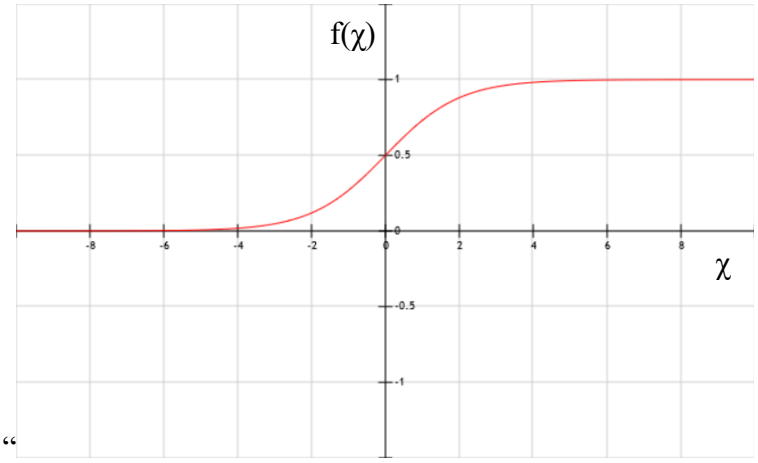
Bu fonksiyon gelen girdinin belirlenen bir eşik değerin altında veya üstünde olması durumuna göre, nöron çıktısı 1 veya 0 değerini alır (Sayad 2010).



Şekil 3.6 Adım aktivasyon fonksiyonu

3.4.3 Sigmoid aktivasyon fonksiyonu

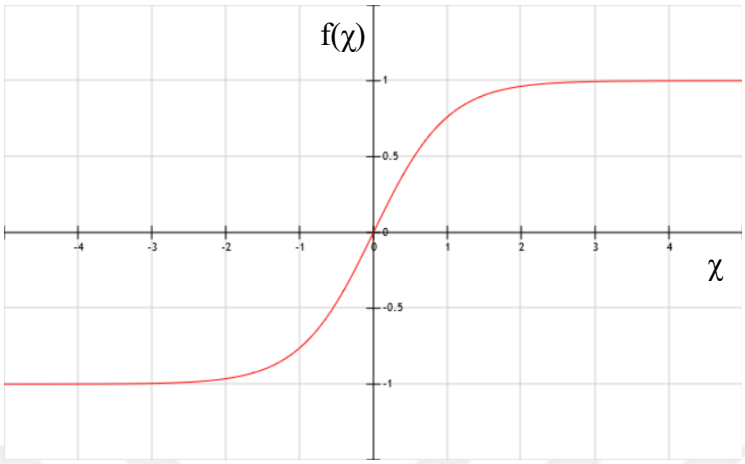
Sigmoid aktivasyon fonksiyonu; doğrusal olmayan, sürekli ve türevi alınabilir bir fonksiyondur. Tahmin uygulamalarında yoğun olarak kullanılmaktadır (Sharma 2017). Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.



Şekil 3.7 Sigmoid aktivasyon fonksiyonu (Sharma 2017)

3.4.4 Tanjant hiperbolik aktivasyon fonksiyonu

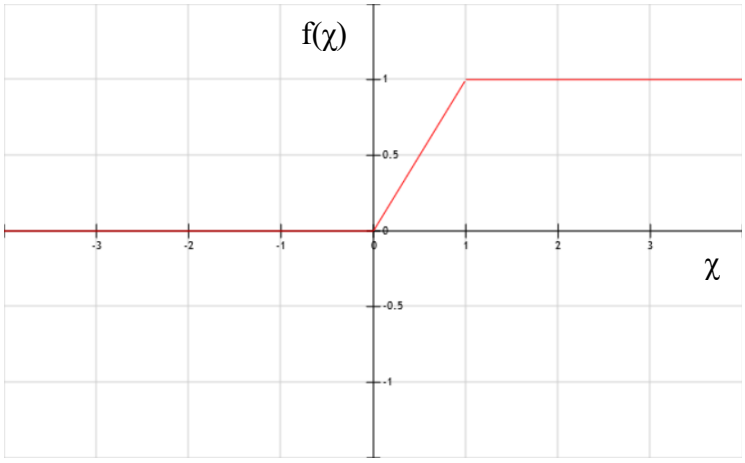
Tanjant hiperbolik (tanh) aktivasyon fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Çıkış değerleri -1 ile 1 arasında olabilir.



Şekil 3.8 Tanjant hiperbolik aktivasyon fonksiyonu

3.4.5 Eşik değer aktivasyon fonksiyonu

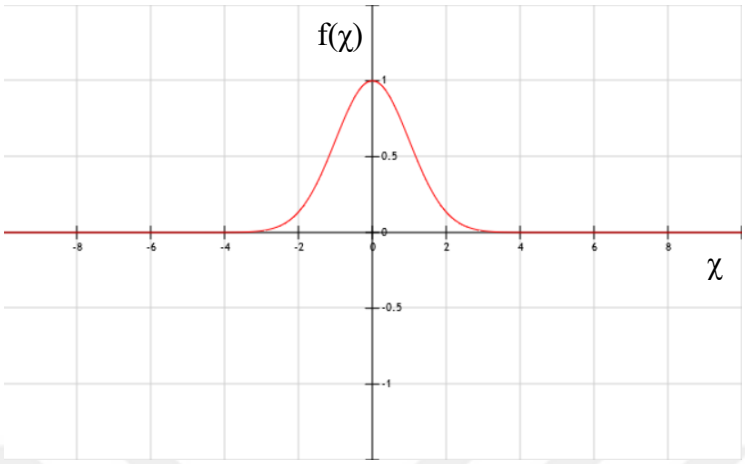
Bu fonksiyon; nörona girdisi, 0'dan küçük eşit olduğunda nöronun çıktısı 0, 1'den büyük eşit olduğunda nöronun çıktısı 1, 0 ile 1 arasında olduğunda ise yine kendisi olan çıktılar üretmektedir.



Şekil 3.9 Eşik değer aktivasyon fonksiyonu

3.4.6 Gauss aktivasyon fonksiyonu

Girdinin seçilen bir ortalama değere ne kadar yakın olduğuna bağlı olarak nöron çıkışı (yakın / uzak) olasılığı 1 ve 0 arasında yorumlanır.



Şekil 3.10 Gauss aktivasyon fonksiyonu

3.5 Hata Fonksiyonu

YSA çıktı hataları, hata fonksiyonları kullanılarak değerlendirilir. Hata fonksiyonları, eğitim setindeki her bir değer ile YSA çıktı farkları kullanılarak oluşturulur. YSA tasarımı ve problem tipine göre farklı hata fonksiyonları kullanılabilmektedir. YSA eğitimi ile hata değerlerinin minimize edilmesi ve mümkün olan en düşük değere gelmesi hedeflenmektedir. İyi eğitilen bir yapay sinir ağı minimum hataya sahiptir. İkinci dereceden (kuadratik) hata fonksiyonu ve çapraz entropi hata fonksiyonu en çok kullanılan hata fonksiyonlarıdır.

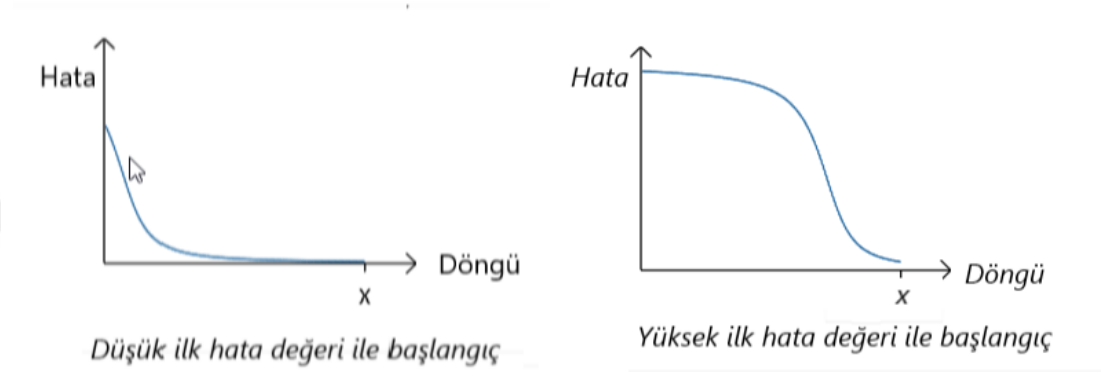
3.5.1 İkinci dereceden hata fonksiyonu

İkinci dereceden hata fonksiyonu yaygın olarak kullanılan hata fonksiyonlarından biridir. Aynı zamanda ortalama kare hatası (ing. Mean Squared Error –MSE) olarak bilinir ve eşitlik 3.2 ile göstermektedir (Grover 2018).

$$C(W,b) \equiv \frac{1}{2n}\sum_x ||Y(\chi) - \alpha||^2 \qquad (3.2)$$

Eşitlik 3.2'de $C(W,b)$ ikinci dereceden hata fonksiyonunu; W ağdaki tüm ağırlıkları toplamını, b tüm eşik değerlerinin toplamını, n eğitim girdilerinin toplam sayısını, $Y(\chi)$ gerçek değeri, α ağ çıktı vektörünü ifade etmektedir. İkinci dereceden hata fonksiyonu hesaplamada, toplamda bulunan hiçbir terim negatif olamayacağı için bu fonksiyon her

zaman sıfırdan büyük olacaktır. Ağ çıktısı ve gerçek değer arasındaki farkın yüksek olduğu durumlarda, şekil 3.11'de görülebileceği gibi eğitimde yavaşlama problemi oluşmaktadır (Nielsen 2015).



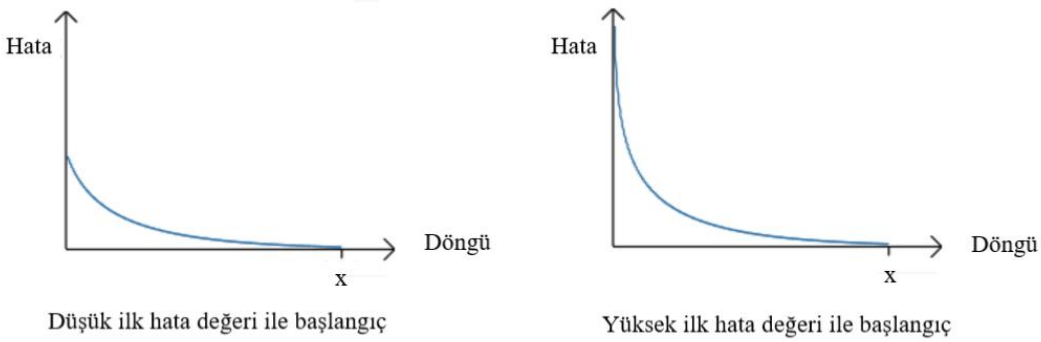
Şekil 3.11 İkinci dereceden hata fonksiyonunun hata değerine göre eğitim hızı değişimi

3.5.2 Çapraz entropi hata fonksiyonu

Çapraz entropi hata fonksiyonu, eğitim hızındaki yavaşlamaların önüne geçmek amacıyla kullanılan alternatif bir fonksiyondur. Tek nörona sahip bir ağ için; x_1, x_2, …, girdiler, w_1, w_2, …, ağılıklar ve α nöron çıktısı olarak ele alınırsa, tek nöronlu bu ağ için çapraz entropi hata fonksiyonu aşağıdaki şekilde tanımlanır (Nielsen 2015).

$$C = -\frac{1}{n}\sum_x [y \ln \alpha + (1-y)\ln(1-\alpha)] \tag{3.3}$$

Eşitlik 3.3'de, n eğitim verilerinin toplam sayısı, x tüm eğitim girdileri, y beklenen değerdir. Çıktının sıfır ile bir arasında olması nedeniyle, bu eşitlik sıfırdan büyük ve ağ çıktısının beklenen değere yakın olduğu durumlarda çapraz entropinin sıfıra yakın olduğu görülebilir (Nielsen 2015). Yani C fonksiyonu her zaman sıfırdan büyük fakat eğitim başarısı arttıkça sıfıra yaklaşan bir fonksiyondur.



Şekil 3.12 Çapraz entropi hata fonksiyonunun hata değerine göre eğitim hızı değişimi

Çapraz entropi hata fonksiyonu kullanılan ağlarda hata büyüdükçe öğrenme hızı artar. Bu nedenle ilk denemelerde yüksek hata ile karşılaşılması durumunda eğitim hızı yüksek olur (şekil 3.12). Tek bir nöron için verilen eşitlik 3.3'ün birden fazla nöron olması durumunda güncellenmesi gerekmektedir (Nielsen 2015).

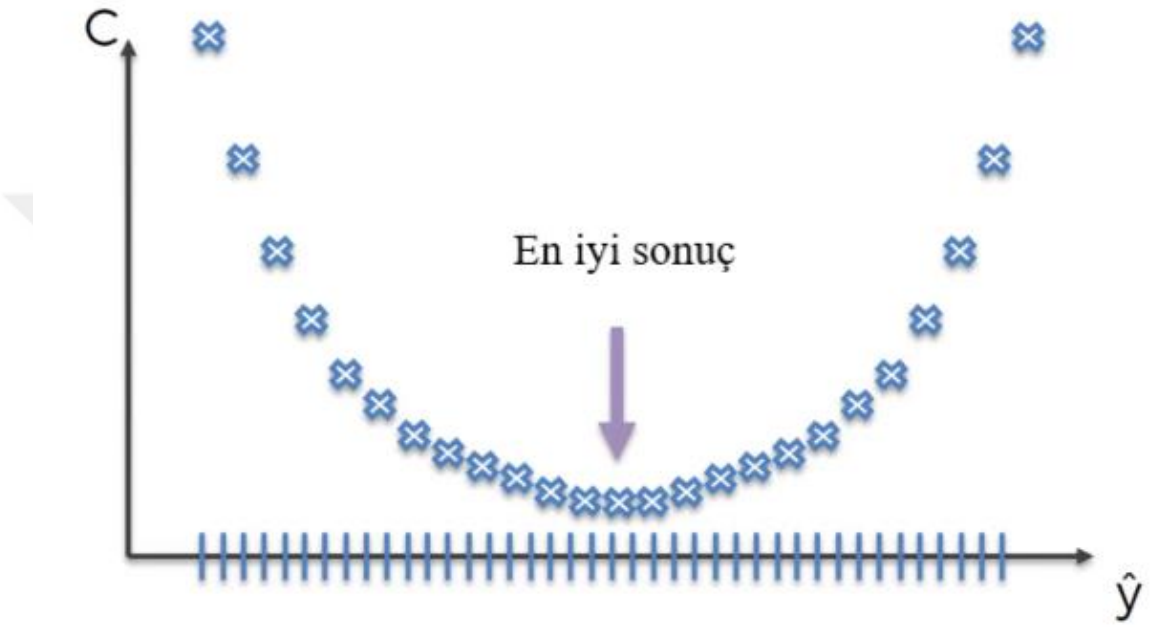
$$C = -\frac{1}{n}\sum_x \sum_j [y_j \ln \alpha_j^L + (1 - y_j) \ln(1 - \alpha_j^L)] \tag{3.4}$$

Tek nöron için kurulan eşitliğe benzer şekilde; beklenen değerler $y_1, y_2, \ldots$, çıktılar $\alpha_1^L, \alpha_2^L, \ldots$, olarak tanımlanır. Çapraz entropi hata fonksiyonu, ikinci dereceden hata fonksiyonunda daha hızlı çalışmasına rağmen hata oranı daha fazladır. Buna rağmen, çıktı nöronlarında sigmoid aktivasyon fonksiyonunun kullanıldığı ağlarda genellikle daha iyi sonuç vermektedir (Nielsen 2015).

3.6 Dereceli Azalma

Dereceli azalma, bir fonksiyonun minimumunu bulmak için birinci dereceden yinelemeli bir optimizasyon algoritmasıdır. Bu algoritma kullanılarak bir fonksiyonun yerel minimumunu bulmak için, mevcut noktadan fonksiyonun yaklaşık eğiminin negatifi ile orantılı adımlar atılır (Lotter 2018).

Örnek olarak basit yapıdaki bir ağ için hata fonksiyonunun ikinci dereceden C fonksiyonu olduğunu ve belli sayıdaki örnek için hata fonksiyonu grafiğinin şekil 3.13'deki gibi olduğunu varsayalım.

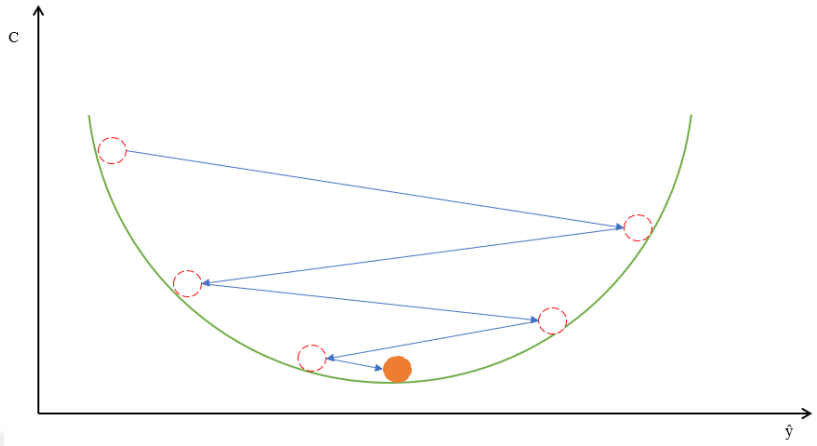


Şekil 3.13 Örnek hata fonksiyonu grafiği (Zhang 2018)

Tüm test verilerini deneyerek en iyi sonucu bulmak çok uzun zaman alacağı için, hata fonksiyonunu minimize etme amacıyla, hata fonksiyonunun eğimine göre hata fonksiyonunda en iyi değere yaklaşmaya çalışılır. Hata fonksiyonu C(W,b), hata fonksiyonunun W ve b parametrelerine göre türevleri olan, ΔW ve Δb değerlerinin sabit bir çarpan olan öğrenme katsayısı α ile çarpılarak W ve b değerlerinin azaltılması ve bunun sonucunda hata fonksiyonunun da minimum değere yaklaşması hedeflenmektedir (Nielsen 2015).

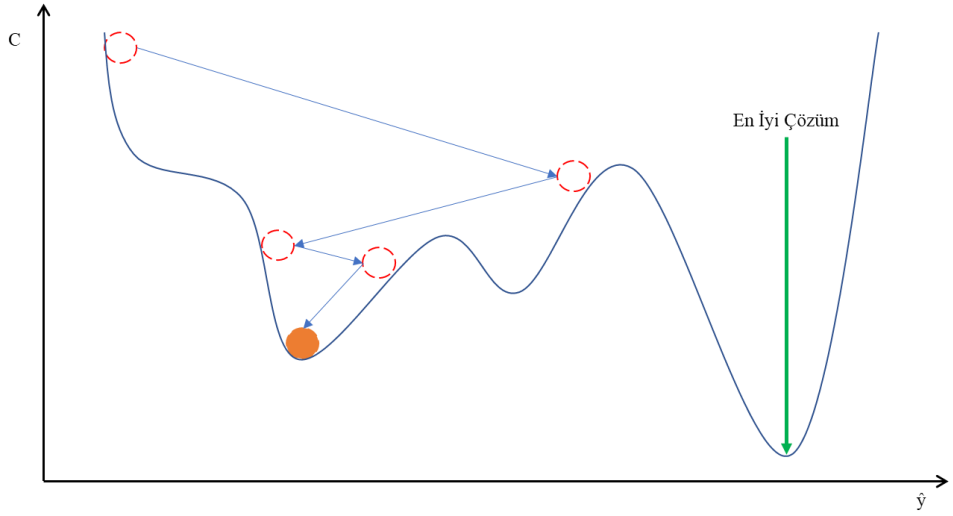
$$W_{ij} = W_{ij} - \alpha \Delta W_{ij} \tag{3.5}$$

$$b = b - \alpha \Delta b \tag{3.6}$$



Şekil 3.14 Dereceli azalma yöntemi ile en iyi hata değerine ulaşma yolu

Tüm parametreleri kontrol ederek en verimli ağ yapısını bulma işlemi ile karşılaştırıldığında dereceli azalma yöntemi oldukça verimli bir yöntemdir. Fakat bu yöntemin başarılı olması için hata fonksiyonunun dışbükey olması gerekmektedir. Şekil 3.14'de verilen hata fonksiyonu da tek yerel minimumu olan dışbükey bir fonksiyondur. Hata fonksiyonunun dışbükey olmadığı durumlarda meyilli eğilim yöntemi ile kullanılarak en iyi çözüm yerine yerel minimum bir çözüm bulunması olasılığı yüksektir (şekil 3.15).



Şekil 3.15 dereceli azalma yöntemi ile yerel minimuma ulaşma örneği (Anonymous 2018)

3.6.1 Olasılıksal Dereceli Azalma

Olasılıksal dereceli azalma, konveks kayıp fonksiyonları altında lineer sınıflandırıcıların ayırt edici öğreniminde basit ama çok etkili bir yaklaşımdır. Makine öğrenmesi alanında uzun zamandır var olsa da, büyük ölçekli verilerin kullanımlarının yaygınlaşması ile birlikte kullanım oranı artmaktadır (Bottou 2010).

Olasılıksal dereceli azalma yönteminde hata fonksiyonun türevi, dereceli azalmada kullanılan yöntemin aksine, bir seferde tüm veri üzerinden hesaplanmak yerine her adımda sadece tek bir veri kullanılarak hesaplanır. Sonrasında W ve b değerleri bu fonksiyonun türevine göre güncellenir ve bu işleme veri kümesindeki tüm veriler tamamlanana kadar devam edilir. Olasılıksal dereceli azalma yönteminde ΔW ve Δb değerleri zincir kuralına göre hesaplanır. Zincir kuralına göre eğer y fonksiyonu x değişkenine göre türevi alınabilen bir fonksiyon ise türev alınabilir.

Olasılıksal dereceli azalma algoritmasının bir avantajı da, rastgele olması nedeniyle, aynı başlangıç ağırlıkları ile her denemede farklı değerlerden geçerek sonuca ulaşmasıdır. Dereceli azalma yöntemi başlangıç ağırlıkları değişmediği sürece aynı yolu takip ederek aynı sonucu verecektir.

Olasılıksal dereceli azalma ile bir yapay sinir ağı eğitimi yedi adımda özetlenebilir.

1. Sıfıra yakın (fakat sıfır olmayacak şekilde) ilk ağırlık değerlerinin verilmesi,

2. Veri kümesindeki ilk satırının girilmesi ve sonuçların alınması.

3. Ağdaki her nöronda bir önceki katmandan gelen verinin işlenerek çıktının oluşturulması.

4. Çıktı ile gerçek değerin karşılaştırılması ve hatanın hesaplanması.

5. Geri yayılım algoritması kullanılarak ağırlıkların hataya göre güncellenmesi.

6. 1. maddeden 5. maddeye kadar olan sürecin tüm veri kümesi tamamlanana kadar devam ettirilmesi.

7. Tüm veri kümesi için çıktıların alınma ve ağırlıkların güncellenmesi işlemi bittikten sonra yeni bir döngüye başlanılması.

3.6.2 Dereceli azalma optimizasyon algoritmaları

Momentum: Olasılıksal dereceli azalmanın eğimin fazla olduğu noktalarda salınım yapması nedeniyle optimum değere ulaşmada gecikmeler yaşamaktadır. Momentum ise bu gibi durumlarda olasılıksal dereceli azalmanın hızlanmasına yardımcı olan bir algoritmadır (Qian 1999).

Adagrad: Adagrad, sık rastlanan özelliklere ait parametrelere daha küçük öğrenme oranı ile güncellemeler uygulayan, az rastlanan özelliklere ait parametrelere büyük öğrenme oranı ile güncellemeler uygulayan bir algoritmadır. Bu nedenle az veri olan çalışmalarda uygulanması daha uygundur (Duchi vd. 2011). Adagrad'da her parametrenin kendi öğrenme hızı vardır ve algoritmanın özelliklerine bağlı olarak öğrenme oranı giderek azalmaktadır. Bu nedenle öğrenme oranı giderek azalır ve zamanın bir noktasında sistem öğrenmeyi bırakır.

Adadelta: Adagrad'ın bir uzantısı olan bu yöntem, geçmiş gradyan kareleri değerlerini depolamak yerine bunları sabit bir w değerine eşitler. Bu sayede adagrad'da karşılaşılan öğrenme oranının azalması problemini önler (Zeiler 2012).

RMSProp: Adadelta algoritması ile yakın zamanlarda farklı ekipler tarafından geliştirilmiş bir algoritmadır. Adagrad'daki eğitim hızı azalması problemine çözüm olarak geliştirilmiştir (Ruder 2016).

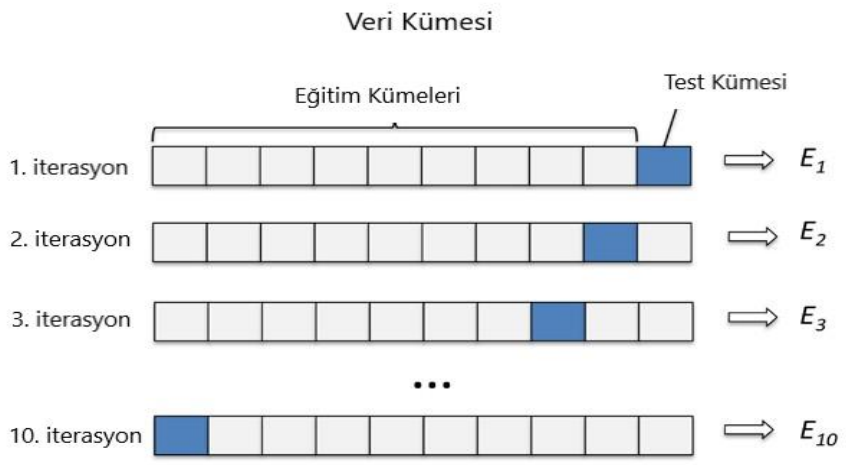
Adam: Adam algoritması her adımda; Adadelta ve RMSProp algoritmalarına benzer şekilde geçmiş gradyanların karelerinin ortalamalarının yanında momentum yöntemindeki gibi geçmiş gradyanların ortalamasını da depolar (Kingma ve Ba 2015). Yani bu algoritma momentum ve RMSProp algoritmalarının birleşimi gibidir.

3.7 Geri Yayılım Algoritması

Geri yayılım algoritması yapay sinir ağlarında çok katmanlı ileri beslemeli ağlar için denetimli bir öğrenme yöntemidir. Daha önce bahsedildiği gibi girdi ve çıktıların işlenmesi ile bulunan hataların geri yayılımı ve ağırlıkların bu bilgilere göre güncellenmesi, ağ eğitme sürecinin bir parçasıdır. Geri yayılım algoritmasının temel işlevi, hata fonksiyonu geri bildirimi ile tüm ağ ağırlıklarının paralel olarak güncellenmesini mümkün kılmaktadır (Rojas 1996). Bu şekilde toplam hata değeri minimize edilebilmektedir.

3.8 K Katmanlı Çapraz Doğrulama

Tasarlanan YSA modeli başarısının sınanması ve ortalama bir değerin belirlenmesi amacıyla kullanılır. K katmanlı çapraz doğrulama yönteminde öncelikle bir k değeri belirlenerek bu k değerine göre veri kümesi eşit parçalara bölünür. Örnek olarak 300 verinin bulunduğu bir kümede k değeri 10 seçilirse, veri kümesi 30'ar verilik 10 eşit parçaya bölünür (şekil 3.16). Bu parçalardan bir tanesi test kümesi olarak ayrılırken diğer kümeler eğitim amaçlı kullanılır (Kale vd. 2011). Tüm parçalar bir kere test kümesi olacak şekilde işlem tekrarlanır.



Şekil 3.16 k katmanlı çapraz doğrulamada k=10 için eğitim ve test setlerinin ayrılması (Rosaen 2016)

Tüm döngünün tamamlanmasının ardından elde edilen doğruluk değerlerinin aritmetik ortalaması alınarak genel bir değer elde edilir.

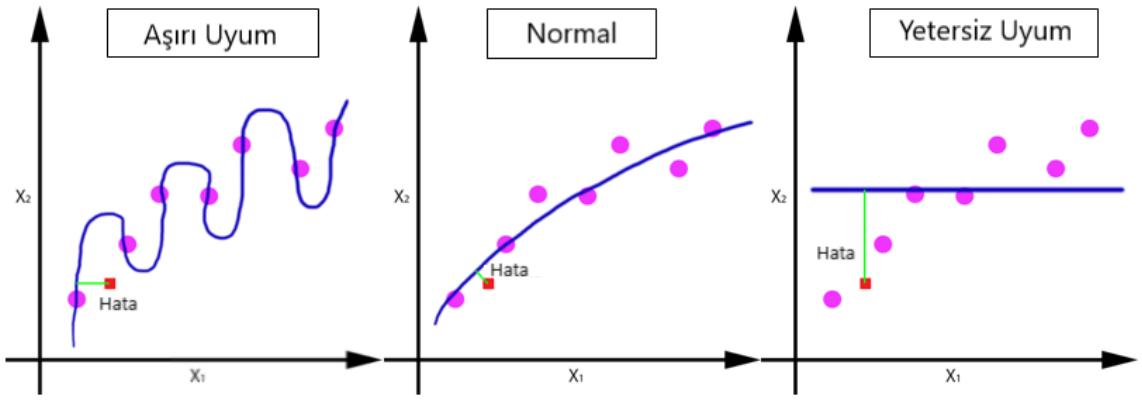
$$\text{Başarı} = \frac{\sum_{i=1}^{k} E_i}{k} \tag{3.7}$$

3.9 Eğitim Performansını Yükseltme Yöntemleri

Yapay sinir ağları eğitimleri sırasında karşılaşılan en yaygın problem aşırı uyum (ing. Overfitting) problemidir. Yeteri sayıda döngü ile ağ eğitildikten sonra hata makul seviyelere geriler fakat ağın test verileri ile beslenmesi sonrası çok yüksek hata değerleri ile karşılaşılır. Bu durum ağın eğitim verilerini ezberlediğini ve ona uygun değerler için ağırlıkları güncellediği ama genelleme yapabilecek yeteneğe ulaşmadığını gösterir. Şekil 3.17'de başarılı ve başarısız eğitim sonuçları görülmektedir.

Genelleme, yapay sinir ağlarının öğrenme aşamasında öğrenilenlerin eğitim verilerinde olmayan örneklere ne kadar iyi uygulandığını ifade eder. İyi bir eğitimin amacı, eğitim verilerinden problem alanındaki tüm verilere göre genelleme yapabilmektir. Bu, eğitim kümelerinde görülmeyen veriler üzerinden tahminler yapılabilmesini sağlar.

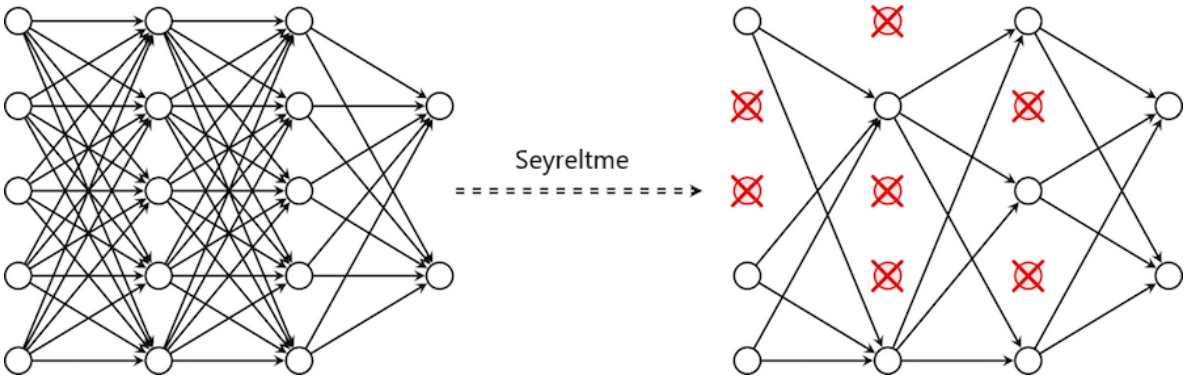
Aşırı uyumu önlemek için ağ tasarımını optimum uyum sağlayacak şekilde tasarlanması gerekmektedir. Ağın çok büyük olması ve eğitim setinin yetersiz olması durumunda aşırı uyum problemi ile karşılaşılabilir. Tersi durumlarda da, yani ağın çok küçük olması durumunda yetersiz uyum problemi ile karşılaşılır. Yetersiz uyum durumunda ağın, eğitim verilerini modelleyememesi ve yeni verileri genelleyememesi problemi görülür. Bu durum, çıktı hataları değerlendirildiğinde, tespit edilmesinin kolay olması nedeniyle tartışılan bir durum değildir.



Şekil 3.17 Aşırı uyum - normal - yetersiz uyum örnek grafikleri (Arora 2016)

3.9.1 Seyreltme düzenleme yöntemi

Seyreltme düzenleme yöntemi (ing. Dropout regularization) aşırı uyum durumlarına karşı kullanılan bir tekniktir. Seyreltme düzenleme yöntemi, eğitim sırasında tasarlanmış derin ağ modelinin uygulandığı katmandan rastgele nöronları çıkarır. Daha sonrasında girdiler yeni ağ üzerinde ilerletilir ve hata elde edilerek geri yayılım algoritması asıl ağa göre parametre güncellemelerini sadece çıkarılmamış nöronlar üzerinde uygular. Böylece aşırı uyum sorununun önlenmesi ve eğitim sürecinde hızlanma sağlanır. Her parametre güncelleme esnasında farklı nöronlar çıkarılır ve komşu nöronların birbirlerine çok fazla bağımlı olmalarının önüne geçilmiş olur (şekil 3.18). Aşırı uyum sorunu olmayan bazı sınıflandırma işlemlerinde de performansı arttırdığı gözlemlenmiştir.



Şekil 3.18 Örnek seyreltme düzenleme yöntemi (Aoullay 2018)

3.9.2 Örgü arama yöntemi ile parametre ayarlama

Yapay sinir ağları modelinde, eğitim süreci boyunca model ağda öğrenilen ve güncellenen parametreler (ağırlıklar) ve ağın modeli tasarlanırken belirlenen ve sabit olan parametreler (nöron sayısı, döngü sayısı gizli katman sayısı vb.) olmak üzere iki parametre tipi vardır. Yapay sinir ağları modellerinde kullanılan sabit parametreler, hiper parametre olarak adlandırılır. K katmanlı çapraz doğrulama kullanılarak istenilen hiper parametre değerleri test edilerek en yüksek başarıya sahip ağ yapısı tespit edilmeye çalışılır.

4. KAYNAK ÖZETLERİ

Telekomünikasyon sistemlerinde sahtekarlık tespiti ve önlenmesi, sistemlerin karmaşık yapısı ve sahtekarlık yöntemlerinin dinamik olması nedeniyle çok zordur. Klasik sahtekarlık tespit yöntemlerinin teknolojilere paralel olarak gelişmesi ve yeni yöntemlere adapte olması gerekmektedir.

IP üzerinden ses servisinin kötüye kullanımını yakalamak için yapılan bir çalışmada, IP detay kayıtlarının (IPDK) kullanılarak kural dizeleri oluşturulması çalışılmıştır. Çalışma sonucunda IPDK kullanılarak uygun bir sistemin geliştirilebileceği ve başarılı bir şekilde tespit yapılabileceği gösterilmektedir (Ruiz-Agundez vd. 2010). Fawcett ve Provost (1997) çalışmalarında geniş bir veri tabanında veri madenciliği yöntemi ile kullanıcı profili oluşturmuşlardır. Sistem sahtekarlık çağrılarının tespitinde kullanacağı kural dizelerini bu profilleri kullanarak oluşturmakta ve uygulamaktadır. Bu şekilde yeni sahtekarlık yöntemlerine adapte olabilmektedir.

Sahtekarlık tespiti için kullanıcı profili oluşturularak normal kullanımın dışındaki kullanımların tespit edilmesi yoğun olarak kullanılan bir yöntemdir (Hilas vd. 2015, Burge ve Shawe-Taylor 2001, Hilas ve Sahalos 2005).

Jain (2017), veri akış analitiği ve veri madenciliğine dayalı yapay zeka sınıflandırması kullanmıştır.

Taniguchi vd. (1998) daha iyi sonuçlar almak için karma sistem kullanmıştır. İlk olarak gözetimli öğrenme ile eğitilmiş bir YSA kullanarak kullanıcıları sınıflandırmıştır. Daha sonra Gauss karışım modeli kullanarak kullanıcıların geçmiş kullanım istatistiklerine göre kullanıcı profillerini oluşturmuştur. Son olarak da belirli bir kullanıcı için farklı sahtekarlık senaryolarının tanımlanması için bayes ağları kullanmıştır.

Gerçek zamanlı veri kullanılarak sahtekarlık tespiti yapmak araştırmacılar için zorlayıcı bir konudur. Manunza vd. (2017) gerçek zamanlı veri tespiti için tasarım ve uygulama önerileri sunmuşlardır. Kerberos adını verdikleri sistem IMS altyapısı üzerinden verilen ses

servislerinde kullanılabilmektedir. Sistemi beslemek için IMS çekirdek sistemlerinin bir özellik olarak sunduğu çevrimiçi ÇDK'ler kullanılmaktadır. Bu ÇDK'lerde başlangıçtan bitişine kadar tüm çağrı aşamaları ile ilgili detaylar bulunmaktadır. Sistem üzerinde tanımlanmış kural dizeleri çevrimiçi ÇDK sunucusundan gelen verilere göre çağrıları işleyerek, kurallara takılan çağrılar için operatör ekranına uyarı göndermektedir (Manunza vd. 2017).

5. MATERYAL ve YÖNTEM

5.1 Çalışmada Kullanılan Veri Seti

Bu çalışmada bir telekomünikasyon şirketi abonelerinin verileri maskelenip kodlanarak kullanılmıştır. Veri seti; abonelere ait arama detay kayıt bilgileri ya da OBP sinyalleşmesi davet mesajları içerikleri, ödeme istatistikleri ve sosyal verileri kullanılmıştır. Alınan veriler sahtekarlık tespitinde kullanılabilirliklerine göre değerlendirilerek ayrıştırılmıştır.

5.1.1 Arama detay kayıtları kullanılarak hazırlanan veri seti

Veri seti içerisinde abonelere ait arama detay bilgileri, ödeme istatistikleri ve sosyal verileri kullanılmıştır. Alınan veriler sahtekarlık tespitinde kullanılabilirliklerine göre değerlendirilerek, çizelge 5.1'de belirtilen 17 alanı içerecek bir veri seti kullanılmıştır. Çizelge 5.2'de bu alanların alabileceği örnek değerler görülmektedir.

Çizelge 5.1 Birinci veri setinde kullanılan alanlar ve açıklamaları

Başlık	Açıklama
Arama Yönü Tarifesi	Aranan numara için uygulanan kuruş cinsinden dakika ücretleri
Çağrı Süresi	Toplam konuşma süresi
Konuşma ücreti	Konuşma için abone faturasına yansıtılan bedel
Çağı Yönlendirme Kullanımı	Çağrı yönlendirme özelliğinin kullanım durumu (1 aktif, 0 aktif değil)
Ücret Durum	Çağrının ücret durumu (1 ücretli, 0 ücretsiz)
Cevaplanma Durum	Çağrının cevaplanma durumu (1 cevaplı, 0 cevapsız)
Abonelik Yaşı	Abonelik başlangıç tarihinden itibaren gün cinsinden geçen süre
Abone Cinsiyet	Abone cinsiyeti (1 erkek, 0 kadın)
Aylık Ortalama Fatura Bedeli	Son 2 ay hariç abone ortalama fatura bedeli
Ödenmiş Fatura Sayısı	Abonelik süresince abonenin yaptığı ödeme sayısı
Çağrı Saat Dilimi	Arama başlangıç saat dilimi
Kodlanmış Arayan Numara	Abone telefon numarası bilgisi (bilgi güvenliği nedeniyle kodlanmış olarak alınmıştır.)
Alan Kodu	Abone il bilgisini barındıran il alan kodu

Çizelge 5.1 Birinci veri setinde kullanılan alanlar ve açıklamaları (devam)

Abonelik Türü	Abonelik türü
HT	Abone dilimi
AHT	Abone alt dilimi
Sahtekarlık Durum	Sahtekarlık durumu (1 sahtekarlık çağrısı, 0 normal çağrı)

Çizelge 5.2 Birinci veri seti örnek değerler

Başlık	Arama 1	Arama 2	Arama 3
Arama Yönü Tarifesi	50	21	1750
Çağrı Süresi	727	18000	19167
Konuşma ücreti	0.605833333	3.15	559.0375
Çağı Yönlendirme Kullanımı	0	0	0
Ücret Durum	1	1	1
Cevaplanma Durum	1	1	1
Abonelik Yaşı	1465	3460	82
Abone Cinsiyet	1	1	1
Aylık Ortalama Fatura Bedeli	110	75	78
Ödenmiş Fatura Sayısı	47	114	1
Çağrı Saat Dilimi	14	16	10
Kodlanmış Arayan Numara	11210D0B6	F42BEEAE	8EE484BE
Alan Kodu	462	412	242
Abonelik Türü	20100	60209	10100
HT	PSTN	PSTN	PSTN
AHT	PSTN	ANK	PSTN
Sahtekarlık Durum	0	0	1

5.1.2 OBP sinyalleşmesi davet mesajları içerikleri kullanılarak hazırlanan veri seti

Veri seti içerisinde abonelere ait OBP sinyalleşmesi davet mesajları içerikleri, ödeme istatistikleri ve sosyal verileri kullanılmıştır. Alınan veriler sahtekarlık tespitinde kullanılabilirliklerine göre değerlendirilerek, çizelge 5.3'te belirtilen 12 alanı içerecek bir veri seti kullanılmıştır. Çizelge 5.4'te bu alanların alabileceği örnek değerler görülmektedir.

Çizelge 5.3 İkinci veri setinde kullanılan alanlar ve açıklamaları

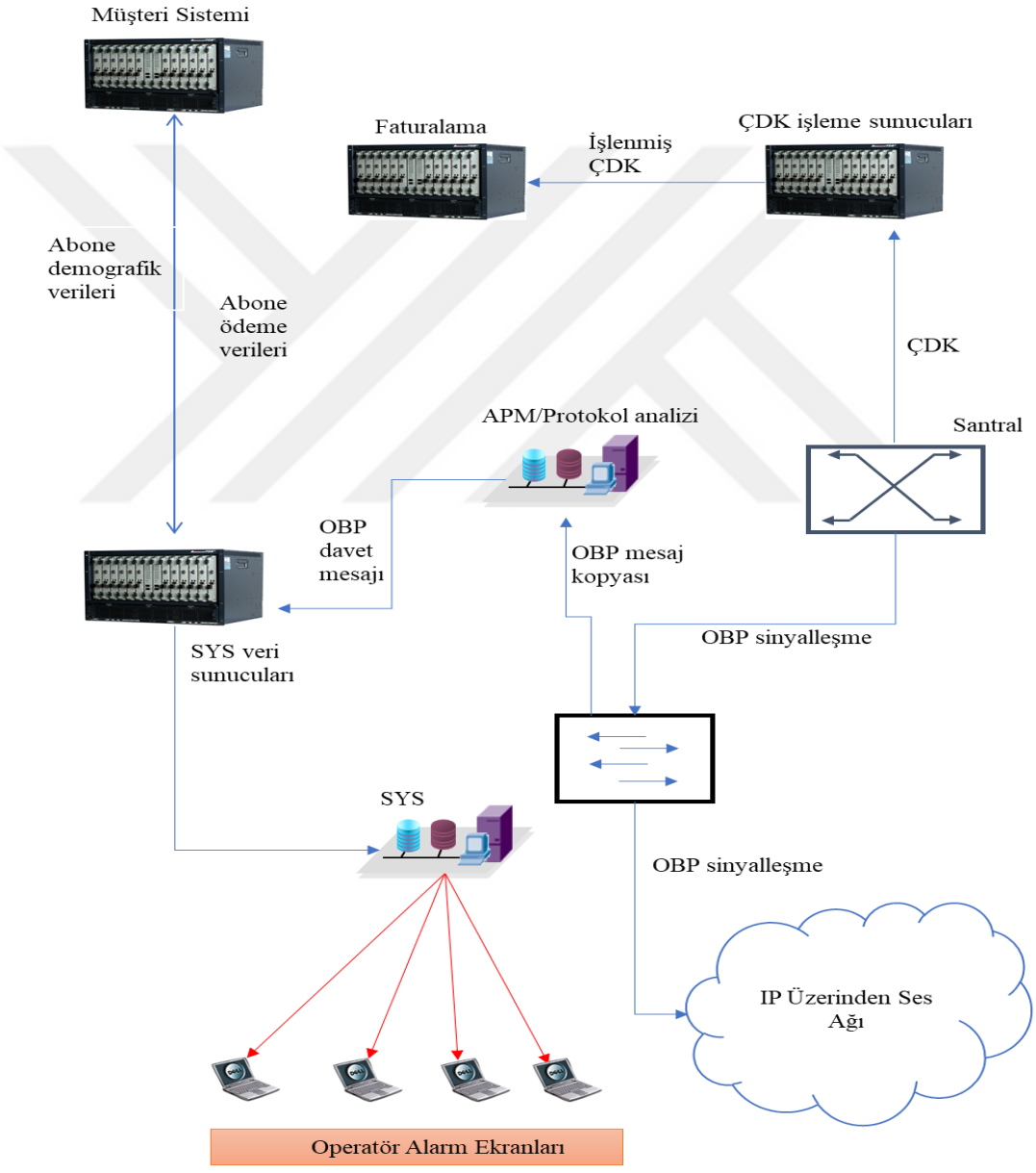
Başlık	Açıklama
Arama Yönü Tarifesi	Aranan numara için uygulanan kuruş cinsinden dakika ücretleri
Abonelik Yaşı	Abonelik başlangıç tarihinden itibaren gün cinsinden geçen süre
Abone Cinsiyet	Abone cinsiyeti (1 erkek, 0 kadın)
Aylık Ortalama Fatura Bedeli	Son 2 ay hariç abone ortalama fatura bedeli
Ödenmiş Fatura Sayısı	Abonelik süresince abonenin yaptığı ödeme sayısı
Çağrı Saat Dilimi	Arama başlangıç saat dilimi
Kodlanmış Arayan Numara	Abone telefon numarası bilgisi (bilgi güvenliği nedeniyle kodlanmış olarak alınmıştır.)
Alan Kodu	Abone il bilgisini barındıran il alan kodu
Abonelik Türü	Abonelik türü
HT	Abone dilimi
AHT	Abone alt dilimi
Sahtekarlık Durum	Sahtekarlık durumu (1 sahtekarlık çağrısı, 0 normal çağrı)

Çizelge 5.4 İkinci veri seti örnek değerler

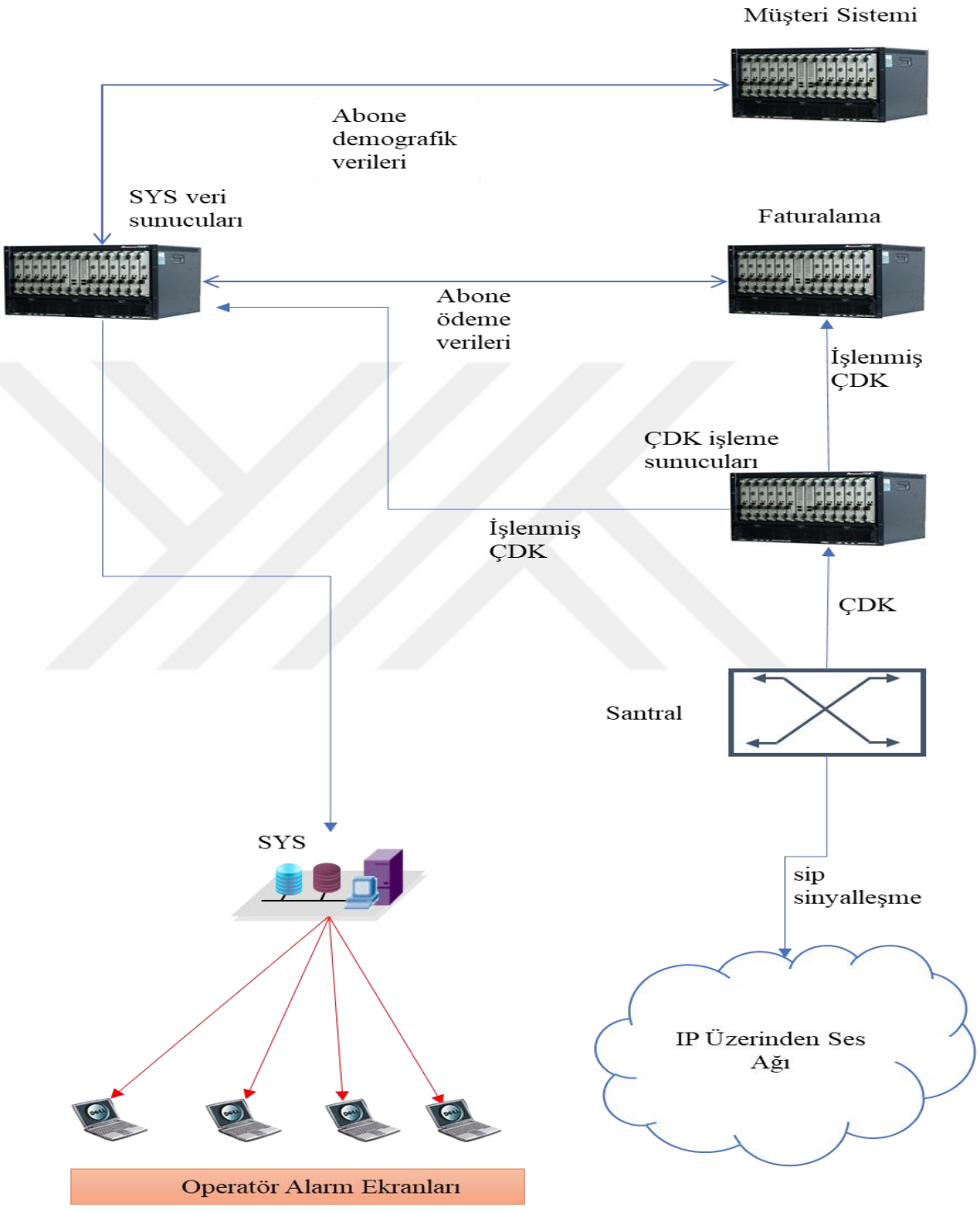
Başlık	Arama 1	Arama 2	Arama 3
Arama Yönü Tarifesi	50	21	1750
Abonelik Yaşı	1465	3460	82
Abone Cinsiyet	1	1	1
Aylık Ortalama Fatura Bedeli	110	75	78
Ödenmiş Fatura Sayısı	47	114	1
Çağrı Saat Dilimi	14	16	10
Kodlanmış Arayan Numara	11210D0B6	F42BEEAE	8EE484BE
Alan Kodu	462	412	242
Abonelik Türü	20100	60209	10100
HT	PSTN	PSTN	PSTN
AHT	PSTN	ANK	PSTN
Sahtekarlık Durum	0	0	1

5.2 Örnek Mimari Tasarımlar

Sistemin verimli çalışması ve çağrıların değerlendirilebilmesi için uygun mimari tasarımların sağlanması gerekmektedir. Şekil 5.1'de ÇDK verileri ile, şekil 5.2'de OBP davet mesajları ile sahtekarlık tespiti yapılabilmesi için mimariler görülmektedir.



Şekil 5.1 ÇDK verileri kullanılarak sahtekarlık çağrılarının tespit edilmesi için örnek mimari



Şekil 5.2 OBP davet mesajları kullanılarak sahtekarlık çağrılarının tespit edilmesi için örnek mimari

5.3 Kullanılan Programlama Dili

Bu çalışmada, kolay programlanabilme ve çevrimiçi toplulukların geniş olması nedeniyle yapay zeka uygulamalarında en çok tercih edilen dil olan Python kullanılmıştır. Python; 1990 yılında geliştirilmeye başlanan, kullanımı kolay, harici bir derleyici ihtiyacı bulunmayan, nesne yönelimli, yapısal ya da fonksiyonel programlama gibi birden fazla programlama paradigmasını destekleyen bir programlama dilidir. Birçok derin öğrenme kütüphanesinin arka planında kullanılmasından dolayı yapay zeka uygulamalarında en çok tercih edilen programlama dilidir.

5.4 Kullanılan Kütüphaneler

5.4.1 TensorFlow

Tensorflow; açık kaynak kodlu, sayısal hesaplama için veri akış grafiklerini kullanan bir derin öğrenme kütüphanesidir. Tensorflow, makine öğrenimi ve derin sinir ağları araştırması yürütmek amacıyla Google'ın beyin takımı tarafından geliştirilmiştir. Esnek mimarisi, hesaplamayı tek bir API ile bir masaüstü, sunucu veya mobil cihazdaki bir veya daha fazla MİB'ye veya GİU'ya dağıtılmasına olanak tanır (Anonymous, Wikipedia 2018).

Tensorflow içerisinde data n-boyutlu bir dizi olarak sunulur. Bu n- boyutlu dizilere de tensör adı verilir. Grafikler, bu veriler (tensör) ve matematiksel işlemlerden oluşur.

5.4.2 Keras

Keras, derin öğrenme için yazılmış bir Python kütüphanesidir. Yine sembolik işlem temelli ve derin öğrenme için de kullanılan Theano ya da TensorFlow kütüphaneleri üzerinden çalışır. GPU ya da CPU üzerinde çalışmasını bu temel kütüphaneler üzerinden sağlar. Daha üst düzey bir kütüphane olduğundan Theano ya da TensorFlow a göre daha kolay uygulama yapılabilmektedir (Anonymous 2018). Bu çalışmada TensorFlow Kütüphanesi üzerinden çalışan Keras Kütüphanesi ile çalışılmıştır.

5.4.3 Numpy

NumPy, Python' da bilimsel hesaplamalarda kullanılan temel pakettir. Çok boyutlu diziler (array), çeşitli türetilmiş nesneler ve pek çok matematiksel, mantıksal, şekil manipülasyonu, sıralama, seçme, ayrık Fourier de dahil olmak üzere, diziler üzerinde hızlı işlemler yapılmasını sağlayan Python kütüphanesidir (Rao 2018).

5.4.4 Scikit-learn

Scikit-learn yapay öğrenme alanında en yaygın olarak kullanılan kütüphanelerden biridir. Bu kütüphane; doğrusal regresyon, lojistik regresyon, karar ağaçları, rastgele orman gibi birçok temel yöntemi içermektedir. İhtiyaç duyulan temel yöntemlerin büyük bir kısmını içermesi ve veri analitiği uygulamalarının baştan sona yürütülmesine imkan sağlaması, yoğun olarak tercih edilmesinin en önemli sebeplerinden sayılabilir (Rao 2018).

5.4.5 Pandas

Pandas; Python programlama dili için geliştirilmiş açık kaynaklı, kullanımı kolay, yüksek performanslı bir veri yapısı ve analizi kütüphanesidir. Pandas dağıtık işlemeye uygun değildir. Bu sebeple işlenecek verinin büyüklüğü, makinenin kapasitesiyle sınırlıdır (Rao 2018).

5.5 Veri Setinin Düzenlenmesi

Mevcut veri setlerinin algoritma içerisinde işlenebilmesi için bazı ön işlemlere tabi tutulması gerekmektedir. Veri dosyalarının okunmasının ardından; girdi ve çıktı kümelerinin oluşturulması, etiketleme ve kodlama işlemlerinin yapılması, eğitim ve test verilerinin belirlenmesi, ölçekleme yapılması gibi işlemlerden geçmesi gerekmektedir.

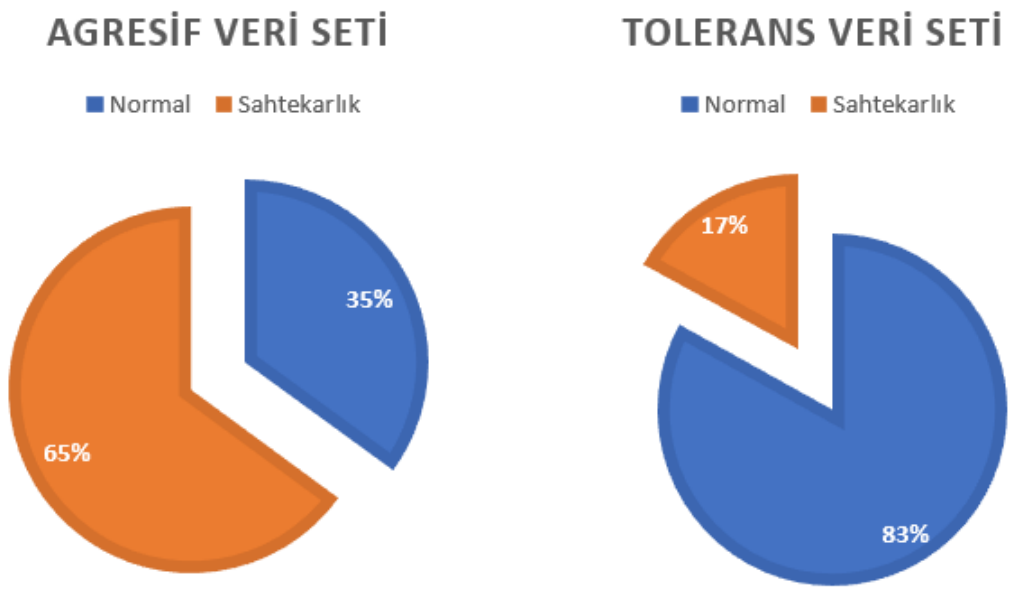
Çalışma ağ eğitimlerinde iki farklı eğitim seti kullanılmıştır. Agresif veri setinde, sahtekarlık çağrılarının gözden kaçmaması için daha yoğun sahtekarlık çağrısının bulunmaktadır. Bu veri seti ile eğitilen ağda normal çağrıların da sahtekarlık çağrısı olarak etiketlenmesi ihtimalini dolayısıyla operatör yükünü arttırması beklenmektedir. Tolerans veri setinde ise

normal çağrılar yoğun olarak bulunmaktadır. Bu veri seti ile eğitilen ağın daha az çağrıyı sahtekarlık çağrısı olarak etiketlemesi beklenmektedir.

5.5.1 Sınıf dengesizliğinin kontrol edilmesi

Sahtekarlık çağrılarını da içeren ve çok fazla çağrı bulunan veri setinin incelenmesi sonrası, sınıf dengesizliği yaşanma ihtimalinin yüksek olduğu tespit edildi. Bu durumdan kaçınmak için benzer çağrı senaryoları tespit edilerek üzerinde çalışılacak veri setinde yer almamaları için çalışma yapıldı.

Son haline getirilmiş veri setinde sahtekarlık çağrılarının oranı; agresif veri setinde %65, tolerans veri setinde %17 olmuştur. Şekil 5.1'de her iki veri setinin de dağılımları görülmektedir.



Şekil 5.3 Çağrı dağılımları

5.5.2 Girdi ve Çıktı Kümelerinin Oluşturulması

Veri setinin uygulamaya aktarılması sonrasında, ağ eğitiminde kullanılacak bağımsız değişkenleri içeren girdi matrisinin oluşturulması gerekmektedir. Bu matris her bir satır için

bağımlı değişkenin hesaplanabilmesi için bir özellik içermektedir. Uygulamada G olarak adlandırılmıştır. Girdi matrisi, ÇDK ile üretilen veri setinin kullanılması durumunda 17 sütun, OBP davet mesajları ile oluşturulan veri setinde ise 11 sütun içermektedir.

Tek bağımlı değişken olan sahtekarlık durumu sütunu ise C çıktı matrisine aktarılacaktır.

5.5.3 Kategorik Değişkenlerin Ele Alınması

Kategorik veya sürekli olmayan bir değişken, iki veya daha fazla kategoriye sahip olan bir değişkendir. Nominal ve sıralı (ing. ordinal) olmak üzere iki tür kategorik değişken vardır. Nominal değişkenin kendi kategorileri içinde bir sıralaması yoktur. Örneğin, cinsiyet kendi içinde bir sıralaması olmayan iki kategoriye (erkek ve kadın) sahip kategorik bir değişkendir. Sıralı değişkenin aldığı değerler arasında net bir sıralama vardır. Örnek olarak; düşük, orta ve yüksek gibi üç sıralı kategoriye sahip bir sıcaklık değişkeni verilebilir.

Kategorik değişkenlerin algoritma içerisinde işlenebilmesi için sayısal değerlere dönüştürülmesi (sayısal olarak etiketlenmesi) gerekmektedir. Daha sonra ikiden fazla değere sahip nominal değişkenlerin tespit edilerek gölge değişken ile kodlanması gerekmektedir. Mevcut veri setinde bu işlem çizelge 5–5'te belirtilen değerler için sayısal olarak yapılmışitır.

Çizelge 5.5 Çağrı detay kayıtlarından tekrarlayan verilerin çıkarılması ile üretilen veri seti kategorik değişkenleri

Kategorik değişken	Değişken sayısı
Saat Dilimi	24
Alan Kodu	25
Arayan Numara	48
Abonelik Türü	9
Hizmet Türü	2
Alt Hizmet Türü	4
Tarife	8

Çizelge 5.5'de de görülebileceği şekilde gölge değişken atama işleminin beş kategori için uygulanması yeterli olacaktır.

5.5.4 Eğitim ve test setlerinin belirlenmesi

Belirlediğimiz girdi ve çıktı matrislerinin scikit-learn kütüphanesi kullanarak rastgele ayrılması mümkündür. Eğitim ve test verilerinin oranlarının %70'e %30 olacak şekilde ayrılma işlemi sonrası sınıf dengesizliği kontrolleri de yapılmıştır.

5.5.5 Ölçeklendirme

Makine öğrenmesi çalışmalarında veri ölçeklendirmesinin yapılması çok önemlidir. Algoritmanın çalışması sırasında pek çok hassas hesaplama yapılmakta ve buna göre geri yayılım algoritmaları kullanılarak sistem verileri güncellenmektedir. Hem bu hesaplamaların daha basite indirgenmesi hem de belli bir bağımsız değişkenin diğer bağımsız değişkenlerden baskın olmasını önlemek için ölçeklendirme yapılması gerekmektedir. Bu çalışmada ölçeklendirme için Scikit-learn kütüphanesi kullanıldı.

Farklı ölçeklendirme yöntemlerinin farklı sonuçları olması nedeniyle, uygun yöntem seçimi için çeşitli varyasyonlar denenmiştir.

Standart Ölçeklendirme: Ortalamayı kaldırır ve verileri birim varyansa ölçeklendirir. Fakat, aykırı değerler (çok yüksek ya da çok düşük) ölçeklendirilmiş değer aralığını düşüren bir etkiye sahiptir.

Asgari-Azami Ölçeklendirme: Tüm değerleri 0-1 aralığında olacak şekilde yeniden ölçeklendirir. Bu çalışmada da bu ölçeklendirme yöntemi tercih edilmiştir.

5.6 Ağ Modeli Oluşturma

Veri setinin algoritma tarafından işlenebilecek şekilde düzenlenmesinin ardından girdi matrisi özellik sayılarına göre giriş katmanı nöron sayısı belirlenir. Bu işlem sonucu en iyi sonucu verecek ağ tasarımını deneme yanılma yolu ile bulunabilir. Bu çalışmada en iyi

performans gösterecek ağ tasarımı için hem ÇDK hem de OBP davet mesajları ile oluşturulan veri setleri için ayrı tasarımlar ve iyileştirmeler yapılması gerekmektedir.

Veri setlerindeki bazı özellikler için gölge değişkenler atanması nedeniyle giriş katmanları fazla sayıda nörondan oluşacaktır. Gizli katmanlardaki nöron sayılarının belirlenirken verilerin gruplanması da dikkate alınmalıdır. Çok fazla gölge değişken, gizli katmanlarda gereğinden fazla nöron ile tasarım yapılmasına neden olabilir.

5.6.1 Giriş katmanını ve ilk gizli katmanın oluşturması

Veri seti içerisinde yer alan; saat dilimi, kodlanmış arayan numara, alan kodu, abonelik türü ve alt dilim özellikleri, gölge değişkenler ile güncellendiği için giriş katmanı nöron sayılarını arttırmışlardır.

Keras kütüphanesinde ilk gizli katman tanımı, giriş katmanı tanımları ile beraber yapılmaktadır. Yapılan deneyler ve sonuçları değerlendirildiğinde aktivasyon fonksiyonunun doğrultulmuş doğrusal ünite (ing. rectified linear unit, relu) ve başlangıç fonksiyonunun da doğrusal olmasına karar verilmiştir.

5.6.2 İkinci ve üçüncü gizli katmanların oluşturulması

Çalışma kapsamında yapılan testler sonucunda ikinci ve üçüncü gizli katman varyasyonları denenmiştir. Birinci veri seti ile en verimli olarak çalışan varyasyonda; ikinci gizli katmanın 20 nörondan, üçüncü gizli katmanın 12 nörondan oluşmasına karar verilmiştir. İkinci veri seti ile en verimli olarak çalışan varyasyonda; ikinci gizli katmanın 20 nörondan, üçüncü gizli katmanın 20 nörondan oluşmasına karar verilmiştir.

5.6.3 Çıktı katmanının oluşturulması

Çıktı katmanında sahtekarlık durumunun belirlenmesi için tek nöron kullanılacaktır. Nöron çıktısına göre sahtekarlık olasılığının belirlenmesinde her iki veri seti için de en uygun aktivasyon fonksiyonunun sigmoid fonksiyonu olduğu görülmüştür.

6. DENEYSEL ÇALIŞMA

6.1 Ağ Parametrelerinin Seçimi

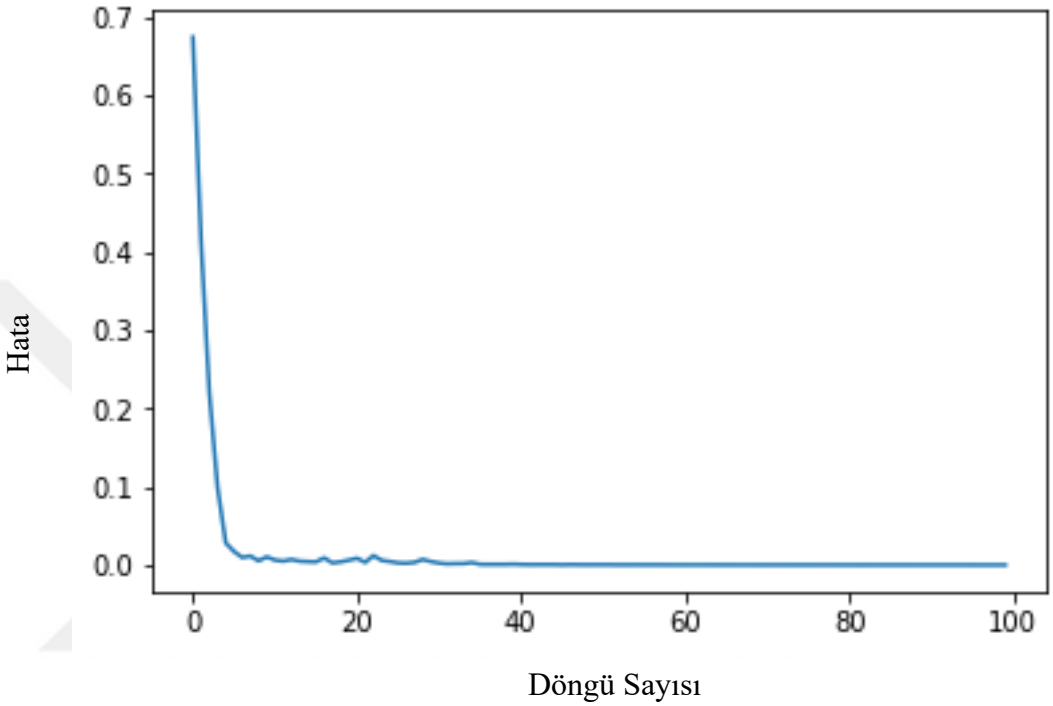
Derin öğrenme algoritmalarında en başarılı ağ belirlenirken deneme yanılma yoluyla pek çok deneme yapılması ve sonuçlarının karşılaştırılması gerekmektedir. Bu çalışma kapsamında ağ tasarlanırken; katmanlardaki nöron sayıları, aktivasyon fonksiyonları, yanlılık (ing. bias) başlangıç fonksiyonu, çekirdek (ing. kernel) başlangıç fonksiyonu, hata fonksiyonu, optimizasyon yöntemleri ve öğrenme oranı parametreleri k katmanlı çapraz doğrulama yöntemi ile tüm kombinasyonları denenerek en başarılı ağ tasarımının bulunmasına çalışılmıştır. Çapraz denemeler çizelge 6.1'deki değerler için yapılmıştır.

Çizelge 6.1 Kombinasyonları denenen parametreler ve değerler

Parametre	Değer 1	Değer 2	Değer 3	Değer 4
Birinci gizli katman nöron sayısı	60	40	20	12
İkinci gizli katman nöron sayısı	30	20	12	
Üçüncü gizli katman nöron sayısı	20	12		
Gizli katman aktivasyon fonksiyonları	ddü	linear		
Çıkış katmanı aktivasyon fonksiyonu	sigmoid	hard_sigmoid		
Yanlılık başlangıç fonksiyonu	zeros	uniform		
Çekirdek başlangıç fonksiyonu	uniform	randomUniform		
Hata fonksiyonu	ortalama kare hatası	ortalama kare logaritmik hata	ikili çapraz entropi	
Optimizasyon yöntemi	adam	RMSprop	adamax	nadam

Toplam 4608 kombinasyon Scikit-learn kütüphanesinde bulunan Örgü arama modeli kullanılarak katmanlı çapraz doğrulama yapılmış, hata fonksiyonları (şekil 6.1) değerlendirilerek en iyi senaryo parametre değerleri tespit edilmiştir. Test sırasında ağ

güncelleme periyodu 10 seçilmiştir. Deneme süresini kısaltmak için döngü sayısı daha önce yapılan testler göz önünde bulundurularak 25 alınmıştır.



Şekil 6.1 Örnek döngü sayısı hata fonksiyonu grafiği

6.2 Ağ Performansı Değerlendirme Metrikleri

Veri düzenleme, ağ tasarlama ve uygulama işlemlerinden sonra elde edilen sonuçların sağlıklı bir şekilde değerlendirilmesi gerekmektedir. Farklı makine öğrenim algoritmaları için farklı değerlendirme metrikleri kullanılması gerekmektedir. Bu çalışmada; doğruluk, kesinlik, hassaslık, belirlilik ve f1 skoru metrikleri ile sonuçlar değerlendirilecektir.

Değerlendirmeler hata matrisi oluşturularak yapılacaktır. Hata matrisi, modelin doğruluğunu bulmak için kullanılan en sezgisel ve en kolay ölçümlerden biridir. Çıktının iki veya daha fazla çıktısı olan ağlarda da kullanılabilir.

Bir ve sıfır olmak üzere iki çıktısı bulunan bir sınıflandırma probleminde tahmin sonuçları hata matrisinde dört kategoriye ayrılır. Şekil 6.2'de de görülen doğru pozitif (DP), yanlış pozitif (YP), yanlış negatif (YN), doğru negatif (DN) ölçütleri kullanılarak metrik hesaplamaları yapılmaktadır.

	Pozitif Değer	Negatif Değer
Pozitif Tahmin	DP	YP
Negatif Tahmin	YN	DN

Şekil 6.2 Hata matrisi tahmin dağılımına göre tanımlamalar

Doğru pozitif ölçütü sahtekarlık çağrılarının doğru tahmin edilmesi durumunu belirtirken, yanlış negatif ölçütü sahtekarlık çağrılarının tahmin edilemediğini belirtir. Doğru negatif ölçütü normal çağrıların doğru tahmin edildiğini belirtir. Yanlış pozitif ölçütü ise normal çağrıların sahtekarlık çağrısı olarak tahmin edilmesi durumunu belirtir.

Hata matrislerinde YN ve YP ölçütlerinin minimize edilmesi amaçlanarak başarılı bir ağ modeli oluşturulmaya çalışılır.

6.2.1 Doğruluk

Sınıflandırma problemlerinde doğruluk, model tarafından yapılan tüm tahminler üzerinden yapılan doğru tahminlerin sayısıdır. Veri setinin dengeli olduğu durumlarda doğruluk metriği iyi bir değerlendirme yöntemidir. Sınıf dengesizliği olan durumlarda bu metriğin kullanılması performans değerlendirmesinde yetersizdir. Eşitlik 6.1'de Φ sembolü doğruluk metriğini belirtmektedir.

$$\Phi = \frac{DP+DN}{DP+YP+YN+DN} \tag{6.1}$$

6.2.2 Kesinlik

Kesinlik metriği, doğru tahmin edilen sahtekarlık çağrılarının tüm tahminler içindeki yerini gösteren metriktir. Eşitlik 6.2'de ζ sembolü kesinlik metriğini ifade etmektedir.

$$\zeta = \frac{\mathrm{DP}}{\mathrm{DP+YP}} \tag{6.2}$$

6.2.3 Hassaslık

Hassaslık metriği, doğru tahmin edilen sahtekarlık çağrılarının tüm sahtekarlık çağrıları içindeki oranını veren bir metriktir. Yakalanamamış sahtekarlık çağrılarının öne çıkması için kullanılır. Eşitlik 6.3'de φ sembolü hassaslık metriğini ifade etmektedir.

$$\varphi = \frac{\mathrm{DP}}{\mathrm{DP+YN}} \tag{6.3}$$

6.2.4 Belirlilik

Hassaslık metriğinin tersi gibi çalışır. Doğru tahmin edilen normal çağrıların tüm çağrılara oranını veren metriktir. Hatalı olarak sahtekarlık çağrısı tahmin edilen çağrıları öne çıkarmak amacıyla kullanılmaktadır. Eşitlik 6.4'de ϖ sembolü belirlilik metriğini ifade etmektedir.

$$\varpi = \frac{\mathrm{DN}}{\mathrm{DN+YP}} \tag{6.4}$$

6.2.5 F1 skoru

F1 skor metriği, ağ performans değerlendirmesi yapılırken farklı metrik değerlerinin karşılaştırmalı olarak değerlendirilmesi yerine, kesinlik ve hassaslık metrikleri ile tek bir skor oluşturulan bir metriktir. Hassaslık ve kesinlik değerlerinin aritmetik ortalaması skorları arasındaki farkın yüksek olması durumunda başarısız olmaktadır. Bu nedenle harmonik ortalama ile f1 skoru hesaplanmaktadır. İki metrik arasındaki farkın yüksek olması durumunda düşük olana yakın bir değer çıkmaktadır.

$$f1 = \frac{2*\frac{DP}{DP+YP}*\frac{DP}{DP+YN}}{\frac{DP}{DP+YP}+\frac{DP}{DP+YN}} \quad (6.5)$$

6.3 ÇDK Bilgileri Kullanılarak Sahtekarlık Tespiti için Ağ Seçimi

Çizelge 6.1'de belirtilen parametrelerin kullanıldığı 4608 ağ modeli k katmanlı çapraz doğrulama yöntemi ile k=3 için yapılan denemelerin sonuç değerlendirmeleri sonrası eğitim performansı en yüksek olan ağların örgü arama modeli test sonuçları çizelge 6.2'de paylaşılmıştır.

En yüksek eğitim performansına sahip 10 model 150 döngü sayısı ile tekrar eğitilerek performans metrikleri çıkarılmıştır.

Çizelge 6.2 Örgü arama modeli testlerinde performansı yüksek ÇDK bilgisi ile eğitilmiş ağlar

Parametre	Model 1	Model 2	Model 3	Model 4	Model 5
Birinci gizli katman nöron sayısı	60	60	60	60	60
İkinci gizli katman nöron sayısı	20	20	20	20	12
Üçüncü gizli katman nöron sayısı	12	12	12	12	12
Gizli katman aktivasyon fonksiyonları	ddü	ddü	ddü	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu	zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu	uniform	random uniform	random uniform	random uniform	uniform
Hata fonksiyonu	ikili çapraz entropi	ortalama kare hatası	ikili çapraz entropi	ikili çapraz entropi	ortalama kare hatası

Çizelge 6.2 Örgü arama modeli testlerinde performansı yüksek ÇDK bilgisi ile eğitilmiş ağlar (devam)

Optimizasyon yöntemi	adam	adamax	adamax	adam	adagrad
Ortalama Test Skoru	0.97763975	0.97763975	0.97763975	0.97763975	0.97639752
Test Varyansı	0.00807707	0.00807707	0.00807707	0.00807707	0.00705013
Ortalama Eğitim Skoru	0.98074340	0.98136529	0.98136529	0.98136529	0.98074455
Eğitim Varyansı	0.00317879	0.00264260	0.00264260	0.00264260	0.00317017
	Model 6	Model 7	Model 8	Model 9	Model 10
Birinci gizli katman nöron sayısı	60	60	60	60	60
İkinci gizli katman nöron sayısı	20	20	20	20	12
Üçüncü gizli katman nöron sayısı	12	12	20	12	12
Gizli katman aktivasyon fonksiyonları	ddü	ddü	linear	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu	zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu	random uniform	uniform	random uniform	uniform	random uniform
Hata fonksiyonu	ortalama kare hatası	ortalama kare hatası	ikili çapraz entropi	ortalama kare hatası	ortalama kare hatası
Optimizasyon yöntemi	adagrad	adagrad	adamax	RMSprop	adagrad
Ortalama Test Skoru	0.97639752	0.97639752	0.97639752	0.97515528	0.97515528
Test Varyansı	0.00705013	0.00980875	0.00980875	0.00634974	0.01154851
Ortalama Eğitim Skoru	0.98261023	0.98198602	0.98198602	0.98323096	0.98198602
Eğitim Varyansı	0.00350486	0.00352044	0.00352044	0.00262702	0.00352044

6.3.1 Model değerlendirme

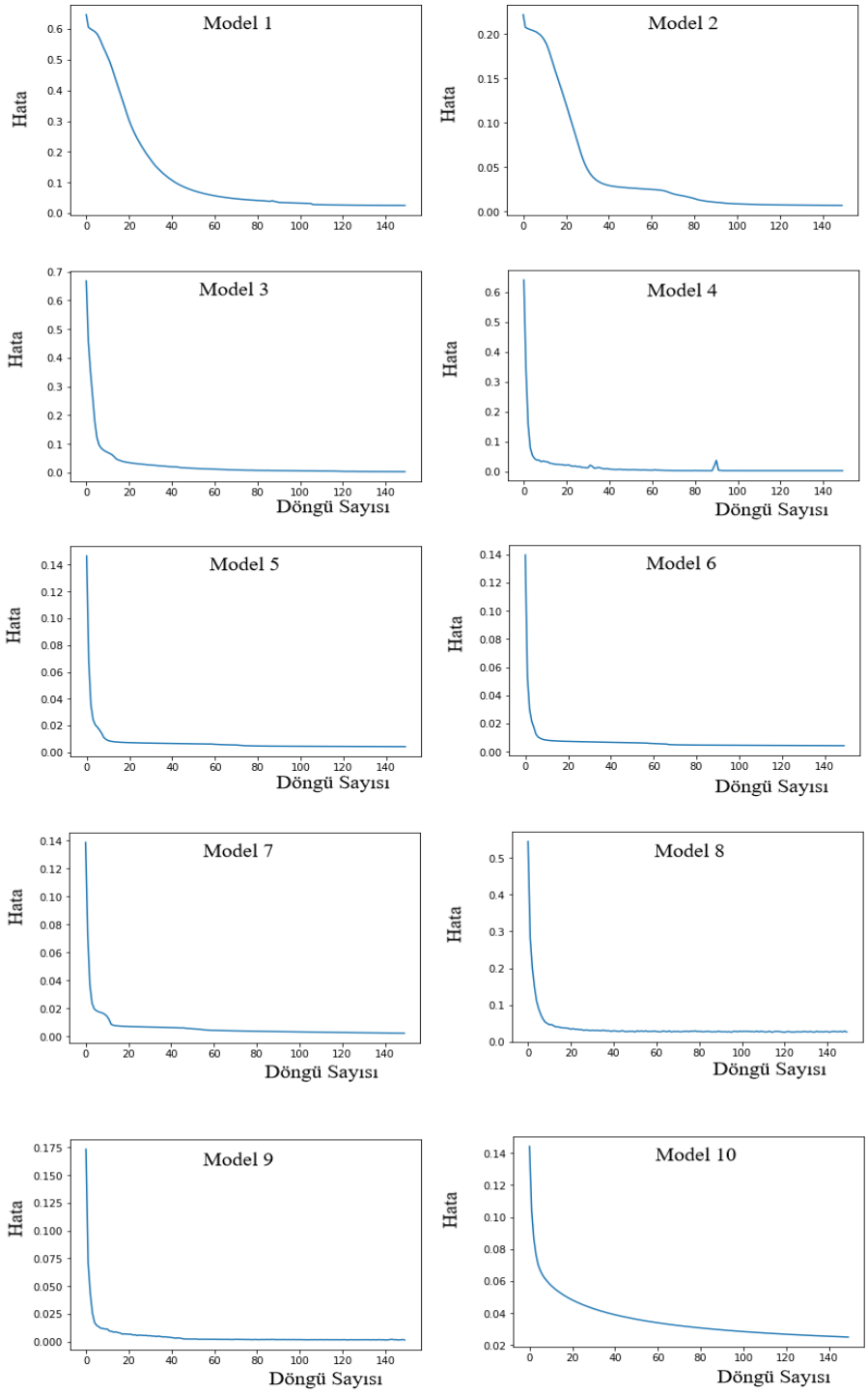
Çizelge 6.3'de üzerinde detaylı çalışılan modeller ve performans metrikleri yer almaktadır. Performans metrikleri ve şekil 6.3'de grafikleri verilen hata fonksiyonları (yatay eksenler döngü sayısını, düşey eksenler hatayı ifade etmektedir) incelenerek 1 numaralı modelin en verimli model olduğuna karar verilmiştir.

Çizelge 6.3 Birinci veri seti ile denenen model performansları

Parametre		Model 1	Model2	Model 3	Model 4	Model 5
Birinci gizli katman nöron sayısı		60	60	60	60	60
İkinci gizli katman nöron sayısı		20	20	20	20	12
Üçüncü gizli katman nöron sayısı		12	12	12	12	12
Gizli katman aktivasyon fonksiyonları		ddü	ddü	ddü	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu		sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu		zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu		uniform	random uniform	random uniform	random uniform	uniform
Hata fonksiyonu		ikili çapraz entropi	ortalama kare hatası	ikili çapraz entropi	ikili çapraz entropi	ortalama kare hatası
Optimizasyon yöntemi		adam	adamax	adamax	adam	adagrad
10 katmanlı çapraz doğrulama	performans	0.97145061	0.96770061	0.96401234	0.97145061	0.97023148
	varyans	0.01659880	0.01488331	0.01873698	0.01239304	0.01671767
Hata Matrisi	DP	115	114	113	111	113
	DN	223	224	224	226	225
	YN	0	1	2	4	2
	YP	7	6	6	4	5
Doğruluk		0.97971014	0.97971014	0.97681159	0.97681159	0.97971014
Hassaslık		1	0.99130435	0.98260870	0.96521739	0.98260870
Kesinlik		0.94262295	0.95	0.94957983	0.96521739	0.95762712
Belirlilik		1	0.99555556	0.99115044	0.98260870	0.99118943
F1 skoru		0.97046414	0.97021277	0.96581197	0.96521739	0.96995708

Çizelge 6.3 Birinci veri seti ile denenen model performansları (devam)

Parametre		Model 6	Model 7	Model 8	Model 9	Model 10
Birinci gizli katman nöron sayısı		60	60	60	60	60
İkinci gizli katman nöron sayısı		20	20	20	20	12
Üçüncü gizli katman nöron sayısı		12	12	20	12	12
Gizli katman aktivasyon fonksiyonları		ddü	ddü	linear	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu		sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu		zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu		random uniform	uniform	random uniform	uniform	random uniform
Hata fonksiyonu		ortalama kare hatası	ortalama kare hatası	ikili çapraz entropi	ortalama kare hatası	ortalama kare hatası
Optimizasyon yöntemi		adagrad	adagrad	adamax	RMSprop	adagrad
10 katmanlı çapraz doğrulama	Performans	0.97516975	0.97393518	0.95526234	0.96273148	0.97516975
	Varyans	0.00958418	0.01025822	0.01272347	0.01361077	0.00958418
Hata Matrisi	DP	113	113	111	113	113
	DN	225	225	225	226	225
	YN	2	2	4	2	2
	YP	5	5	5	4	5
Doğruluk		0.97971014	0.97971014	0.97391304	0.98260870	0.97971014
Hassaslık		0.98260870	0.98260870	0.96521739	0.98260870	0.98260870
Kesinlik		0.95762712	0.95762712	0.95689655	0.96581197	0.95762712
Belirlilik		0.99118943	0.99118943	0.98253275	0.99122807	0.99118943
F1 skoru		0.96995708	0.96995708	0.96103896	0.97413793	0.96995708



Şekil 6.3 Birinci veri seti ile denenen modellerin hata fonksiyonlarının döngü sayısına göre değişim grafiği

6.4 OBP Davet Mesajı kullanılarak Sahtekarlık Tespiti için Ağ Seçimi

Çizelge 6.1'de belirtilen parametrelerin kullanıldığı 4608 ağ modeli k katmanlı çapraz doğrulama yöntemi ile k=3 için yapılan denemelerin sonuç değerlendirmeleri sonrası eğitim performansı en yüksek olan ağlar çizelge 6.4'de paylaşılmıştır.

En yüksek eğitim performansına sahip 5 model 500 döngü sayısı ile tekrar eğitilerek performans metrikleri çıkarılmıştır.

Çizelge 6.4 Örgü arama modeli testlerinde yüksek performans gösteren modeller ve sonuçları

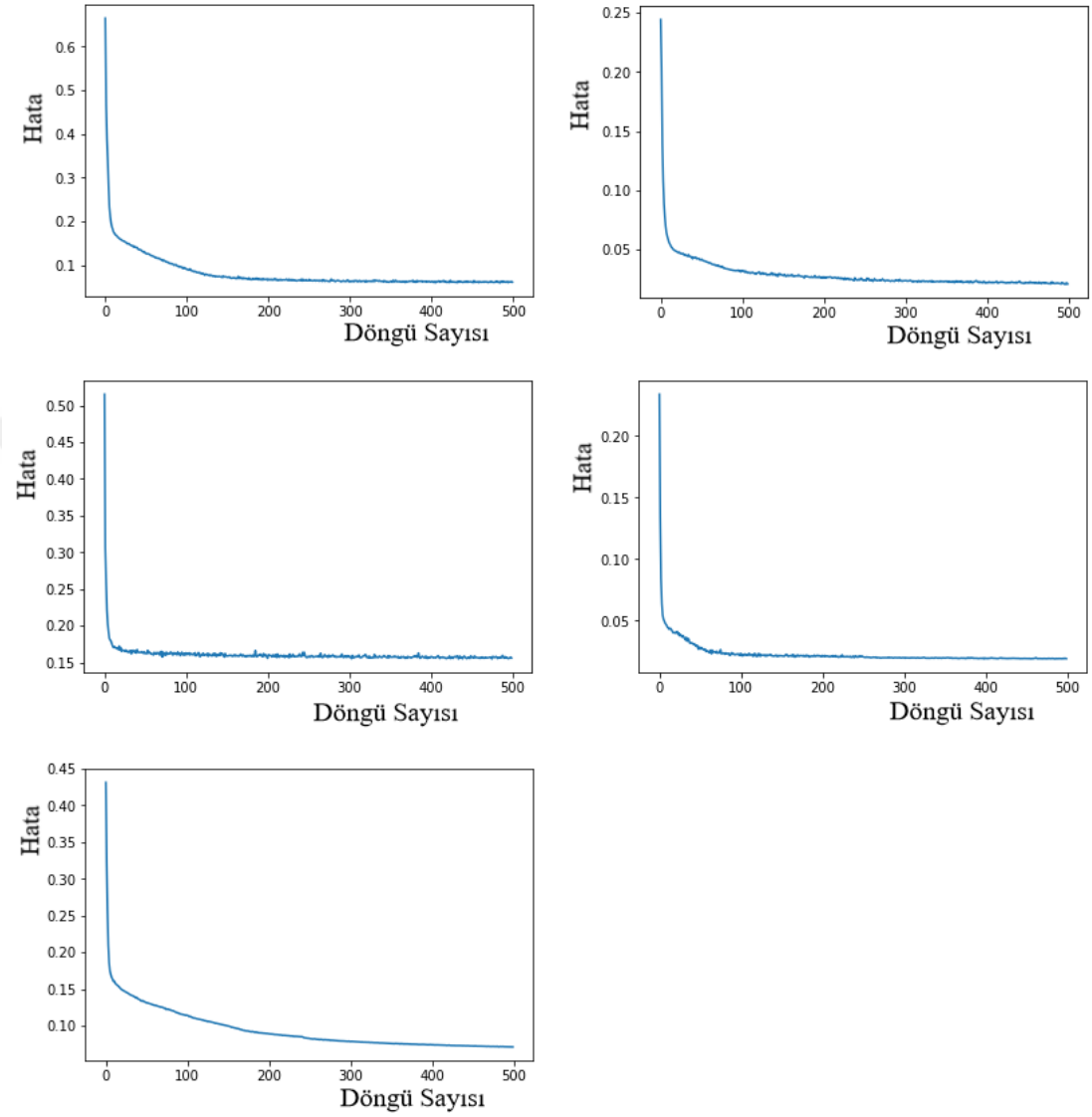
Parametre	Model SIP1	Model SIP2	Model SIP3	Model SIP4	Model SIP5
Birinci gizli katman nöron sayısı	60	60	60	60	60
İkinci gizli katman nöron sayısı	12	12	20	20	20
Üçüncü gizli katman nöron sayısı	20	12	20	12	12
Gizli katman aktivasyon fonksiyonları	ddü	ddü	linear	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu	zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu	uniform	random uniform	uniform	uniform	random uniform
Hata fonksiyonu	İkili çapraz entropi	ortalama kare hatası	İkili çapraz entropi	ortalama kare hatası	İkili çapraz entropi
Optimizasyon yöntemi	adamax	adamax	adam	adam	adagrad
Ortalama Test Skoru	0.9105590	0.9055901	0.9055901	0.9043478	0.9031056
Test Varyansı	0.0210100	0.0194616	0.0212316	0.0036015	0.0120206
Ortalama Eğitim Skoru	0.9627364	0.9658412	0.9608811	0.9695610	0.9677022
Eğitim Varyansı	0.0054621	0.0074896	0.0144881	0.0061707	0.0068545

6.4.1 Model Değerlendirme

Çizelge 6.5'de üzerinde detaylı çalışılan modeller ve performans metrikleri yer almaktadır. Şekil 6.4'te grafikleri verilen hata fonksiyonları (yatay eksenler döngü sayısını, düşey eksenler hatayı ifade etmektedir) ve performans metrikleri incelenerek hassaslık, belirlilik ve doğruluk değerleri en yüksek olan 3 numaralı model (çalışmanın geri kalanında Model SIP1 olarak adlandırılacaktır) ile çalışılmasına karar verilmiştir.

Çizelge 6.5 İkinci veri seti ile denenen model performansları

Parametre		Model SIP1	Model SIP2	Model SIP3	Model SIP4	Model SIP5
Birinci gizli katman nöron sayısı		60	60	60	60	60
İkinci gizli katman nöron sayısı		12	12	20	20	20
Üçüncü gizli katman nöron sayısı		20	12	20	12	12
Gizli katman aktivasyon fonksiyonları		ddü	ddü	linear	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu		sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu		zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu		uniform	random uniform	uniform	uniform	random uniform
Hata fonksiyonu		İkili çapraz entropi	ortalama kare hatası	İkili çapraz entropi	ortalama kare hatası	İkili çapraz entropi
Optimizasyon yöntemi		adamax	adamax	adam	adam	adagrad
10 katmanlı çapraz doğrulama	performans	0.90304011	0.90307098	0.89814814	0.89441357	0.90558641
	varyans	0.04700026	0.03937414	0.03758500	0.04635278	0.04057438
Hata Matrisi	DP	72	75	78	70	69
	DN	237	237	236	243	240
	YN	14	11	8	16	17
	YP	22	22	23	16	19
Doğruluk		0.89565217	0.90434783	0.91014493	0.90724638	0.89565217
Hassaslık		0.83720930	0.87209302	0.90697674	0.81395349	0.80232558
Kesinlik		0.76595745	0.77319588	0.77227723	0.81395349	0.78409091
Belirlilik		0.94422311	0.95564516	0.96721311	0.93822394	0.93385214
F1 skoru		0.8	0.81967213	0.83422460	0.81395349	0.79310345



Şekil 6.4 İkinci veri seti ile denenen modellerin hata fonksiyonlarının döngü sayısına göre değişim grafiği

6.5 Seçilen Ağ Performanslarını Yükseltme Çalışmaları

Ağ test performanslarını artırmak için seçili modeller üzerinde seyreltme düzenleme yöntemi uygulanmış ve farklı öğrenme oranlarındaki performanslar gözlenmiştir.

6.5.1 Model 1

Yapılan seyreltme düzenleme ve eğitim oranı değişiklikleri sonrası model performans metrik değerleri çizelge 6.6'da paylaşılmıştır. Yapılan denemeler ile hata grafikleri ve performans metriklerine göre 2 numaralı deneme sonrası elde edilen ağırlıklar kaydedilmiş ve ağ modeli olarak kullanılmıştır.

Çizelge 6.6 Model 1 deneme sonuçları

		Model 1	Deneme 1	Deneme 2	Deneme 3	Deneme 4	Deneme 5	Deneme 6
Birinci gizli katman seyreltme oranı		0	0.5	0	0.5	0.25	0.1	0.25
İkinci gizli katman seyreltme oranı		0	0.3	0	0.2	0.3	0.3	0.3
Üçüncü gizli katman seyreltme oranı		0	0.5	0	0.5	0.4	0.4	0.4
Öğrenme Oranı		0.001	0.001	0.01	0.01	0.1	0.0001	0.1
Hata Matrisi	DP	115	113	114	113	110	110	110
	DN	223	227	226	227	226	226	226
	YN	0	2	1	2	5	5	5
	YP	7	3	4	3	4	4	4
Doğruluk		0.979710	0.985507	0.985507	0.985507	0.973913	0.973913	0.973913
Hassaslık		1	0.982609	0.991304	0.982609	0.956522	0.956522	0.956522
Kesinlik		0.942623	0.974138	0.966102	0.974138	0.964912	0.964912	0.964912
Belirlilik		1	0.991266	0.995595	0.991266	0.978355	0.978355	0.978355
F1 skoru		0.970464	0.978355	0.978541	0.978355	0.9606988	0.960698	0.960699

6.5.2 Model SIP1

Yapılan seyreltme düzenleme ve eğitim oranı değişiklikleri sonrası model performans metrik değerleri çizelge 6.7'de paylaşılmıştır. Yapılan denemeler ile hata grafikleri ve performans metriklerine göre 3 numaralı deneme sonrası elde edilen ağırlıklar kaydedilmiştir.

Çizelge 6.7 Model SIP1 deneme sonuçları

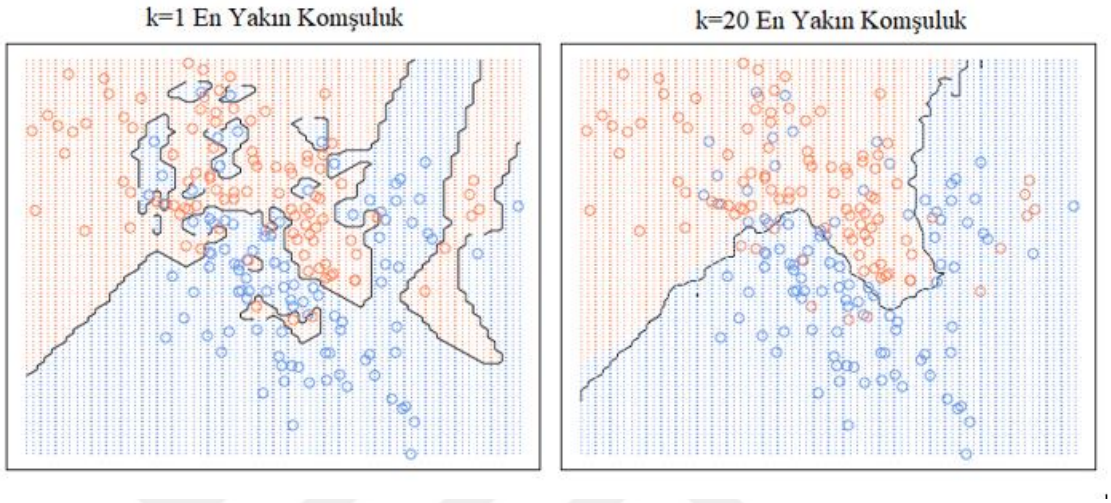
		Model SIP1	Deneme1	Deneme2	Deneme3	Deneme4	Deneme5	Deneme6
Birinci gizli katman seyreltme oranı		0	0.35	0.5	0.5	0.1	0.4	0.5
İkinci gizli katman seyreltme oranı		0	0.35	0.2	0.2	0.5	0.5	0.2
Üçüncü gizli katman seyreltme oranı		0	0.35	0.2	0.2	0.2	0.5	0.1
Öğrenme Oranı		0.001	0.01	0.001	0.00001	0.005	0.001	0.00001
Hata Matrisi	DP	78	76	75	78	74	74	76
	DN	236	237	237	237	237	237	241
	YN	8	10	11	8	12	12	10
	YP	23	22	22	22	22	22	18
Doğruluk		0.9101449	0.9072464	0.9043478	0.9130435	0.9014493	0.9014493	0.9188406
Hassaslık		0.9069767	0.8837209	0.8720930	0.9069767	0.8604651	0.8604651	0.8837209
Kesinlik		0.7722772	0.7755102	0.7731959	0.78	0.7708333	0.7708333	0.8085106
Belirlilik		0.9672131	0.9595142	0.9556452	0.9673469	0.9518072	0.9518072	0.9601594
F1 skoru		0.8342246	0.8260870	0.8196721	0.8387097	0.8131868	0.8131868	0.8444444

6.6 KEYK Algoritması ile Tasarım

Tasarlanan ağların performans karşılaştırmalarının yapılabilmesi amacıyla farklı bir algoritma, aynı eğitim ve test verileri ile bir KEYK ağı tasarlanmıştır. Basit yapısına rağmen, güçlü ve karmaşık algoritmalarda daha iyi performans gösterebilmesi nedeniyle karşılaştırmalarda KEYK algoritmasının kullanılmasına karar verilmiştir.

KEYK algoritmasının bir hiper parametresi olan k değeri, diğer makine öğrenme yöntemlerinde olduğu gibi deneme yanılma yöntemi ile bulunur. Data seti ile en iyi uyumu sağlayacak parametrenin seçilmesi yüksek performans için gereklidir. Küçük bir k değerinin seçilmesi eğitim setine en esnek uyumu sağlar. Fakat düşük sapma ve yüksek varyans nedeniyle aşırı uyum problemi ile karşılaşma riskini ortaya çıkarır. Yüksek k değeri ise daha

genel bir yapıda dağılım yaparken detaya inmede başarılı olamaz. Yüksek k değerinde yüksek sapma ve düşük varyans problemi ile karşılaşılabilir.



Şekil 6.5 Örnek k değerlerinin KEYK modellemesi üzerine etkisi (Zakka 2016)

6.6.1 K değerinin belirlenmesi

K değerinin belirlenmesi için ÇDK bilgileri kullanılarak üretilen veri seti ile yapılan denemeler ve sonuçları çizelge 6.8'de verilmiştir. Deneme sonuçlarına göre en iyi performans sağlanan durum olan k hiper parametresinin 1 olması durumudur.

Çizelge 6.8 KEYK algoritmasının birinci veri seti ile eğitimi sonrası performans metrik değerleri

		KEYK CDR1	KEYK CDR2	KEYK CDR3	KEYK CDR4	KEYK CDR5
k		3	10	2	50	1
Hata Matrisi	DP	95	87	88	80	101
	DN	219	220	228	210	226
	YN	20	28	27	35	14
	YP	11	10	2	20	4
Doğruluk		0.910144928	0.889855072	0.915942029	0.84057971	0.947826087
Hassaslık		0.826086957	0.756521739	0.765217391	0.695652174	0.87826087
Kesinlik		0.896226415	0.896907216	0.977777778	0.8	0.961904762
Belirlilik		0.916317992	0.887096774	0.894117647	0.857142857	0.941666667
F1 skoru		0.859728507	0.820754717	0.858536585	0.744186047	0.918181818

K değerinin belirlenmesi için OBP davet mesajı içeriğindeki bilgiler kullanılarak üretilen veri seti ile yapılan denemeler ve sonuçları çizelge 6.9'da verilmiştir. Deneme sonuçlarına göre en iyi performans sağlanan durum, k hiper parametresinin 1 olması durumudur.

Çizelge 6.9 KEYK algoritmasının ikinci veri seti ile eğitimi sonrası performans metrik değerleri

		KEYK SIP1	KEYK SIP2	KEYK SIP3	KEYK SIP4	KEYK SIP5
k		3	10	2	50	1
Hata Matrisi	DP	95	88	90	80	101
	DN	217	221	228	210	225
	YN	20	27	25	35	14
	YP	13	9	2	20	5
Doğruluk		0.904347826	0.895652174	0.92173913	0.84057971	0.944927536
Hassaslık		0.826086957	0.765217391	0.782608696	0.695652174	0.87826087
Kesinlik		0.87962963	0.907216495	0.97826087	0.8	0.952830189
Belirlilik		0.915611814	0.891129032	0.901185771	0.857142857	0.941422594
F1 skoru		0.852017937	0.830188679	0.869565217	0.744186047	0.914027149

6.7 İşlemci İhtiyacı Az Olan İki Ağ Tasarımı

Tasarlanan Model 1 ağına alternatif olarak, farklı eğitim setleri ile eğitilmiş ve daha az işlemci kullanan ortak karar veren iki ağ tasarlanmıştır. Ağlar birinci veri setinin iki parçaya ayrılması ile oluşturulan veri setleri ile eğitilmişlerdir.

Agresif veri seti olarak adlandırılan veri setinde birinci veri setindeki tüm sahtekarlık aramaları ve 190 adet normal arama olmak üzere toplam 542 adet arama bulunmaktadır. Bu eğitim setinin kullanılmasının amacı ağın daha agresif karar alması ve tüm sahtekarlık çağrılarının tespit edilmesidir.

Tolerans veri seti olarak adlandırılan veri setinde ikinci veri setindeki tüm normal aramalar ve 163 adet sahtekarlık araması olmak üzere toplam 961 arama bulunmaktadır. Bu eğitim setinin kullanılmasının amacı ağın daha toleranslı olarak karar alması ve normal çağrıların hatalı olarak sahtekarlık çağrısı süreçlerine dahil edilmemesidir.

Bu eğitim setlerine göre agresif veri seti ile eğitilen ağın hassaslık metriğinin yüksek olması, tolerans veri seti ile eğitilen ağın belirlilik metriğinin yüksek olması beklenmektedir. İki ağın birleştirilmesi sonrası yapılan testlerde tüm ağ metriklerinde iyileşme görülmesi ve daha az işlemci ihtiyacı olan iki farklı ağ ile yapılandırılmış karma bir yapının alternatif bir yöntem olup olamayacağı değerlendirilecektir.

6.7.1 Agresif ve Tolerans veri setleri ile eğitilecek ağların seçimi

Ağ seçimi için örgü arama yöntemi ile çizelge 6.10'da verilen 192 farklı ağ parametrelerinin performans değerlendirmeleri yapılmış, en iyi değerlere sahip ağlar çizelge 6.11'de paylaşılmıştır.

Çizelge 6.10 Agresif ve Tolerans modelleri için kombinasyonları denenen farklı parametre ve değerler

Parametre	Değer 1	Değer 2	Değer 3
Birinci gizli katman nöron sayısı	40	30	20
İkinci gizli katman nöron sayısı	20	12	
Üçüncü gizli katman nöron sayısı	20	12	
Aktivasyon fonksiyonları	ddü	linear	
Çekirdek başlangıç fonksiyonu	uniform	random uniform	
Hata fonksiyonu	ortalama kare hatası	ikili çapraz entropi	
Optimizasyon yöntemi	adam	adamax	

Çizelge 6.11 Örgü arama modeli testlerinde yüksek performans gösteren modeller ve test sonuçları

Parametre	Agresif 1	Agresif 2	Agresif 3	Tolerans 1	Tolerans 2	Tolerans 3
Birinci gizli katman nöron sayısı	40	40	30	30	40	40
İkinci gizli katman nöron sayısı	20	12	12	20	12	12
Üçüncü gizli katman nöron sayısı	12	12	12	12	12	12
Gizli katman aktivasyon fonksiyonları	ddü	ddü	ddü	ddü	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu	zeros	zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu	uniform	random uniform	random uniform	uniform	random uniform	random uniform
Hata fonksiyonu	ikili çapraz entropi	ortalama kare hatası	ikili çapraz entropi	ikili çapraz entropi	ikili çapraz entropi	ikili çapraz entropi
Optimizasyon yöntemi	adagrad	adagrad	adagrad	adagrad	adagrad	adagrad
Ortalama Test Skoru	0.93461538	0.92307692	0.92307692	0.95404412	0.95220588	0.95955882
Test Varyansı	0.00509352	0.01401739	0.01401739	0.01385904	0.00945997	0.0094676
Ortalama Eğitim Skoru	1	1	1	0.99908173	0.99724264	0.9954061
Eğitim Varyansı	0	0	0	0.000013	0.035900	0.000013

En yüksek eğitim performansına sahip her bir eğitim seti için 3 model 300 döngü sayısı ile tekrar eğitilerek performans metrikleri çıkarılmıştır.

6.7.2 Model değerlendirme

Çizelge 6.12'de üzerinde detaylı çalışılan modeller ve performans metrikleri yer almaktadır. Verilen performans metrikleri incelenerek agresif eğitim seti kullanılmış ağ için en yüksek hassaslık değerine sahip olan ağ; tolerans eğitim seti ile eğitilmiş ağ için ise en yüksek doğruluk oranına sahip olan ağ ile çalışmalara devam edilmiştir.

Çizelge 6.12 Agresif ve Tolerans İkinci veri setleri ile denenen model performansları

Parametre		Agresif 1	Agresif 2	Agresif 3	Tolerans 1	Tolerans 2	Tolerans 3
Birinci gizli katman nöron sayısı		40	40	30	30	40	40
İkinci gizli katman nöron sayısı		20	12	12	20	12	12
Üçüncü gizli katman nöron sayısı		12	12	12	12	12	12
Gizli katman aktivasyon fonksiyonları		ddü	ddü	ddü	ddü	ddü	ddü
Çıkış katmanı aktivasyon fonksiyonu		sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Yanlılık başlangıç fonksiyonu		zeros	zeros	zeros	zeros	zeros	zeros
Çekirdek başlangıç fonksiyonu		uniform	random uniform	random uniform	uniform	random uniform	uniform
Hata fonksiyonu		İkili çapraz entropi	Ortalama kare hatası	İkili çapraz entropi	İkili çapraz entropi	İkili çapraz entropi	İkili çapraz entropi
Optimizasyon yöntemi		adagrad	adagrad	adagrad	adagrad	adagrad	adagrad
10 katmanlı çapraz doğrulama	performans	0.9346154	0.9230769	0.9230769	0.9540441	0.9522059	0.9595588
	varyans	0.0050935	0.0140174	0.0140174	0.0138590	0.0094600	0.0094676
Hata Matrisi	DP	103	106	106	35	34	39
	DN	50	52	51	246	249	246
	YN	2	3	3	4	5	2
	YP	6	4	5	4	1	4
Doğruluk		0.9503106	0.9575758	0.9515152	0.9723183	0.9792388	0.9793814
Hassaslık		0.9809524	0.9724771	0.9724771	0.8974359	0.8717949	0.9512195
Kesinlik		0.9449541	0.9636364	0.9549550	0.8974359	0.9714286	0.9069767
Belirlilik		0.9615385	0.9454545	0.9444444	0.984	0.9803150	0.9919355
F1 skoru		0.9626168	0.9680365	0.9636364	0.8974359	0.9189189	0.9285714

Agresif eğitim seti ile eğitilmiş ağlar içinde Agresif 1 ağı en yüksek hassaslık oranına sahiptir. Tolerans eğitim seti ile eğitilmiş ağlar içinde Tolerans 3 ağı en yüksek doğruluk oranına sahiptir. Çalışmalara Agresif 1 v Tolerans 3 ağları ile devam edilmiştir. Tolerans 3 ağı çalışmanın devamında Tolerans 1 olarak adlandırılmaktadır.

6.7.3 Seçilen ağ performanslarını yükseltme çalışmaları

Ağ test performanslarını artırmak için seçili modeller üzerinde seyreltme düzenleme yöntemi uygulanmış ve farklı öğrenme oranlarındaki performanslar gözlenmiştir.

Çalışma sonrası Agresif model performans metrik değerleri çizelge 6.13'de paylaşılmıştır. Yapılan denemeler sonrası performans metriklerine göre en yüksek hassaslık metriği elde edilen 4 numaralı deneme ağırlıklar kaydedilmiştir ve karşılaştırmalarda kullanılmıştır.

Çizelge 6.13 Agresif 1 model deneme sonuçları

		Agresif 1	Deneme1	Deneme2	Deneme3	Deneme4	Deneme5	Deneme6
Birinci gizli katman seyreltme oranı		0	0.1	0.1	0.5	0.5	0.5	0.25
İkinci gizli katman seyreltme oranı		0	0.4	0.4	0.2	0.4	0.2	0.2
Üçüncü gizli katman seyreltme oranı		0	0.1	0.3	0.5	0.2	0.1	0.1
Öğrenme Oranı		0.01	0.01	0.001	0.01	0.1	0.001	0.0001
Hata Matrisi	DP	106	105	107	105	107	106	107
	DN	50	50	43	53	51	44	0
	YN	1	2	13	2	0	1	0
	YP	6	6	0	3	5	12	56
Doğruluk		0.957055	0.950920	0.920245	0.969325	0.9693252	0.920245	0.656442
Hassaslık		0.990654	0.981308	0.891667	0.981308	1	0.990654	1
Kesinlik		0.946429	0.945946	1	0.972222	0.955357	0.898305	0.656442
Belirlilik		0.980392	0.961539	0.767857	0.963636	1	0.977778	0
F1 skoru		0.968037	0.963303	0.942731	0.976744	0.9771689	0.9422222	0.7925926

Tolerans model performans metrik değerleri çizelge 6.14'de paylaşılmıştır. Yapılan denemeler sonrası performans metriklerine göre en yüksek doğruluk metriği elde edilen 5 numaralı deneme ağırlıklar kaydedilmiştir ve karşılaştırmalarda kullanılmıştır.

Çizelge 6.14 Tolerans 1 model deneme sonuçları

		Toleran1	Deneme1	Deneme2	Deneme3	Deneme4	Deneme5	Deneme6
Birinci gizli katman seyreltme oranı		0	0.3	0.2	0.2	0.1	0.3	0.25
İkinci gizli katman seyreltme oranı		0	0.5	0.1	0.3	0.4	0.3	0.2
Üçüncü gizli katman seyreltme oranı		0	0.2	0.2	0.1	0.1	0.1	0.1
Öğrenme Oranı		0.01	0.1	0.001	0.01	0.01	0.01	0.001
Hata Matrisi	DP	106	31	22	36	34	39	27
	DN	50	247	248	246	246	246	244
	YN	2	8	17	3	5	0	12
	YP	4	3	2	4	4	4	6
Doğruluk		0.962963	0.961938	0.934256	0.975779	0.968858	0.9861592	0.9377163
Hassaslık		0.981482	0.794872	0.564103	0.923077	0.871795	1	0.6923077
Kesinlik		0.963636	0.911765	0.916667	0.9	0.894737	0.9069767	0.8181818
Belirlilik		0.961539	0.968628	0.935849	0.987952	0.980080	1	0.953125
F1 skoru		0.972477	0.849315	0.698413	0.911392	0.883117	0.9512195	0.75

6.7.4 Agresif ve Tolerans ağlarının birleştirilmesi

Kullanılan her bir ağ modeli için tahmin olasılıkları birer listede tutulmuştur. Bu olasılıkların ağırlıkları değiştirilerek ortalamaları alınmıştır. Örnek olarak agresif ağın ağırlıklarının yüksek tutulduğu durumda, aramaların sahtekarlık çağrısı olarak işaretlenme olasılığı daha yüksek olacaktır. Ağırlık denemeleri Agresif ve Tolerans ağları için sırasıyla; 0,25-0,75, 0,3-0,7, 0,5-0,5, 0,6-0,4 olarak alınmıştır. Çizelge 6.15'de verilen performans metrik değerlendirmeleri sonrası çalışmalara Karma 1 kombinasyonu ile devam edilmiştir

Çizelge 6.15 Ağ ağırlıklarına göre performans metrikleri

		Karma 1	Karma 2	Karma 3	Karma 4
Agresif ağırlık		0.25	0.3	0.5	0.6
Tolerans ağırlık		0.75	0.7	0.5	0.4
Hata Matrisi	DP	348	349	360	361
	DN	8424	8414	7838	6310
	YN	13	12	1	0
	YP	91	101	677	2205
Doğruluk		0.98828301	0.98726904	0.923614241	0.751577287
Hassaslık		0.96398892	0.966759003	0.997229917	1
Kesinlik		0.792710706	0.775555556	0.347155256	0.140685892
Belirlilik		0.998459168	0.998575837	0.999872433	1
F1 skoru		0.87	0.860665845	0.515021459	0.246668944

6.8 Ağ performanslarının karşılaştırılması

6.8.1 Model 1 - KEYK karşılaştırması

Ağ eğitimlerinin tamamlanmasının ardından, eğitimlerde kullanılmamış farklı bir veri seti ile ağların performanslarının karşılaştırması yapılmıştır. Karşılaştırılan ağlar; performans metriklerine göre değerlendirilmiş, çizelge 6.17'de hatalı kayıtları gözlenmiştir. Çizelge 6.16'da verilen test sonuçlarında Model 1'in daha başarılı olduğu görülmektedir.

Çizelge 6.16 Model 1 - KEYK karşılaştırmalı test sonuçları

Parametre		Model 1 (eşik değer = 0,3)	Model 1 (eşik değer = 0,5)	Model 1 (eşik değer = 0,7)	KEYK
Hata Matrisi	DP	112	112	112	87
	DN	2548	2550	2550	2541
	YN	0	0	0	25
	YP	3	1	1	10
Doğruluk		0,998873451	0,999624484	0,999624484	0,986857
Hassaslık		1	1	1	0,776756
Kesinlik		0,973913043	0,991150442	0,991150442	0,896907
Belirlilik		1	1	1	0,990257
F1 skoru		0,986784141	0,995555556	0,995555556	0,832536

Model 1 için eşik değer değerlendirmeleri sonucunda eşik değerin 0,5 olarak kullanılması uygun görülmüştür.

Çizelge 6.17 Model 1 ve KEYK algoritmasının tahmin farklılıkları ve gerçek değerler tablosu

Sıra	Gerçek Değer	KEYK	Model1	Doğru Tahmin	Sıra	Gerçek Değer	KEYK	Model1	Doğru Tahmin
62	1	0	1	Model1	1283	1	0	1	Model1
99	0	1	0	Model1	1403	1	0	1	Model1
112	1	0	1	Model1	1514	1	0	1	Model1
144	1	0	1	Model1	1572	1	0	1	Model1
161	0	0	1	KEYK	1663	1	0	1	Model1
180	1	0	1	Model1	1738	1	0	1	Model1
225	0	1	0	Model1	1754	1	0	1	Model1
329	1	0	1	Model1	1788	1	0	1	Model1
521	1	0	1	Model1	1807	1	0	1	Model1
574	1	0	1	Model1	1810	1	0	1	Model1
600	1	0	1	Model1	1879	0	1	0	Model1
750	0	1	0	Model1	1916	1	0	1	Model1
758	1	0	1	Model1	1978	0	1	0	Model1
851	1	0	1	Model1	2102	1	0	1	Model1
859	1	0	1	Model1	2284	1	0	1	Model1
999	0	1	0	Model1	2312	0	1	0	Model1
1069	0	1	0	Model1	2364	0	1	0	Model1
1081	1	0	1	Model1	2494	0	1	0	Model1

6.8.2 Model SIP1 - KEYK karşılaştırması

Ağ eğitimlerinin tamamlanmasının ardından, eğitimlerde kullanılmamış farklı bir veri seti ile ağların performanslarının karşılaştırması yapılmıştır. Karşılaştırılan ağlar; performans metriklerine göre değerlendirilmiş, çizelge 6.19'de hatalı kayıtları gözlenmiştir. Çizelge 6.18'de verilen test sonuçları incelendiğinde, bu veri seti için Model SIP1 eşik değerinin 0,5 olması gerektiği görülmektedir. Performans metrikleri kontrollerinde Model SIP1 ve KEYK algoritmasının doğruluk metriklerinin eşit olduğu, hassaslık metriğinde ise Model SIP1'in yüksek başarı gösterdiği görülmektedir.

Çizelge 6.18 Model SIP1 - KEYK karşılaştırmalı test sonuçları

Parametre		Model SIP1 (eşik değer = 0,3)	Model SIP1 (eşik değer = 0,5)	Model SIP1 (eşik değer = 0,7)	KEYK
Hata Matrisi	DP	110	106	89	87
	DN	2452	2517	2522	2541
	YN	2	6	23	27
	YP	99	34	29	13
Doğruluk		0,96207285	0,984979347	0,980473151	0,984979
Hassaslık		0,982142857	0,946428571	0,794642857	0,758929
Kesinlik		0,526315789	0,757142857	0,754237288	0.86735
Belirlilik		0,999185004	0,997621879	0,990962672	0,989474
F1 skoru		0,685358255	0,841269841	0,773913043	0,809524

Çizelge 6.19 Model SIP1 ve KEYK algoritmasının tahmin farklılıkları ve gerçek değerler tablosu

Sıra	Gerçek Değer	KEYK	Model SIP1	Doğru Tahmin	Sıra	Gerçek Değer	KEYK	Model SIP1	Doğru Tahmin
10	0	0	1	KEYK	1147	1	1	0	KEYK
20	1	0	1	Model SIP1	1178	0	0	1	KEYK
27	0	0	1	KEYK	1214	1	1	0	KEYK
62	1	0	1	Model SIP1	1283	1	1	0	KEYK
84	0	0	1	KEYK	1285	0	0	1	KEYK
99	0	1	0	Model SIP1	1403	1	0	0	Yok
112	1	0	0	Yok	1455	1	0	1	Model SIP1
132	1	1	0	KEYK	1514	1	0	1	Model SIP1

Çizelge 6.19 Model SIP1 ve KEYK algoritmasının tahmin farklılıkları ve gerçek değerler tablosu (devam)

144	1	0	1	Model SIP1	1572	1	0	1	Model SIP1
161	0	0	1	KEYK	1663	1	0	1	Model SIP1
180	1	0	0	Yok	1680	1	1	0	KEYK
218	0	0	1	KEYK	1738	1	0	1	Model SIP1
225	0	1	0	Model SIP1	1753	1	0	1	Model SIP1
269	0	1	0	Model SIP1	1754	1	0	0	Yok
301	0	0	1	KEYK	1788	1	0	1	Model SIP1
329	1	0	1	Model SIP1	1803	0	0	1	KEYK
488	0	0	1	KEYK	1807	1	0	1	Model SIP1
504	0	0	1	KEYK	1810	1	0	1	Model SIP1
521	1	0	1	Model SIP1	1850	0	0	1	KEYK
557	0	0	1	KEYK	1879	0	1	1	Yok
574	1	0	1	Model SIP1	1916	1	0	1	Model SIP1
575	1	1	0	KEYK	1978	0	1	0	Model SIP1
600	1	0	0	Yok	1980	0	0	1	KEYK
621	0	0	1	KEYK	2019	0	0	1	KEYK
681	0	0	1	KEYK	2102	1	0	0	Yok
736	0	0	1	KEYK	2124	0	1	0	Model SIP1
749	0	0	1	KEYK	2175	1	1	0	KEYK
750	0	1	0	Model SIP1	2179	1	1	0	KEYK
758	1	0	1	Model SIP1	2201	0	0	1	KEYK
764	1	1	0	KEYK	2210	1	1	0	KEYK
851	1	0	1	Model SIP1	2213	1	1	0	KEYK
892	1	1	0	KEYK	2226	0	0	1	KEYK
908	1	1	0	KEYK	2284	1	0	1	Model SIP1
970	1	0	1	Model SIP1	2312	0	1	0	Model SIP1
977	0	1	1	Yok	2364	0	1	0	Model SIP1
999	0	1	1	Yok	2367	0	0	1	KEYK
1062	0	0	1	KEYK	2428	1	1	0	KEYK
1069	0	1	0	Model SIP1	2494	0	1	0	Model SIP1
1081	1	0	1	Model SIP1	2500	1	1	0	KEYK
1088	0	0	1	KEYK	2518	1	1	0	KEYK
1103	0	0	1	KEYK	2601	1	1	0	KEYK
1135	0	0	1	KEYK					

6.8.3 Model 1 – Model SIP1 performans karşılaştırması

Ağ eğitimlerinin tamamlanmasının ardından, eğitimlerde kullanılmamış farklı bir veri seti ile ağların performanslarının karşılaştırması yapılmıştır. Karşılaştırılan ağlar performans metriklerine göre değerlendirilmiş, çizelge 6.21'de hatalı kayıtları gözlenmiştir. Test sonuçlarında ÇDK bilgileri kullanılarak oluşturulan veri setleri ile eğitimin daha başarılı olduğu görülmektedir. İlişkilendirilebilecek daha fazla veri olması nedeniyle beklenen sonuç da bu şekildedir. Çizelge 6.20'de performans metrikleri ve detaylı sonuçlar verilmektedir.

Çizelge 6.20 Eşik değere göre model başarı parametre değerleri

Parametre		Model SIP1			Model 1		
Eşik Değer		0,3	0,5	0,7	0,3	0,5	0,7
Hata Matrisi	DP	354	339	303	359	357	357
	DN	8148	8339	8418	8506	8509	8510
	YN	7	22	58	2	4	4
	YP	367	122	97	9	6	5
Doğruluk		0.9578639	0.9836772	0.9825372	0.9987607	0.9988734	0.9989860
Hassaslık		0.9806094	0.9390582	0.8393352	0.9944598	0.9889197	0.9889197
Kesinlik		0.4909847	0.7353579	0.7575	0.9755435	0.9834711	0.9861878
Belirlilik		0.9991416	0.9973687	0.9931571	0.9997649	0.9995301	0.9995302
F1 skoru		0.6543438	0.8248175	0.7963206	0.9849108	0.9861878	0.9875519

Model 1 için en iyi kesinliğe sahip olduğu eşik değeri 0,3 olmasına rağmen en iyi doğruluk ve f1 skoru 0,7 eşik değerinde gerçekleşmektedir. Hata matrisinin eşik değer değişimlerinden çok fazla etkilenmemesi ağın istisnai durumlar dışında başarılı olduğunu gösteriyor. Model 1 ağı 0,7 eşik değerinde, tüm çağrıları %99,90 doğruluk oranı ile sınıflandırabilmiş, yine bu eşik değerinde sahtekarlık çağrılarını %98,89 oranında tespit edebilmiştir.

Model SIP1 için en iyi hassaslık ve en düşük doğruluk 0,3 eşik değerinde gerçekleşiyor. Düşük doğruluk nedeniyle 0,3 değerinin baz alınması sağlıklı sonuçlar vermeyecek, aksine çok fazla çağrıyı hatalı olarak sahtekarlık çağrısı olarak işaretleyecektir. Bu modelde kullanılabilecek en uygun eşik değeri, en yüksek f1 skorunun ve doğruluğun sağlandığı 0,5

olacaktır. Model SIP1 ağı 0,5 eşik değerinde, test setinde bulunan tüm çağrıları %98,06 doğruluk oranı ile sınıflandırmış, %93,91 oranında sahtekarlık çağrılarını tespit etmiştir.

Çizelge 6.21 Model 1 ve Model SIP1 tahmin farklılıkları ve gerçek değerler tablosu

Sıra	Gerçek Değer	Model 1	Model SIP1	Doğru Tahmin	Sıra	Gerçek Değer	Model 1	Model SIP1	Doğru Tahmin
29	0	1	1	Yok	2704	0	0	1	Model 1
34	0	1	1	Yok	2872	0	0	1	Model 1
35	0	0	1	Model 1	3000	0	0	1	Model 1
38	0	0	1	Model 1	3092	0	0	1	Model 1
47	1	1	0	Model 1	3222	0	0	1	Model 1
69	0	1	0	Model SIP1	3308	0	0	1	Model 1
115	0	0	1	Model 1	3405	1	0	0	Yok
116	0	0	1	Model 1	3429	0	0	1	Model 1
132	0	0	1	Model 1	3439	0	0	1	Model 1
152	0	0	1	Model 1	3595	0	0	1	Model 1
155	0	0	1	Model 1	3596	0	0	1	Model 1
192	0	0	1	Model 1	3597	0	0	1	Model 1
196	0	1	1	Yok	3710	1	1	0	Model 1
207	0	0	1	Model 1	3785	0	0	1	Model 1
208	0	0	1	Model 1	3855	0	0	1	Model 1
214	0	0	1	Model 1	3914	0	0	1	Model 1
218	0	0	1	Model 1	4128	0	0	1	Model 1
220	0	0	1	Model 1	4129	0	0	1	Model 1
221	0	0	1	Model 1	4130	0	0	1	Model 1
287	0	0	1	Model 1	4159	1	1	0	Model 1
289	0	0	1	Model 1	4162	1	1	0	Model 1
290	0	0	1	Model 1	4352	1	1	0	Model 1
302	0	0	1	Model 1	4675	0	0	1	Model 1
344	0	0	1	Model 1	4909	0	0	1	Model 1
349	0	0	1	Model 1	5101	0	0	1	Model 1
361	0	0	1	Model 1	5118	0	0	1	Model 1
372	1	1	0	Model 1	5213	0	0	1	Model 1
388	0	0	1	Model 1	5229	0	0	1	Model 1
444	0	1	0	Model SIP1	5309	0	0	1	Model 1
488	0	0	1	Model 1	5486	0	0	1	Model 1
490	0	0	1	Model 1	5562	0	0	1	Model 1
493	1	1	0	Model 1	5576	0	0	1	Model 1
524	1	0	1	Model SIP1	5637	0	0	1	Model 1
538	0	0	1	Model 1	5639	0	0	1	Model 1
539	1	1	0	Model 1	5652	0	0	1	Model 1

Çizelge 6.21 Model 1 ve Model SIP1 tahmin farklılıkları ve gerçek değerler tablosu (devam)

541	0	0	1	Model 1	5653	0	0	1	Model 1
543	1	0	0	Yok	5833	0	0	1	Model 1
549	0	0	1	Model 1	5834	0	0	1	Model 1
595	0	1	1	Yok	5969	0	0	1	Model 1
821	0	0	1	Model 1	5970	0	0	1	Model 1
883	0	0	1	Model 1	6088	0	0	1	Model 1
1014	0	0	1	Model 1	6089	0	0	1	Model 1
1015	0	0	1	Model 1	6126	1	1	0	Model 1
1016	0	0	1	Model 1	6441	0	0	1	Model 1
1147	0	0	1	Model 1	6442	0	0	1	Model 1
1148	0	0	1	Model 1	6534	0	0	1	Model 1
1231	0	0	1	Model 1	6632	1	1	0	Model 1
1256	1	1	0	Model 1	6694	1	1	0	Model 1
1299	0	0	1	Model 1	6717	0	0	1	Model 1
1412	1	1	0	Model 1	6719	0	0	1	Model 1
1417	0	0	1	Model 1	6783	0	0	1	Model 1
1418	0	0	1	Model 1	6798	0	0	1	Model 1
1419	0	0	1	Model 1	6914	0	0	1	Model 1
1420	0	0	1	Model 1	7011	0	0	1	Model 1
1492	1	1	0	Model 1	7076	0	0	1	Model 1
1638	0	0	1	Model 1	7126	0	0	1	Model 1
1639	0	0	1	Model 1	7268	0	0	1	Model 1
1908	0	0	1	Model 1	7275	1	0	0	Yok
1909	0	0	1	Model 1	7436	0	0	1	Model 1
1912	1	1	0	Model 1	7437	0	0	1	Model 1
2035	0	0	1	Model 1	7977	0	0	1	Model 1
2036	0	0	1	Model 1	8137	0	0	1	Model 1
2037	0	0	1	Model 1	8149	0	0	1	Model 1
2116	0	0	1	Model 1	8236	0	0	1	Model 1
2158	1	1	0	Model 1	8390	0	0	1	Model 1
2334	0	0	1	Model 1	8569	0	0	1	Model 1
2335	0	0	1	Model 1	8570	0	0	1	Model 1
2336	0	0	1	Model 1	8571	0	0	1	Model 1
2337	0	0	1	Model 1	8619	1	1	0	Model 1
2338	0	0	1	Model 1	8670	0	0	1	Model 1
2339	0	0	1	Model 1	8816	0	0	1	Model 1
2340	0	0	1	Model 1	8870	1	1	0	Model 1
2483	0	0	1	Model 1	8875	1	1	0	Model 1
2484	0	0	1	Model 1					

6.8.4 Model 1 – Karma 1 ağ performans karşılaştırması

Ağ eğitimlerinin tamamlanmasının ardından, eğitimlerde kullanılmamış farklı bir veri seti ile ağların performanslarının karşılaştırması yapılmıştır. Karşılaştırılan ağlar performans metriklerine göre değerlendirilmiş, çizelge 6.23'de hatalı kayıtları gözlenmiştir. Test sonuçlarında Model 1 ağının daha başarılı olduğu gözlenmektedir. Çizelge 6.22'de performans metrikleri ve detaylı sonuçlar verilmektedir.

Çizelge 6.22 Eşik değere göre model başarı parametre değerleri

Parametre		Karma 1			Model 1		
Eşik Değer		0,3	0,5	0,7	0,3	0,5	0,7
Hata Matrisi	DP	350	348	344	359	357	357
	DN	8389	8424	8474	8506	8509	8510
	YN	11	13	17	2	4	4
	YP	126	91	41	9	6	5
Doğruluk		0.9845651	0.9882830	0.9934655	0.9987607	0.9988734	0.9989860
Hassaslık		0.9695291	0.9639889	0.9529086	0.9944598	0.9889197	0.9889197
Kesinlik		0.7352941	0.7927107	0.8935065	0.9755435	0.9834711	0.9861878
Belirlilik		0.9986905	0.9984592	0.9979979	0.9997649	0.9995301	0.9995302
F1 skoru		0.8363202	0.87	0.9222520	0.9849108	0.9861878	0.9875519

Model 1 ağı en iyi performansı 0,7 eşik değerinde elde edilmiştir. Karma1 için en iyi hassaslık 0,3 eşik değerinde gerçekleşiyor. Düşük doğruluk nedeniyle 0,3 değerinin baz alınması sağlıklı sonuçlar vermeyecek, aksine çok fazla çağrıyı hatalı olarak sahtekarlık çağrısı olarak işaretleyecektir. Bu modelde kullanılabilecek en uygun eşik değeri, en yüksek f1 skorunun ve doğruluğun sağlandığı 0,7 olacaktır. Karma 1 ağı 0,7 eşik değerinde, test setinde bulunan tüm çağrıları %99,35 doğruluk oranı ile sınıflandırmış, %95,29 oranında sahtekarlık çağrılarını tespit etmiştir.

Çizelge 6.23 Model 1 ve Karma 1 tahmin farklılıkları ve gerçek değerler tablosu

Sıra	Gerçek Değer	Karma 1	Model 1	Doğru Tahmin	Sıra	Gerçek Değer	Karma 1	Model 1	Doğru Tahmin
29	0	1	1	Yok	2747	0	1	0	Model 1
34	0	1	1	Yok	3026	0	1	0	Model 1
38	0	1	0	Model 1	3249	0	1	0	Model 1
67	1	0	1	Model 1	3405	1	0	0	Yok
69	0	1	1	Yok	3540	0	1	0	Model 1
85	0	1	0	Model 1	3684	0	1	0	Model 1
89	0	1	0	Model 1	3875	0	1	0	Model 1
100	0	1	0	Model 1	3876	0	1	0	Model 1
112	0	1	0	Model 1	3993	0	1	0	Model 1
122	0	1	0	Model 1	4063	0	1	0	Model 1
155	0	1	0	Model 1	4259	0	1	0	Model 1
171	0	1	0	Model 1	4447	0	1	0	Model 1
175	0	1	0	Model 1	4449	0	1	0	Model 1
176	0	1	0	Model 1	4631	0	1	0	Model 1
196	0	0	1	Karma 1	4749	0	1	0	Model 1
198	1	0	1	Model 1	5309	0	1	0	Model 1
199	1	0	1	Model 1	5426	0	1	0	Model 1
218	0	1	0	Model 1	5609	0	1	0	Model 1
232	0	1	0	Model 1	5774	0	1	0	Model 1
233	0	1	0	Model 1	5887	0	1	0	Model 1
236	0	1	0	Model 1	5971	0	1	0	Model 1
260	0	1	0	Model 1	6010	1	0	1	Model 1
273	0	1	0	Model 1	6031	0	1	0	Model 1
277	0	1	0	Model 1	6089	0	1	0	Model 1
279	0	1	0	Model 1	6336	0	1	0	Model 1
347	0	1	0	Model 1	6337	0	1	0	Model 1
348	0	1	0	Model 1	6454	0	1	0	Model 1
399	0	1	0	Model 1	6465	0	1	0	Model 1
401	0	1	0	Model 1	6573	0	1	0	Model 1
408	0	1	0	Model 1	6575	0	1	0	Model 1
409	0	1	0	Model 1	6577	0	1	0	Model 1
419	1	0	1	Model 1	6694	1	0	1	Model 1
444	0	1	1	Yok	6746	0	1	0	Model 1
524	1	0	0	Yok	6749	0	1	0	Model 1
543	1	0	0	Yok	6940	0	1	0	Model 1
595	0	1	1	Yok	7096	0	1	0	Model 1
656	0	1	0	Model 1	7122	0	1	0	Model 1
673	1	0	1	Model 1	7216	0	1	0	Model 1
694	0	1	0	Model 1	7223	0	1	0	Model 1

Çizelge 6.23 Model 1 ve Karma 1 tahmin farklılıkları ve gerçek değerler tablosu (devam)

835	0	1	0	Model 1	7275	1	1	0	Karma 1
836	0	1	0	Model 1	7382	0	1	0	Model 1
839	0	1	0	Model 1	7563	0	1	0	Model 1
981	0	1	0	Model 1	7901	0	1	0	Model 1
1096	0	1	0	Model 1	8218	0	1	0	Model 1
1187	0	1	0	Model 1	8339	0	1	0	Model 1
1243	0	1	0	Model 1	8341	0	1	0	Model 1
1372	0	1	0	Model 1	8344	0	1	0	Model 1
1377	0	1	0	Model 1	8706	0	1	0	Model 1
1478	0	1	0	Model 1	8796	0	1	0	Model 1
1751	0	1	0	Model 1	8865	0	1	0	Model 1
1861	0	1	0	Model 1	8869	1	0	1	Model 1
1998	0	1	0	Model 1	8875	1	0	1	Model 1
2542	0	1	0	Model 1	8876	1	0	1	Model 1

7. SONUÇ

Bu çalışmada, telekomünikasyon sektöründe hem operatörler hem de aboneler için maddi ve manevi zararlara neden olan sahtekarlık çağrılarını, abonelerin ödeme, sosyal bilgileri ve gerçek zamanlı sinyalleşme detayları ya da arama detay kayıtları ile beraber kullanarak, tespit edebilen iki ayrı yapay sinir ağı tasarlanmıştır. Tasarlanan bu ağlar aynı test verileri kullanarak KEYK algoritması ve farklı yapıdaki daha az işlemci gücü ihtiyacı olan ağlar ile karşılaştırılmıştır.

Bu çalışmada bir telekomünikasyon şirketi abonelerinin verileri maskelenip kodlanarak kullanılmıştır. Çalışmada beş veri seti kullanılmıştır. İlk veri seti abonelerin konuşma detay kayıtlarındaki bilgiler, abonelerin ödeme bilgileri ve sosyal bilgileri ile oluşturulmuştur. Model 1 ağ eğitimlerinde kullanılmıştır. İkinci veri seti ses sinyalleşme davet mesajları içeriğindeki bilgiler, abone ödeme bilgileri ve sosyal bilgileri ile oluşturulmuştur. Model SIP1 ağının eğitiminde kullanılmıştır. Üçüncü veri seti Agresif 1 ağı eğitimlerinde kullanılmak amacıyla ilk veri seti kullanılarak yoğun sahtekarlık aramaları içerecek şekilde oluşturulmuştur. Dördüncü veri seti ise birinci veri setinin normal aramaları ağırlıklı olacak şekilde düzenlenmesiyle elde edilmiştir. Tolerans 1 ağının eğitimlerinde kullanılmıştır. Son veri seti ise eğitilen ağların test edilmesi amacıyla düzenlenmiştir. En geniş veri seti olan bu sette 361 sahtekarlık araması ve 8515 normal arama olmak üzere toplam 8876 arama içermektedir.

Tasarlanan ağların hiper parametreleri, örgü arama yöntemi ile seçilmiştir. Sonrasında ağ performansları seyreltme düzenleme yöntemi ile farklı katsayılarda denenmiş en iyi test performansı sağlayan ağ parametreleri karşılaştırma testlerinde kullanılmak üzere kaydedilmiştir.

İlk karşılaştırma çalışmasında; ÇDK verileri, abone ödeme bilgileri ve abone sosyal veriler kullanarak oluşturulan veri seti ile eğitilmiş ağ (Model 1), tasarlanan KEYK algoritması ile karşılaştırılmıştır. 2663 arama içeren veri seti ile yapılan test sonucunda Model 1; %99,96 doğruluk, %100 hassaslık sağlamıştır. KEYK algoritması, aynı veri seti ile test edilmiş ve

%98,69 doğruluk, %77,68 hassaslık sağlamıştır. Tasarlanan model tüm performans metriklerinde daha yüksek sonuçlar almıştır.

İkinci karşılaştırma çalışmasında; gerçek zamanlı sinyalleşme bilgileri, abone ödeme bilgileri ve abone sosyal verileri kullanılarak oluşturulan 2663 arama içeren veri seti ile yapılan test sonucunda Model SIP1 %98,49 doğruluk, %94,64 hassaslık sağlamıştır. KEYK algoritması, aynı veri seti ile test edilmiş ve %98,49 doğruluk, %75,89 hassaslık sağlamıştır. KEYK algoritması, ilk karşılaştırma testlerine yakın bir performans ile yüksek bir doğruluk yakalamış olmasına rağmen, Model SIP1 ağı aynı doğruluğu daha yüksek bir hassaslık ile elde etmiştir.

Üçüncü karşılaştırma çalışması Model 1 ve Model SIP1 performansları en geniş veri seti olan test veri seti ile yapılmıştır. 8876 arama kaydı ile yapılan testlerde Model 1 %99,9 doğruluk, %98,89 hassaslık sağlamıştır. Model 1 eğitim veri setine göre daha az bilgi içeren veri seti ile eğitilen Model SIP1 ağı; %98,38 doğruluk, %93,91 hassaslık sağlamıştır. Model SIP1 ağının, 8876 arama kaydı için tahmin süresi 0,112 saniye olarak ölçülürken, Model 1 ağının aynı veriler için tahmin süresi 0,115 saniye olarak ölçülmüştür.

Dördüncü karşılaştırma Model 1 ve Karma 1 ağları arasında yapılmıştır. Fazla sayıda nörondan oluşan Model 1 ağının başarısının düşük işlemci gücü ile çalışan iki adet ağ ile sağlanıp sağlanamayacağı test edilmiştir. 8876 arama kaydı ile yapılan testlerde Model 1 %99,9 doğruluk, %98,89 hassaslık sağlamıştır. Karma 1 ağı 0,7 eşik değerinde, test setinde bulunan tüm çağrıları %99,35 doğruluk oranı ile sınıflandırmış, %95,29 oranında sahtekarlık çağrılarını tespit etmiştir. Karma 1 ağı; Model 1 performansını yakalayamamış olmasına rağmen, daha geniş eğitim setleri ile eğitilmesi durumunda aynı performansı daha esnek konfigürasyon ve değerlendirme imkanı ile sağlayabileceği düşünülmektedir.

Tasarlanan modellerin eğitim süreleri tasarlanan ağa göre değişkenlik gösterebilmektedir. Model 1 ağının 200 döngü ile eğitilmesi 17,398 saniye, aynı döngü sayısı ile Model SIP1 ağı 15,31 saniyede eğitilmiştir. Karma 1 ağını oluşturan iki ağın eğitim süreleri ise 7,418 saniye ve 12,055 saniye olarak ölçülmüştür.

Gerçek zamanlı sinyalleşme verileri kullanılarak eğitilen ağın; güvenilir bir sistem olarak kullanılması uygun olmasa da sahtekarlık yönetim sistemleri ile beraber çalışarak erken tespit ile maddi kayıpların operasyon kalitesine bağlı olarak %50 ile %80 arasında azaltılabileceği değerlendirilmektedir.

Sahtekarlık yönetim sistemlerinde dinamik ve hızlı öğrenen bir yapı olan yapay zeka uygulamaları kullanımıyla; kendini hızla yenileyip farklı yöntemlerle ortaya çıkan çağrı sahtekarlıklarının büyük oranda tespit edilmesinin sağlanacağı düşünülmektedir.

KAYNAKLAR

Anonymous. 2015. CS231n Convolutional Neural Networks for Visual Recognition. Web Sitesi: http://cs231n.github.io/classification/#nn. Erişim Tarihi: 2018.

Anonymous. 2016. A Complete Guide to K-Nearest-Neighbors with Applications in Python and R. 7. Web Sitesi: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/. Erişim Tarihi: 2018.

Anonymous. 2018. Web Sitesi: https://transnexus.com/resources/telecom-industry-topics/telecom-fraud-guide. Erişim Tarihi: 2018.

Anonymous. 2018. Web Sitesi: https://en.wikipedia.org/wiki/Deep_learning. Erişim Tarihi: 2018.

Anonymous. 2018. 20 3. Web Sitesi: https://pyblog.xyz/neural-network-in-a-nutshell/. Erişim Tarihi: 2018.

Anonymous. 2018. 2017 Global Fraud Loss Survey. Survey, Communications Fraud Control Association.

Anonymous. 2018. Keras Documentation. Web Sitesi: https://keras.io/. Erişim Tarihi: 2018.

Anonymous. 2018. Wikipedia. 5. Web Sitesi: https://en.wikipedia.org/wiki/TensorFlow. Erişim Tarihi: 2018.

Aoullay, Amine. 2018. Towards Data Science. 6. Web Sitesi: https://towardsdatascience.com/how-to-use-noise-to-your-advantage-5301071d9dc3. Erişim Tarihi: 2018.

Arora, Nipun. 2016. GeeksProgramming. 8. Web Sitesi: https://www.geeksprogramming.com/what-is-machine-learning-an-introduction/. Erişim Tarihi: 2018.

Ataseven, Burçin. 2013. YAPAY SİNİR AĞLARI İLE ÖNGÖRÜ MODELLEMESİ. İstanbul Kültür Üniversitesi, İstanbul.

Bottou, L. 2010. Large-Scale Machine Learningwith Stochastic Gradient Descent. Proceedings of COMPSTAT'2010 içinde, yazan Y. Lechevallier ve G. Saporta. Physica-Verlag HD.

Bruijne, Marleende. 2016. Machine learning approaches in medical image analysis: From detection to diagnosis. Medical Image Analysis, 10: 94-97.

Burge, Peter, ve John Shawe-Taylor. 2001. An Unsupervised Neural Network Approach to. Journal of Parallel and Distributed Computing, 5: 915-925.

Chen, Chenyi , Ari Seff, Alain Kornhauser, ve Jianxiong Xiao. 2015. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. The IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE. 2722-2730.

Duchi, John, Elad Hazan, ve Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. The Journal of Machine Learning Research, 2: 2121-2159.

Fawcett, Tom , ve Foster Provost. 1997. Adaptive Fraud Detection. Data Mining and Knowledge Discovery, 9: 291-316.

FIÇICI, Cansel ÖĞRETMENOĞLU. 2016. MR GÖRÜNTÜLERİNDE BAŞ BÖLGESİNE AİT ANORMAL YAPILARIN BELİRLENMESİ VE HACİMLERİNİN HESAPLANMASI. Yüksek Lisans Tezi, Ankara: Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ElektronikMühendisliği Anabilim Dalı.

Grover, Prince. 2018. Choosing the right loss function for fitting a model. 6. Web Sitesi: https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0. Erişim Tarihi: 2018.

Guo, Guodong , S. Z. Li, ve Kapluk Chan. 2000. Face recognition by support vector machines. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition. Grenoble: IEEE. 196-201.

Harrington, Peter. 2012. Machine Learning in Action. Greenwich: Manning Publications Co.

Hilas, Constantinos , ve John N. Sahalos. 2005. User Profiling for Fraud Detection.

Hilas, Constantinos S., Paris A. Mastorocostas, ve Ioannis T. Rekanos. 2015. Clustering of Telecommunications User Profiles forFraud Detection and Security Enhancement in LargeCorporate Networks: A case Study. Applied Mathematics & Information Sciences, 7: 1709-1718.

Jain, Vanita. 2017. Perspective analysis of telecommunication fraud detection using data stream analytics and neural network classification based data mining. International Journal of Information Technology, 9: 303-310.

Kabalcı, Ersan. 2013. Yapay Sinir Ağları. 9. Web Sitesi: https://ekblc.files.wordpress.com/2013/09/ysa.pdf. Erişim Tarihi: 2018.

Kale, Satyen, Ravi Kumar, ve Sergei Vassilvitskii. 2011. Cross-Validation and Mean-Square Stability. Innovations in Computer Science - ICS 2010. Beijing: Tsinghua University.

Kingma, Diederik P. , ve Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. Interna-tional Conference on Learning Representations. San Diego. 1-13.

Liu, Xuedong, Jingjing Jia, ve Jie Yang. 2017. Improved image coloring algorithm based on LapRLS. 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Chongqing: IEEE. 2252-2255.

Lotter, Sebastian. 2018. Web Sitesi: http://machinelearning.tf.fau.de. 6. Erişim Tarihi: 2018.

Manunza , Luca, Stefano Marseglia , ve Simon Pietro Romano. 2017. Kerberos: A real-time fraud detection system for IMS-enabled VoIP networks. Journal of Network and Computer Applications, 15 2: 22-34.

Mohammed, Abdul Adeel, Rashid Minhas, Q.M Jonathan Wu, ve Maher A. Sid-Ahmed. 2011. Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognition, 10-11: 2588-2597.

Nielsen, Michael A. 2015. Neural Networks and Deep Leanring. Determination Press.

Öztemel, Ercan. 2003. Yapay Sinir Ağları. İstanbul: Papatya Yayıncılık Eğitim.

Phua, Clifton, Damminda Alahakoon,, ve Vincent Lee. 2004. Minority Report in Fraud Detection: Classification of Skewed Data. ACM SIGKDD Explorations Newsletter, 6: 50-59.

Qian, Ning. 1999. On the momentum term in gradient descent learning algorithms. Neural networks : the official journal of the International Neural Network Society, 2: 145-151.

Rao, Anirudh. 2018. Edureka. Web Sitesi: https://www.edureka.co/blog/python-libraries/. Erişim Tarihi: 2018.

Rojas, Raul. 1996. Neural Networks - A Systematic Introduction. Berlin, New-York: Springer-Verlag.

Rosaen, Karl. 2016. Web Sitesi: http://karlrosaen.com/ml/learning-log/2016-06-20/. Erişim Tarihi: 2018.

Ruder, Sebastian. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747.

Ruiz-Agundez, Igor, Yoseba K. Penya, ve Pablo Garcia Bringas. 2010. Fraud Detection for Voice over IP Serviceson Next-Generation Network. 4th IFIP WG 11.2 International Workshop on Information SecurityTheory and Practices: Security and Privacy of Pervasive Systems and Smart Devices (WISTP). Passau. 199-212.

Sallab, Ahmad El, Mohammed Abdou, Etienne Perot, ve Senthil Yogamani. 2017. Deep Reinforcement Learning framework for Autonomous Driving. Autonomous Vehicles and Machines. Society for Imaging Science and Technology. 70-76.

Sayad, Saed. 2010. An Introduction to Data Science. Web Sitesi: https://www.saedsayad.com/artificial_neural_network.htm. Erişim Tarihi: 2018.

Sharma, Sagar. 2017. Towards Data Science. 9. Web Sitesi: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. Erişim Tarihi: 2018.

Taniguchi, M., M. Haft, J. Hollmen, ve V. Tresp. 1998. Fraud detection in communication networks using neural and probabilistic methods. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98. Seattle: IEEE. 1241-1244.

Uslu, Metim. 2016. Yapay Sinir Ağları (YSA) Nedir – Bölüm 2. 5. Web Sitesi: http://veribilimci.org/yapay-sinir-aglari-ysa-nedir-bolum-2/. Erişim Tarihi: 2018.

Zakka, Kevin. 2016. A Complete Guide to K-Nearest-Neighbors with Applications in Python and R. 7. Web Sitesi: https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/. Erişim Tarihi: 2018.

Zeiler, Matthew D. 2012. ADADELTA: AN ADAPTIVE LEARNING RATE METHOD. arXiv:1212.5701.

Zhang, Jason. 2018. iData. 6. Web Sitesi: https://medium.com/idatatoday/4-train-machine-learning-model-artificial-intelligence-e164ff825d1b. Erişim Tarihi: 2018.

ÖZGEÇMİŞ

Adı Soyadı : Murat BAĞ

Doğum Yeri : ANKARA

Doğum Tarihi : 06/12/1981

Medeni Hali : Evli

Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Çankaya Anadolu Lisesi (Eylül 1992 – Haziran 1999)

Lisans : Orta Doğu Teknik Üniversitesi Elektrik ve Elektronik Mühendisliği (Eylül 2000 – Ocak 2007)

Yüksek Lisans : Ankara Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı (Eylül 2007 – Eylül 2019)

Çalıştığı Kurum/Kurumlar ve Yıl

Netmon A.Ş. (Mayıs 2007 – Eylül 2007)

Türk Telekomünikasyon A.Ş (Ekim 2007 – Nisan 2018)