T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

COPULA FONKSİYONLARINI KULLANARAK BİLGİSAYAR AĞLARINDA SALDIRI TESPİTİ

Mehmet BURUKANLI

HAZİRAN 2020

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

COPULA FONKSİYONLARINI KULLANARAK BİLGİSAYAR AĞLARINDA SALDIRI TESPİTİ

Hazırlayan
Mehmet BURUKANLI

Danışman
Dr. Öğr. Üyesi Musa ÇIBUK

Jüri Üyeleri
Prof. Dr. Sabir RÜSTEMLİ
Dr. Öğr. Üyesi Musa ÇIBUK
Dr. Öğr. Üyesi İhsan TUĞAL

HAZİRAN 2020

ONAY

Mehmet BURUKANLI tarafından hazırlanan "**Copula Fonksiyonlarını Kullanarak Bilgisayar Ağlarında Saldırı Tespiti**" adlı tez çalışması …/…/…tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Prof. Dr. Sabir RÜSTEMLİ (Başkan)	_____
Dr. Öğr. Üyesi Musa ÇIBUK (Danışman)	_____
Dr. Öğr. Üyesi İhsan TUĞAL (Üye)	_____

Bu tezin kabulü, Lisansüstü Eğitim Enstitüsü Yönetim Kurulu'nun …/…/…gün ve …/… sayılı kararı ile onaylanmıştır.

Prof. Dr. Zeki ARGUNHAN
Enstitü Müdürü

BİTLİS EREN ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI
ETİK BEYANI

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre hazırlamış olduğum "**Copula Fonksiyonlarını Kullanarak Bilgisayar Ağlarında Saldırı Tespiti**" adlı tezimin özgün bir çalışma olduğunu, tez hazırlanırken tüm aşamalarda bilimsel etik ilkelerine uygun davrandığımı, tez kapsamında sunulan tüm verileri bilimsel etik ilkelerine uygun elde ettiğimi, tezde faydalandığım tüm eserlere atıf yaptığımı ve kaynaklar kısmında bu eserleri gösterdiğimi beyan ederim. ……/……/2020

Mehmet BURUKANLI

ÖZET

COPULA FONKSİYONLARINI KULLANARAK BİLGİSAYAR AĞLARINDA SALDIRI TESPİTİ

Mehmet BURUKANLI

Yüksek Lisans Tezi

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Musa ÇIBUK
Haziran 2020, 116 sayfa

Günümüzde hızla gelişen teknolojiyle beraber, dünya üzerinde teknolojiye olan ilgi her geçen gün artmaktadır. Teknolojideki bu hızlı gelişmeler, siber saldırı, izinsiz erişim ve dijital korsanlık gibi istenmeyen birçok saldırıyı da beraberinde getirmektedir. Bu tür saldırıları engellemek için sıklıkla saldırı tespit sistemlerinden faydalanılmaktadır. Bu tez çalışmasında, günümüzde en çok kullanılan makine öğrenme sınıflandırıcıları ile copula tabanlı sınıflandırıcılar kullanılarak saldırı tespiti gerçekleştirilmiştir. Makine öğrenme sınıflandırıcıları olarak; Karar Ağaçları, Topluluk Öğrenme ve Destek Vektör Makineleri sınıflandırıcıları tercih edilmiştir. Bu üç sınıflandırma tekniği kullanılarak KDD'99 veri seti üzerinde sınıflandırma işlemi gerçekleştirilmiştir.

Copula tabanlı olarak da; gumbel, independent, clayton, gaussian, student's-t ve frank sınıflandırıcıları tercih edilmiştir. Bu sınıflandırıcılar kullanılarak KDD'99 veri seti üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Sınıflandırma aşamasında 10-kat çapraz doğrulama tekniği kullanılmış olup, KDD'99 veri seti üzerinde en iyi başarım oranı %99.41 ile gaussian copula tabanlı sınıflandırıcı elde etmiştir. Sonuç olarak, copula tabanlı sınıflandırıcıları saldırı tespitinde etkili bir başarıma ulaştığı gözlemlenmiştir.

Anahtar kelimeler: Saldırı Tespiti, Copula Fonksiyonları, KDD'99 Veri Seti, Naive Bayes Sınıflandırıcısı, Topluluk Öğrenme

ABSTRACT

INTRUSION DETECTION USING COPULA FUNCTIONS IN COMPUTER NETWORKS

Mehmet BURUKANLI

Master Thesis

Bitlis Eren University Graduate Education Institute
Department of Electrical and Electronics Engineering
Supervisor: Asst. Prof. Dr. Musa ÇIBUK
June 2020, 116 pages

Today, with the rapidly developing technology, the interest in technology in the world is increasing day by day. These rapid developments in technology has brought many undesired attacks such as cyber attack, unauthorized access and digital piracy. Intrusion detection systems have been often used to prevent such attacks. In this study, attack detection has been carried out using the most used machine learning classifiers and copula-based classifiers. As machine learning classifiers; Decision Trees, Ensemble Learning and Support Vector Machines classifiers have been preferred. Classification has been performed on the KDD'99 data set using these three classification techniques.

As Copula-based; gumbel, independent, clayton, gaussian, student's-t and frank classifiers have been preferred. Using these classifiers, the classification process has been carried out on the KDD'99 data set. In the classification stage, 10-fold cross-validation technique has been used, and the best performance rate has obtained gaussian copula based classifier with 99.41% on the KDD'99 data set. As a result, it has been observed that copula based classifiers have achieved an effective success in intrusion detection.

Keywords: Intrusion Detection, Copula Functions, KDD'99 Dataset, Naive Bayesian Classifier, Ensemble Learning

TEŞEKKÜR

Yoğun geçen tez çalışması sırasında, tez konusunun belirlenmesinden başlayarak son aşamaya kadar her türlü bilgi, teşvik ve deneyimlerini benden esirgemeyen danışman hocam sayın Dr. Öğr. Üyesi Musa ÇIBUK'a şükranlarımı sunarım. Yüksek lisans eğitimim boyunca yardımlarından dolayı Elektrik-Elektronik Mühendisliği Anabilim dalı başkanı sayın Prof. Dr. Sabir RÜSTEMLİ hocama, copulalar konusunda bizden desteğini esirgemeyen Fen Edebiyat Fakültesi İstatistik bölümü öğretim üyesi sayın Doç. Dr. Ayşe METİN KARAKAŞ hocama teşekkürlerimi sunarım.

Bu günlere gelmemde büyük emekleri olan başta ailem olmak üzere çalışma temposu boyunca yanımda olan eşime ve çocuklarıma teşekkürlerimi sunarım.

ÖNSÖZ

Bu tez çalışmasında, son zamanlarda çok popüler olan copula fonksiyonları ve makina öğrenme algoritmaları kullanılarak saldırı tespitini gerçekleştirilmiştir. Copula fonksiyonları olarak literatürde sıklıkla tercih edilen gaussian, student's-t, clayton, frank, gumbel ve independent copulaları kullanılmıştır. Benzer sekilde yine makina öğrenme algoritmaları olarak da literatürde çokça tercih edilen karar ağaçları, topluluk öğrenme, destek vektör makinesi algoritmaları kullanılmıştır.

Veri seti olarak da; literatürde en çok kullanılan veri setlerinden biri olan KDD'99 veri seti kullanılmıştır. Seçilen her bir copula ailesi ve her bir makine öğrenme algoritması KDD'99 veri seti üzerinde uygulanarak başarım sonuçları elde edilmiştir. Eğitim ve test işlemleri MATLAB ortamında gerçekleştirilmiştir.

İÇİNDEKİLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

ÇİZELGE **Sayfa**

ŞEKİLLER DİZİNİ

ŞEKİL **Sayfa**

SİMGELER DİZİNİ

F	Ortak Dağılım Fonksiyonu
$F_1(x_1), \dots, F_n(x_n)$	Tek Değişkenli Marjinal Dağılım Fonksiyonu
$f(x_1, \dots, x_n)$	Çok Değişkenli Yoğunluk Fonksiyonu
$C(.,.), c(.,.)$	Copula Fonksiyonu
τ	Kendall Tau Katsayısı
ρ	nxn Türünde Korelasyon Matrisi
θ	Copula Parametresi
$\hat{\theta}_i$	Marijin Dağılım Parametresi
$\in$	Elemanı
$\forall$	Tümü İçin
R	Gerçek Sayılar
α	Alfa Parametresi, Copula Parametresi
$\hat{\alpha}_{IFM}$	IFM Metodu Kulanarak Copula Parametre Tahmini
$\hat{\alpha}_{CML}$	CML Metudu Kulanarak Copula Parametre Tahmini
Σ	Toplam Sembolü
Π	Çarpım Sembolü
∂	Türev
λ	Kuyruk Bağımlılığı (Lambda)
λ_u	Üst Kuyruk Bağımlılığı
λ_l	Alt Kuyruk Bağımlılığı
π	Pi Sayısı
φ	Arşimedyan Copulalarda Üreteç Fonksiyonu
$D()$	Debye Fonksiyonu
log	Logaritma
lim	Limit
exp	Üstel Fonksiyon
W	Yapay Sinir Ağındaki Ağırlıklar
ξ	Pozitif Gevşeklik Değişkeni
$K(.,.)$	Çekirdek Fonksiyonu
p_i	i Sınıfındaki Örneklerin Sayısı
U	Öz vektör

Λ	Öz değer
$f^{mRMR}()$	mRMR Fonksiyonu
I	Birim Fonksiyonu
$I(.,.)$	Ortaklık Bilgisi Fonksiyonu
$H(X)$	X'nin entropisi
$H(x,y)$	Ortak Dağılım Fonksiyonu
$f^{mRMR\_mid}()$	Ortak Bilgi Farkı Fonksiyonu
$f^{mRMR\_miq}()$	Ortak Bilgi Oranı Fonksiyonu
X,Y	Rastgele İki Değişken
Φ_ρ	n-Boyutlu Normal (Gaussian) Dağılım Fonksiyonu
$C_\rho^{Gaussian}$	Gaussian Copula Fonksiyonu
$N(x)$	Tek Değişkenli Normal Dağılım Fonksiyonu
$C_{v,\rho}^{Student's-t}$	Student's-t Copula Fonksiyonu
v	Serbestlik Derecesi
$t_v(x)$	Tek Değişkenli Student's-t Dağılım Fonksiyonu
δ_C	C, Copulasının Köşegen Kesiti
$C_\theta^{Clayton}$	Clayton Copula Fonksiyonu
C_θ^{Gumbel}	Gumbel Copula Fonksiyonu
C_θ^{Frank}	Frank Copula Fonksiyonu
$\Pi^{Independent}$	Independent (Independence) Copula Fonksiyonu
$l(\theta)$	Log-Olasılık Fonksiyonu
Θ	Parametre Uzayı
$\hat{\theta}_{EML/ML}$	Tam Maksimum Olasılık/Maksimum Olasılık Parametre Tahmin Edicisi
$\hat{\theta}_{IFM}$	Marjinleri için Çıkarım İşlevleri Parametre Tahmin Edicisi
$\hat{\theta}_{CML}$	Standart Maksimum Olasılık Parametre Tahmin Edicisi
$\hat{F}_n(.)$	Deneysel Dağılım Fonksiyonu
$\hat{f}_i(x)$	Çekirdek Yoğunluk Tahmin Edicisi
$\widehat{S_D}$	Maksimum Sonsal Olasılık Sınıflandırıcısı
$P(S_d\|X)$	S_d Sınıfının Sonraki Olasılığı
$P(S_d)$	S_d Sınıfının Önceki Olasılığı
NET	Yapay Sinir Ağlarında Toplama Fonksiyonu

KISALTMALAR DİZİNİ

STS	Saldırı Tespit Sistemi	Intrusion Detection System (IDS)
YSA	Yapay Sinir Ağları	Artificial Neural Networks (ANN)
ÇKA	Çok Katmanlı Algılayıcı	Multi Layer Perceptron (MLP)
TBA	Temel Bileşen Analizi	Principal Component Analysis (PCA)
SA	Sinir Ağlar	Neural Networks (NN)
KA	Karar Ağacı	Decision Tree (DT)
TÖ	Topluluk Öğrenme	Ensemble Learning (EL)
DVM	Destek Vektör Makinesi	Support Vector Machine (SVM)
CML	Standart Maksimum Olasılık	Canonical Maximum Likelihood
IFM	Marjinleri için Çıkarım İşlevleri	Inference Functions for Margins
EML	Tam Maksimum Olasılık	Exact Maximum Likelihood
ML	Maksimum Olasılık	Maximum Likelihood
MAP	Maksimum Sonsal Olasılık	Maximum Aposteriori Probability
NBC	Naive Bayes Sınıflandırıcısı	Naive Bayesian Classifier (NBC)
MID	Ortak Bilgi Farkı	Mutual Information Difference
MIQ	Ortak Bilgi Oranı	Mutual Information Quotient
CLT	Sınıflandırma Öğrenme Aracı	Classification Learner Toolbox
GYA	Geri Yayılım Algoritması	Backpropagation Algorithm (BA)
MSE	Ortalama Kare Hatası	Mean Squared Error
RTF	Radyal Tabanlı Fonsiyon	Radial Basis Function (RBF)
GA	Genetik Algoritma	Genetic Algorithm
TÜFE	Tüketici Fiyatları Endeksi	Consumer Price Index
DoS	Hizmet Engelleme	Denial of Service
DARPA	Savunma İleri Araştırma Projeleri Ajansı	Defense Advanced Research Projects Agency
mRMR	Minimum Fazlalık Maksimum İlişkili Özellik Seçimi	Minimum Redundancy Maximum Relevance Feature Selection
R2L	Admin Hesabını Ele Geçirerek Yerel Ağda Oturum Açma	Remote To Local
U2R	Kullanıcı Hesabını Admin Hesabına Yükseltme	User To Root

KDD10	Doğrulanmış KDD’99 Veri Setinin Yüzde 10 (%10)’ luk Kısmı	KDDCUPCORRECTED (%10)
KDD100	Doğrulanmış KDD’99 Veri Setinin Yüzde 100 (%100)’lük Kısmı	KDDCUPCORRECTED (%100)
KDDTEST	Doğrulanmış KDD’99 Test Veri Seti	KDDCUPTEST

1. GİRİŞ

Günümüzde internet, hem kişisel hemde iş ilişkileri arasındaki bilgi akışını sağlayan önemli bir iletişim aracıdır. Bu iletişim aracı beraberinde güvenlik tehlikelerini de getirmiştir. Özellikle, internet üzerinden yapılan e-ticaret uygulamaları ciddi oranda tehlikeli saldırılara maruz kalmaktadır. Bu saldırılar, kritik iş uygulamalarında iş gücü, zaman ve ürün kaybına yol açarak şirketlerin ciddi anlamda zarara uğratılmasına neden olmaktadır [1]. Örneğin; çalışanların hataları, bilgisayar virüsleri ve zararlı yazılımlar sadece bunlardan bir kaçıdır. Yapılan saldırılar sonucunda, bilgi kayıpları yaşanmakta ve gizli kalması gereken bilgiler ifşa edilebilmektedir. İnternetteki güvenlik açıkları, web tabanlı şirketlere ve kamu hizmetlerine büyük zarar verebilmektedir. Bu yüzden şirketler ve kamu hizmetleri yürüten kurumlar, güvenlik tedbirlerini her geçen gün arttırmakta ve yeni tehditlere karşı önlem almak amacıyla daha büyük yatırımlar yapmak zorunda kalmaktadırlar [1, 2]. Bundan dolayı, bilgisayar sistemlerinin güvenliğini sağlayan araçlar gittikçe önem kazanmakta ve özellikle de Saldırı Tespit Sistemlerine (STS) duyulan önem her geçen gün artmaktadır.

STS, ağ üzerinden yapılan her türlü saldırılara karşı bilişim sistemlerinin korunmasına yardımcı olup, uyarı niteliği taşıyan yazılım veya donanım bileşenlerinin tümüne denilmektedir [2, 3]. STS kullanılarak, ağ üzerinden yapılan saldırılar tespit edilebilmekte ve ilgili mekanizmalar harekete geçirilerek engellenebilmektedir. STS'leri gerçekleştirmek için birçok yöntem bulunmaktadır. Bu yöntemlerden bazıları, Yapay Sinir Ağları (YSA), Destek Vektör Makineleri (DVM), Karar Ağacı (KA) ve Topluluk Öğrenme (TÖ) olarak sıralanabilir. Bu yöntemler dışında copula fonksiyonları gibi yöntemlerde uygulanmaya başlanmıştır [4]. Bu tez çalışmasında STS alanı için yeni bir yaklaşım olan copula fonksiyonları kullanılarak saldırı tespiti yapılmıştır. Bu çalışmada esinlenilen sınıflandırma algoritması ilk olarak R.Salinas-Gutierrez ve ark [5] tarafından önerilmiştir. Gauss copula tabanlı olan bu sınıflandırma algoritması daha sonra M.Scavnicky [6] tarafından genelleştirilmesi sağlanarak copula tabanlı bir sınıflandırma algoritması elde edilmiştir. Bu tez çalışmasında M.Scavnicky [6] tarafından genelleştirilmesi yapılan copula tabanlı sınıflandırma yaklaşımı temel alınarak geliştirilen algoritmalar kullanılmıştır. Literatürde saldırı tespiti ile ilgili birçok çalışma yapılmıştır. Yapılan bu çalışmalardan bazılarını şöyle sıralayabiliriz.

B.W.Masduki ve ark [2], yaptıkları çalışmada DVM kullanarak saldırı tespiti yapılmışlardır. R2L saldırılarında başarım oranı %96.08 olarak gerçekleşmiştir.

Ş.Sağıroğlu ve ark [3], yaptıkları çalışmada YSA tabanlı zeki bir STS geliştirmişlerdir. Geliştirdikleri zeki STS, oldukça başarılı sonuçlar vermiştir. KDD'99 veri setini kullanarak zeki

STS'yi test etmişlerdir. KDD'99 veri setinden 65536 örnek kullanmışlardır. Elde ettikleri en yüksek başarım oranı % 97.92 ve en düşük başarım oranı ise %81.93 olarak gerçekleşmiştir.

S.Mukkamala ve ark [7], yaptıkları çalışmada YSA ve DVM kullanarak saldırı tespiti yapılmışlardır. Yaptıkları çalışmada, bir sistemdeki kullanıcı davranışını tanımlayan faydalı örüntüleri veya özellikleri keşfetmek ve gerçek zamanlı olarak anormallikleri ve bilinen saldırıları tanıyabilen sınıflandırıcılar oluşturmak için bir dizi özellikler kullanmışlardır. YSA ve DVM yöntemlerini kullanarak 6980 test verisi üzerinde performansları kıyaslamışlardır. Her iki yöntemin başarım oranları %99'nün üstünde gerçekleşmiştir.

M.Moradi ve ark [8], yaptıkları çalışmada YSA kullanarak çevrim dışı (off-line) saldırı tespiti yapmışlardır. Yaptıkları daha önceki çalışmalarda normal ve saldırı olmak üzere iki tip sınıf kullanmışken, bu çalışmada ise birçok saldırı tipi kullanmışlardır. Veri setini YSA kullanarak eğitmişlerdir. YSA'da iki adet gizli katman kullandıklarında yaklaşık %91 oranında başarım elde ederken, bir adet gizli katman kullandıklarında ise %87 oranında başarım elde etmişlerdir.

S.Mukkamala ve ark [9], yaptıkları çalışmada DVM ve YSA kullanarak Defense Advanced Research Projects Agency (DARPA) veri seti ile eğitim yapmışlardır. DVM, YSA'dan daha iyi doğruluk oranı elde etmiştir. SVM yaklaşık %99 başarım oranına sahip iken YSA'da ise en düşük başarım oranı %48 en yüksek başarım oranı %95 olarak gerçekleşmiştir.

H.A.Sonawane ve ark [10], yaptıkları çalışmada Sinir Ağları (SA) ve SA tabanlı temel bileşen analizi (TBA) olmak üzere iki yöntem önermişlerdir. SA tabanlı TBA metodu KDD'99 veri setinin birkaç özelliğini kullanırken, SA metodunda ise KDD'99 veri setinin bütün özellikleri kullanmıştır. Bu iki metodun kıyaslanmaları yaparak, SA'nın TBA'ya göre daha iyi sonuç verdiğini gözlemlemişlerdir. SA'nın en yüksek başarım oranı %90.20 olarak elde etmişlerdir.

M.Govindarajan ve ark [11], yaptıkları çalışmada topluluk sınıflandırıcı (RTF+DVM) kullanarak saldırı tespiti yapmışlardır. Radyal tabanlı fonksiyon (RTF) ve DVM kullanarak topluluk sınıflandırıcısı elde etmişlerdir. RTF+DVM'nın en iyi başarım oranı %85.19 olarak gerçekleşmiştir.

A.Dastanpour ve ark [12], yaptıkları çalışmada DVM, YSA ve Genetik Algoritma (GA) algoritmaları kullanarak saldırı tespiti yapmışlardır. GA algoritmasını DVM üzerine uyguladıklarında, KDD'99 veri setinin 24 özelliği kullanarak %100 başarım elde ederlerken, GA algoritması YSA üzerine uyguladıklarında ise KDD'99 veri setinin 18 özelliğinde %100 başarım elde etmişlerdir. YSA+GA algoritması daha az özellikle daha iyi performans elde etmiştir.

K.A.Jalil ve ark [13], yaptıkları çalışmada SA, DVM ve KA algoritmaları kullanarak saldırı tespiti yapmışlardır. Bu algoritmaları birbirleriyle kıyaslamışlardır. En iyi performansı %99.70 ile KA'ları algoritmasından olan J48 algoritması elde etmiştir.

V.A.Golovko ve ark [14], yaptıkları çalışmada TBA, devirdaim SA ve çok katmanlı algılayıcı (ÇKA) ağ modelleri kullanarak saldırı tespiti yapmışlardır. Model 1, Model 2 ve Model 3 olarak 3 adet model önermişlerdir. Önerdikleri modellerden model 3, %93.21 başarım oranı ile en iyi performansı elde etmiştir.

W.Wang ve ark [15], yaptıkları çalışmada TBA kullanarak saldırı tespiti yapmışlardır. En iyi başarım oranı %98.80 olarak elde etmişlerdir.

S.Kumar ve ark [16], yaptıkları çalışmada YSA kullanarak saldırı tespiti yapmışlardır. KDD'99 veri seti üzenindeki başarım oranı %91.90 olarak gerçekleşmiştir. Eğitim için 494021 örnek ve test için 311027 örnek veri kullanmışlardır.

F.Haddadi ve ark [17], yaptıkları çalışmada iki katmanlı ileri beslemeli sinir ağlar (İKİBSA) kullanarak saldırı tespiti gerçekleştirmişlerdir. Dataset1 ve Dataset2 olmak üzere iki adet veri seti kullanmışlardır. Dataset1 veri seti 19070 örnekten ve Dataset2 veri seti ise 34070 örnekten oluşmaktadır. Dataset1 veri seti üzerindeki en iyi başarım oranı %99 ve Dataset2 veri seti üzerindeki en iyi başarım oranı %99.10 olarak elde etmişlerdir.

J.Esmaily ve ark [18], yaptıkları çalışmada KA ve YSA kullanarak saldırı tespiti yapmışlardır. Ayrıca, KA ve YSA algoritmalarını birbirleriyle kıyaslamışlardır. KDD'99 veri seti üzerinde; YSA, %99.71 ile daha iyi sonuç vermiştir. KA algoritmasının başarım oranı ise %97.93 olarak gerçekleşmiştir.

Y.B.Bhavsar ve ark [19], yaptıkları çalışmada DVM kullanarak saldırı tespiti yapmışlardır. Normalde başarım oranı %94.18 iken 10 kat çapraz doğrulama ve RBF çekirdeği kullanarak başarım oranı %98.57'ye çıkarmışlardır.

G.Poojitha ve ark [20], yaptıkları çalışmada YSA kullanarak saldırı tespiti yapmışlardır. KDD'99 veri setinden eğitim için 6363 örnek ve test için 6360 örnek olmak üzere toplamda 12723 örnek kullanmışlardır. Başarım oranı % 94.93 olarak gerçekleşmiştir.

J.Shum ve ark [21], yaptıkları çalışmada SA kullanarak saldırı tespiti yapmışlardır. Başarım oranı bilinen saldırılarda (known attack) %100 iken, bilinmeyen (unknown attack) saldırılarda ise %76 olarak gerçekleşmiştir.

I.Ahmad ve ark [22], yaptıkları çalışmada ileri beslemeli sinir ağlar (İBSA) kullanarak saldırı tespiti yapmışlardır. Eğitim ve test işlemlerinde başarım oranı %92 üstünde gerçekleşmiştir.

K.M.Ali ve ark [23], yaptıkları çalışmada YSA kullanarak saldırı tespiti yapmışlardır. Başarım oranı % 97.27 olarak gerçekleşmiştir.

E.Hodo ve ark [24], yaptıkları çalışmada YSA kullanarak saldırı tespiti yapmışlardır. Eğitim için 2313 örnek, doğrulama ve test için 496 örnek kullanmışlardır. Başarım oranı % 99.40 olarak gerçekleşmiştir.

L.P. Dias ve ark [25], yaptıkları çalışmada YSA kullanarak saldırı tespiti yapmışlardır. KDD'99 veri seti üzerindeki başarım oranı % 99.90 olarak gerçekleşmiştir.

B.Huyot ve ark [4] yaptıkları çalışmada copula teorisini veya fonsiyonlarını kullanarak DARPA veri seti üzerinde çevrimiçi (online) denetimsiz (unsupervised) saldırı tespitini gerçekleştirmişlerdir. DARPA veri seti üzerindeki başarım oranını %79 olarak elde etmişlerdir. Her ne kadar literatürde copulalar kullanılarak yapılan pek çok [5, 6, 26–37] çalışma incelenmişsede, bu çalışma dışında copula fonksiyonları kullanılarak literatürde saldırı tespiti ile ilgili bir çalışmaya rastlanılmamıştır.

1.1. Tez Organizasyonu

Bölüm 1'de tez çalışması ile ilgili giriş bilgilerine yer verilmiş olup, tez organizasyonundan bahsedilmiştir. Ayrıca, bu bölüm'de tez çalışmasının konusu ile ilgili önceden yapılan çalışmalardan bahsedilmiştir.

Bölüm 2'de tezde kullanılan materyal ve yöntemden genel olarak bahsedilmiştir. Bu tezin ana konusu olan copula fonksiyonlarından genel olarak bahsedilmiş olup, her bir copula fonksiyonu hakkında gerekli olan bilgiler verilmiştir. Ayrıca, copula tabanlı sınıflandırıcılar inşa edilirken kullanılan naive bayes sınıflandırma algoritmasından da bahsedilmiştir.

Ayrıca bu bölümde; YSA'ların katmanlı yapısı, sınıflandırma biçimi ve kullanım alanları gibi bilgilere yer verilmiştir. DVM, TÖ, KA ve Temel Bileşen Analizi (TBA) gibi yöntemlerden de bahsedilmiştir. STS hakkında genel bilgiler verilmiş olup, STS'lerin türleri olan hizmet engelleme (DoS), admin hesabını ele geçirerek yerel ağda oturum açma (R2L), kullanıcı hesabını yönetici hesabına yükseltme (U2R) ve bilgi tarama (probe) gibi saldırı türlerinden bahsedilmiştir.

Bunlara ek olarak, bu çalışmada kullanılan KDD'99 veri setinden bahsedilmiştir. KDD'99 veri seti üzerinde uygulanan özellik seçim yöntemi olan Minimum Fazlalık Maksimum İlişki (mRMR)'dan bahsedilmiştir. Bölüm 3'te tezde kullanılan makine öğrenme sınıflandırıcıları ile copula tabanlı sınıflandırıcılardan bahsedilmiş olup, KDD'99 veri setleri üzerindeki başarım oranları verilmiştir. Bölüm 4'te elde edilen sonuç kısmına yer verilmiştir. Bölüm 5'te ise kaynaklar kısmına yer verilmiştir.

2. MATERYAL VE YÖNTEM

2.1. Copula Fonksiyonları

2.1.1. Copula

Copula terimi, latincede bağlantı, ilişki anlamında kullanılmaktadır [6, 38–40]. Copula terimi ilk olarak 1959 yılında Abe Sklar tarafından önerilmiştir [4, 6, 36, 40–45]. Copulalar genellikle değişkenler arasındaki bağımlılığı ifade etmek (ölçmek için) kullanılmaktadır [4, 6, 29, 40, 46]. Copulaların temel amacı, birkaç rasgele değişkenin karşılıklı ilişkisini (bağımlılığını) tanımlamaktır [40]. Ayrıca, copulalar rastgele değişkenler arasında bağımlılık yapılarını incelemek ve çok değişkenli dağılım fonksiyonu elde etmek için kullanılmaktadır [47–50].

Çok değişkenli ortak dağılım fonksiyonunu inşa etmek için onların marjinleri ile copula fonksiyonları kullanılmaktadır [5, 45, 51–53]. İstatistik biliminde; copula, rastgele değişken vektörünün ortak dağılım fonksiyonları ile bu dağılımın marjinalleri arasındaki ilişkiyi sağlayan çok değişkenli fonksiyonlardır [6, 26, 28, 38, 54–62]. Diğer bir deyişle; copulalar, marjinal dağılımları kullanarak ortak dağılım elde etmek için kullanılan fonksiyonlardır [4, 27, 38, 52, 63–69].

Copulalar, birçok uygulama alanında kullanılmaktadır. Bu uygulama alanlarının başında sigorta [39], finans [70], istatistik [71], ekonomi [72], risk yönetimi [73] ve güvenlik [74] gibi alanlar gelmektedir [5, 57, 58, 75]. Copulalar; bir sınıf içerisindeki elemanlarının kolayca inşa edilebilmesi, büyük değişkenler içermesi ve iyi cebirsel özelliklere sahip olmasından dolayı uygulamalarda önemli bir yere sahiptir [76].

$u = (u_1, u_2, u_3, \dots, u_n; \theta) \in [0,1]^n$ rastgele değişken vektörü ve C ise ortak dağılım fonksiyonu ifade etmektedir. Denklem (2.1)'de C ortak dağılım fonksiyonu gösterilmiştir.

$$\mathrm{C}(\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \dots, \mathrm{u}_\mathrm{n}; \theta) = \mathrm{P}(\mathrm{U}_1 \leq \mathrm{u}_1, \mathrm{U}_2 \leq \mathrm{u}_2, \mathrm{U}_3 \leq \mathrm{u}_3, \dots, \mathrm{U}_\mathrm{n} \leq \mathrm{u}_\mathrm{n}) \tag{2.1}$$

Burada θ copula paremetresidir. Copulalar türüne göre tek bir paremetre alabildikleri gibi birden fazla parametrede alabilmektedirler.

C iki boyutlu (değişkenli) copula fonksiyonu, marjinalleri $u = [0,1]$ ve $C: [0,1]^2 \rightarrow [0,1]$ şeklinde tanımlanan sürekli bir dağılım fonksiyonu aşağıdaki özelliklere sahiptir.

- $\forall_n \in [0,1]$ için $C(0,u) = C(u,0) = 0$
- $\forall_n \in [0,1]$ için $C(1,u) = C(u,1) = u$
- $u_1 \leq v_1$ ve $u_2 \leq v_2$ olan her $(u_1,u_2),(v_1,v_2) \in [0,1] * [0,1]$ için;
- $C(v_1,v_2) - C(v_1,u_2) - C(u_1,v_2) + C(u_1,u_2) \geq 0$ olur[4, 27, 29, 38, 51, 58, 59, 61, 65, 77–79].

$C(u_1,u_2,\dots,u_n)$, bir copula fonksiyonu olarak verilsin; Burada; Frechet-Hoeffding sınırları denklem (2.2)'deki gibi hesaplanmaktadır [4, 6, 38, 58].

$$\max\left(\sum_{i=1}^{n} u_i + 1 - n, 0\right) \leq C(u_1,u_2,\dots,u_n) \leq \min(u_1,u_2,\dots,u_n) \tag{2.2}$$

C, copula fonksiyonu olsun; C domaininde her (u_1,u_2), (v_1,v_2) için denklem (2.3)'teki eşitsizlik yazılabilir [4, 38].

$$|C(u_2,v_2) - C(u_1,v_1)| \leq |u_2 - u_1| + |v_2 - v_1| \tag{2.3}$$

2.1.2. Sklar Teoremi

Sklar teoremi, copulaların en önemli teoremidir [47, 77, 80]. Sklar teoremi, çok değişkenli ortak dağılım fonksiyonları ile onun (tek değişkenli) marjinal dağılım fonksiyonlarını birbirine bağlamasında copulaların işlevinden bahseder [27, 38, 43, 53, 56, 58, 81]. Bu teorem, copulaları daha anlaşılır kılmış olup, copulalar ile ortak dağılım fonksiyonları arasındaki ilişkiyi kolay bir şekilde ifade etmiştir [57–59].

Marjinalleri $F_1,F_2,\dots,F_d$ olan F, d-boyutlu ortak dağılım fonksiyonu olsun.
$C: [0,1]^d \rightarrow [0,1]$ ve Her $x_1,x_2,\dots,x_d$ için;

$$F(x_1,x_2,\dots,x_d) = C\big(F_1(x_1),F_2(x_2),\dots,F_d(x_d)\big) \tag{2.4}$$

Denklem (2.4)'teki gibi tanımlanmış olan bir C copulası vardır. Marjinalleri, sürekli (devamlı) ise o zaman C copulası tektir. Aksi taktirde; C copulası, $RanF_1 X RanF_2 \dots RanF_d$ (kartezyen çarpımı) üzerinden benzersiz olarak (uniquely) belirlenir. Burada; $RanF_i$, F_i'nin aralığını ifade etmektedir. Bunun tersi de geçerlidir. Eğer C bir copula ve $F_1, F_2, \dots, F_d$ tek değişkenli marjinal dağılım fonksiyonları ise; bu durumda, marjinalleri $F_1, F_2, \dots, F_d$ olan F, d-boyutlu ortak dağılım fonksiyonu denklem (2.4)'teki gibi tanımlanır [4–6, 34, 38, 40, 42, 44, 49, 51–55, 58–62, 64, 65, 67, 68, 75, 77, 78, 80, 82–88].

Ayrıca, her $i \in \{1,2, \dots, d\}$ için $u_i = {F_i}^{-1}(x_i)$ ifadesi denklem (2.4)'te yerine yazılırsa; denklem (2.5) elde edilir [6, 38–40, 42, 44, 57, 77, 78].

$$C(u_1, u_2, \dots, u_d) = F\left({F_1}^{-1}(u_1), {F_2}^{-1}(u_2), \dots, {F_d}^{-1}(u_d)\right) \tag{2.5}$$

Burada; $x_1, x_2, \dots, x_d$ rastgele değişkenlerinin marjinal dağılımları sırasıyla $u_1 = F_1(x_1), u_2 = F_2(x_2), \dots, u_d = F_d(x_d)$ dir [44, 62].

2.1.3. Copula Yoğunluk Fonksiyonu

F, ortak dağılım fonksiyonu olarak verilsin. f, F ortak dağılım fonksiyonunun yoğunluk fonksiyonu olsun. C, copula fonksiyonu olsun. $F_1, F_2, F_3, \dots, F_n$ ise F ortak dağılım fonksiyonun marjinal dağılımlarıdır. Copula yoğunluk fonksiyonu (c) denklem (2.6)'daki gibi hesaplanır [6, 40, 42, 45, 58, 62, 64, 67, 78, 84, 89–92].

$$c(u_1, u_2, \dots, u_n) = \frac{\partial^n C(u_1, u_2, \dots, u_n)}{\partial u_1, \partial u_2, \dots, \partial u_n} = \frac{f({F_1}^{-1}(u_1), {F_2}^{-1}(u_2), \dots, {F_n}^{-1}(u_n))}{\prod_{i=0}^{n} f_i({F_i}^{-1}(u_i))} \tag{2.6}$$

Çok değişkenli yoğunluk fonksiyonu $f(u_1, u_2, \dots, u_n)$ ile copula yoğunluk fonksiyonu arasındaki ilişki denklem (2.7)'de gösterilmiştir [6, 30, 62, 63, 67, 77, 78, 80, 84, 89, 91, 93, 94].

$$f(u_1, u_2, \dots, u_n) = c(F_1(u_1), F_2(u_2), \dots, F_n(u_n); \alpha) \prod_{i=1}^{n} f_i(u_i) \tag{2.7}$$

Burada; f_i,F_i marjinal dağılım fonksiyonunun tek değişkenli yoğunluk fonksiyonudur. c, copula yoğunluk fonksiyonudur. α ise copula parametresidir. Copula yoğunluk fonksiyonu bir copulanın

parametresini tahmin etmek için kullanılmaktadır [30]. Copula fonksiyonları genel olarak elliptical ve arşimedyan copulaları olmak üzere iki temel kategoriye ayrılmaktadır [36, 42, 62, 68, 84].

2.1.4. Elliptical Copula Fonksiyonları

Elliptical copula ailesi; istatistik, ekonomi ve finans alanında yaygın olarak kullanılmaktadır [65, 95]. Elliptical copulalar, çok değişkenli elliptical dağılımlarından oluşmaktadır [6]. Elliptical copulaların similasyonu gerçekleştirmek oldukça kolaydır [47, 61, 80]. Elliptical copulaların en önemli avantajlarından biri, marjinalleri arasında farklı korelasyon seviyeleri belirleyebilmesi ve numerik olarak ifade edilebilmesidir [39]. Dezavantajları ise kapalı forma sahip olmaması ve sınırlı radyal simetriye sahip olmalarıdır [61, 80, 96]. İki değişkenli elliptical copula ailesi için, korelasyon katsayısı (ρ) ve Kendall tau (τ) arasındaki ilişki denklem (2.8)'deki gibi ifade edilir [65, 96].

$$\rho(X,Y) = \sin\left(\frac{\pi}{2}\tau\right) \tag{2.8}$$

Elliptical copula ailesinden kullanılan copulalar; gaussian ve student's-t copulalarıdır [38, 95–97]. Çizelge 2.1'de elliptical copula aileleri, bu copula ailelerinin paramatere aralıkları, Kendall tau katsayıları ve kuyruk bağımlılıkları gösterilmiştir.

Çizelge 2.1. Elliptical copula aileleri, bu copula ailelerinin parametre aralıkları, Kendall tau katsayıları ve kuyruk bağımlılıkları

No	Aile	Parametre Aralığı (ρ, v)	Kendall Tau (τ)	Kuyruk Bağımlılığı (λ_u, λ_l)
1	Gaussian	$\rho \in (-1,1)$	$\frac{2}{\pi} arcsin(\rho)$	0
2	Student's-t	$\rho \in (-1,1), v > 2$	$\frac{2}{\pi} arcsin(\rho)$	$2t_{v+1}\left(-\sqrt{v+1}\sqrt{\frac{1-\rho}{1-\rho}}\right)$

2.1.4.1. Gaussian (Normal) Copula Fonksiyonu

Gaussian copula, çok değişkenli gaussian dağılımından elde edilmektedir [77, 80, 98]. Gaussian copula, bağımlılık yapısında radyal olarak simetriktir [36, 64, 65]. Gaussian copulası üst kuyruk bağımlılığı $(\lambda_u = 0)$ ve alt kuyruk bağımlılığı $(\lambda_l = 0)$ olduğundan dolayı kuyruk

bağımlılığını modelleyemez [36]. Gaussian copulasının genel gösterimi $C_{\rho}^{Gaussian}$ şeklinde ifade edilir.

ρ, nxn türünde bir doğrusal korelasyon matrisi olsun;

$0 \leq u_i \leq 1$ ve $\mathrm{i} = 1, \ldots, \mathrm{n}$ için; n-boyutlu gaussian copulası denklem (2.9)'daki gibi ifade edilmektedir [6, 30, 39, 40, 47, 64, 65, 77, 78, 80, 88, 98, 99].

$$C_{\rho}^{Gaussian}(u_1, \ldots, u_n; \rho) = \Phi_{\rho}\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_n)\right) \tag{2.9}$$

Burada; $\Phi^{-1}(u)$, tek değişkenli normal dağılım fonksiyonunun tersidir. Φ_{ρ}, ρ korelasyon matrisine sahip n-boyutlu normal dağılımın ortak kümülatif dağılım fonksiyonudur. ρ ile Kendall tau (τ) arasındaki ilişki: $\rho_{\tau} = \frac{2}{\pi} arcsin(\rho)$ 'dir. ρ ile spearman rho (ρ_s) arasındaki ilişki: $\rho_s = \frac{6}{\pi} arcsin\left(\frac{\rho}{2}\right)$ dir. Korelasyon matrisi, eğer iyi derecede pozitif veya negatif bağımlılık göstermezse üst kuyruk bağımlılık katsayısı $\lambda_u = 0$, alt kuyruk bağımlılık katsayısı ise $\lambda_l = 0$ olarak bulunur [5, 30, 38, 59, 61, 67, 70, 80, 95, 97, 98]. n-boyutlu gaussian copulasının yoğunluk fonksiyonu denklem (2.10)'daki gibi hesaplanır.

$$C(u_1, \ldots, u_n; \rho) = \frac{1}{|\rho|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} S^t(\rho^{-1} - I)S\right) \tag{2.10}$$

Burada; $S = (N^{-1}(u_1), \ldots, N^{-1}(u_n))^t$. t, transposeyi ve I ise birim matrisini ifade etmektedir [30, 100].

2.1.4.2. Student's-T Copula Fonksiyonu

Student's-t copula, çok değişkenli student's-t dağılımlardan elde edilmektedir [80, 98]. Student's-t copulasının genel gösterimi $C_{v,\rho}^{Student's-t}$ şeklinde ifade edilir. Student's-t copula, bağımlılık yapısında radyal olarak simetriktir [36, 64, 65]. Bu copula, hem alt kuyruk hem de üst kuyruk bağımlılığını modeller [36].

$0 \leq u_i \leq 1$ ve $\mathrm{i} = 1, \ldots, \mathrm{n}$; ρ, nxn türünde bir doğrusal korelasyon matrisi; n-boyutlu stutent's-t copulası denklem (2.11)'deki gibi ifade edilmektedir [6, 30, 39, 40, 47, 64, 65, 67, 77, 78, 98].

$$C_{v,\rho}^{Student's-t}(u_1, \ldots, u_n; v, \rho) = \mathrm{t}_{v,\rho}\left({t_v}^{-1}(u_1), \ldots, {t_v}^{-1}(u_n)\right) \tag{2.11}$$

Burada; $t_v{}^{-1}(u)$, tek değişkenli stutent's-t dağılım fonksiyonun tersidir. $t_{v,\rho}$, v serbestlik derecesine ve ρ korelasyon matrisine sahip olan n-boyutlu student's-t dağılımın ortak kümülatif dağılım fonksiyonudur [77, 80, 98]. ρ ile Kendall tau (ρ_τ) arasındaki ilişki: $\rho_\tau = \frac{2}{\pi} arcsin(\rho)$'dir. Student's-t copulasının üst kuyruk bağımlılığı $\lambda_u = 2t_{v+1}\left(-\sqrt{v+1}\sqrt{\frac{1-\rho}{1-\rho}}\right)$, alt kuyruk bağımlılığı ise $\lambda_l = 2t_{v+1}\left(-\sqrt{v+1}\sqrt{\frac{1-\rho}{1-\rho}}\right)$ olarak ifade edilir [38, 61, 70, 80, 95, 97, 98]. n-boyutlu student's-t copulasının yoğunluk fonksiyonu denklem (2.12)'deki gibi ifade edilmektedir.

$$C(u_1, \dots, u_n; \rho, v) = |\rho|^{\frac{1}{2}} \frac{\Gamma(\frac{v+k}{2})}{\Gamma(\frac{v}{2})} \left[\frac{\Gamma(\frac{v}{2})}{\Gamma(\frac{v+1}{2})}\right]^k \frac{(1+\frac{S^{trs}\rho^{-1}s}{v})^{\frac{-v-k}{2}}}{\prod_{j=1}^{k}(1+\frac{S_j^2}{v})^{\frac{-v-1}{2}}} \tag{2.12}$$

Burada; $S = (t_v{}^{-1}(u_1), \dots, (t_v{}^{-1}(u_n))^{trs}$. trs, transposeyi Γ ise gama fonksiyonunu ifade etmektedir [30, 100].

2.1.5. Arşimedyan Copula Fonksiyonları

Arşimedyan copula ailesi, en önemli copula ailelerindendir. Arşimedyan copulaları; kolay inşa edilebilmeleri, copula ailelerinin çeşitli bağımlılık özelliklerini modelleyebilmesi ve uygulamalarda kullanımının kolay olmasından dolayı sıklıkla tercih edilmektedir [26, 38, 39, 42, 57, 58, 64, 65, 71]. Arşimedyan copulaları kullanışlı kılan avantajlarından biri, kapalı forma sahip olmalarıdır [61, 67]. Arşimedyan copula aileleri, denklem (2.13)'teki formül kullanılarak üretilirler [56, 64].

$$C_\theta(u_1, u_2, \dots, u_n) = \varphi^{-1}\left(\sum_{i=1}^{n} \varphi(u_i)\right) \tag{2.13}$$

Burada; θ, bağımlılık parametresi φ ise üreteç fonksiyonudur.
Tanım kümesi $\varphi: I \to [0, \infty]$ sürekli ve kesin azalan bir fonksiyon ve $\varphi(1) = 0$ olarak verilsin. Bu durumda φ'nin sözde tersi olan $\varphi^{[-1]}$, denklem (2.14)'teki gibi ifade edilir.

$$\varphi^{[-1]}(t) = \begin{cases} \varphi^{(-1)}(t), & 0 \le t \le \varphi(0) \\ 0, & \varphi(0) \le t \le \infty \end{cases} \tag{2.14}$$

Burada; $\varphi^{[-1]}$, $[0,\infty]$ domaininde sürekli ve azalmayan, $[0,\varphi(0)]$ domaininde ise kesin ve azalandır. Ayrıca, I domaininde $\varphi^{[-1]}\big(\varphi(u)\big) = u$ ve

$$\varphi(\varphi^{[-1]}(t)) = \begin{cases} t, & 0 \le t \le \varphi(0), \\ \varphi(0), & \varphi(0) \le t \le \infty, \end{cases}$$
$$= \min(t, \varphi(0)).$$

Eğer $\varphi(0) = \infty$ ise $\varphi^{[-1]} = \varphi^{(-1)}$ olur [29, 38].

$\varphi: [0,1] \to [0,\infty]$ sürekli kesin azalan bir fonksiyon olsun. $\varphi(0) = \infty$ ve $\varphi(1) = 0$. φ^{-1}, φ'nün tersi olarak verilsin. $C: [0,1] \to [0,1]^2$, iki boyutlu arşimedyan copula foksiyonu olsun.
Bu durumda her $u, v \in [0,1]$ için iki boyutlu arşimedyan copulası denklem (2.15)'teki gibi hesaplanmaktadır [4, 6, 26, 27, 29, 38, 39, 57–59, 61, 65, 67, 78, 80, 98, 101].

$$C(u,v) = \varphi^{[-1]}(\varphi(u) + \varphi(v)) \tag{2.15}$$

Burada; C , iki boyutlu arşimedyan copulası ve φ ise arşimedyan copulasının üretecidir.

$x \in [0,1]$ ve $\varphi(x) = -lnx$ üreteç fonksiyonu olsun.
Bu durumda;
$\varphi(0) = \infty$ ve $\varphi^{[-1]}(x) = \varphi^{(-1)}(x) = \exp(-x)$ olur. Her $u, v \in [0,1]$ için;

$$C(u,v) = \varphi^{[-1]}(\varphi(u) + \varphi(v)) = \exp(-[(-lnu) + (-lnv)]) = uv = \prod(u,v) \tag{2.16}$$

Denklem (2.16)'daki gibi ifade elde edilir. Burada; Π, üreteci $\varphi(x) = -lnx$ olan kesin arşimedyan copulası olarak ifade edilir [38, 65].
$x \in [0,1]$ ve $\varphi(x) = 1 - x$ üreteç fonksiyonu olsun. Bu durumda; $x \in [0,1]$ olması durumunda $\varphi^{[-1]}(x) = 1 - x$, x>1 olması durumunda ise $\varphi^{[-1]}(x) = 0$ olur. Örneğin; $\varphi^{[-1]}(x) = max(1 - x, 0)$'dır. Her $u, v \in [0,1]$ için;

$$C(u,v) = \varphi^{[-1]}(\varphi(u) + \varphi(v)) = \max(1-x, 0) = \max(u+v-1, 0) = W(u,v) \qquad (2.17)$$

Denklem (2.17)'deki gibi ifade elde edilir. Burada; W, üreteci $\varphi(x) = 1 - x$ olan arşimedyan copula olarak ifade edilmektedir [38].

$\varphi: [0,1] \to [0,\infty]$ sürekli kesin azalan bir fonksiyon olsun. $\varphi(0) = \infty$ ve $\varphi(1) = 0$. φ^{-1}, φ'nün tersi olarak verilsin. Her $n \geq 2$ için $C: [0,1]^n \to [0,1]$, n-boyutlu arşimedyan copulası denklem (2.18)'deki gibi hesaplanır [67].

$$C(u_1, u_2, \dots, u_n) = \varphi^{-1}\big(\varphi(u_1) + \varphi(u_2) + \cdots + \varphi(u_n)\big) \qquad (2.18)$$

$\varphi^{(-1)}, [0,\infty]$ aralığında tamamen monoton ise, C, n-boyutlu bir arşimedyan copula olarak ifade edilir.

C, φ üretecine sahip bir arşimedyan copula olsun.

1. C, simetriktir. Örneğin I domaininde her u, v için $C(u,v) = C(v,u)$ dir.
2. C, birleşmelidir. Örneğin I domaininde her u, v, w için $C(C(u,v), w) = C(u, C(v,w))$ dir.
3. Eğer $c > 0$ ve c, herhangi bir sabit ise o zaman $c\varphi$, C'nin bir üretecidir [26, 29, 38].

C, birleşmeli bir capula olarak verilsin. $u \in (0,1)$ için; $\delta_C(u,v) < u$ olması durumda; C, bir arşimedyan copuladır. Burada; δ_C, C copulasının köşegen kesiti (diagonal section) olarak ifade edilir [38].

$\varphi: [0,1] \to [0,\infty]$, sürekli kesin azalan konveks bir fonksiyon, C bir arşimedyan copula ve X, Y rastgele değişkenler olarak verilsin.
X ve Y için Kendall tau yığın biçimi (population version) (τ_C) denklem (2.19)'daki gibi ifade edilir [4, 26, 38].

$$\tau_C = 1 + 4\int_0^1 \frac{\varphi(t)}{\varphi'(t)} \qquad (2.19)$$

Bu tez çalışmasında arşimedyan copula ailesinden en çok kullanılan clayton, gumbel, frank ve independent olmak üzere dört adet arşimedyan copulası kullanılmıştır. Çizelge 2.2'de önemli

bazı arşimedyan copulaları, bu copulaların üreteçleri, parametre aralıkları, Kendall tau katsayısı ve kuyruk bağımlılıkları gösterilmiştir.

Çizelge 2.2. Arşimedyan copula aileleri, bu copula ailelerinin parametre aralıkları, Kendall tau katsayıları ve kuyruk bağımlılıkları

No	Aile	Üreteç ($\varphi(t)$)	Paremetre Aralığı (θ)	Kendall Tau (τ)	Kuyruk Bağımlılığı (λ_u, λ_l)
1	Clayton	$\theta^{-1}(t^{-\theta}-1)$	$0<\theta<\infty$	$\frac{\theta}{2+\theta}$	$\left(0, 2^{-1/\theta}\right)$
2	Gumbel	$(-\log t)^{\theta}$	$1\leq\theta<\infty$	$1-\frac{1}{\theta}$	$\left(2-2^{1/\theta}, 0\right)$
3	Frank	$-\ln(\frac{e^{-\theta t}-1}{e^{-\theta}-1})$	$\infty<\theta<\infty$	$1-4\theta^{-1}\left(1-D_1(\theta)\right)$	(0,0)
4	Independent	$-\log t$	$-$	0	(0,0)

2.1.5.1. Clayton Copula Fonksiyonu

Clayton copulası, en çok kullanılan arşimedyan copulalarından biridir. Clayton copulasının genel gösterimi $C_{\theta}^{Clayton}$ şeklinde ifade edilir.

Clayton copulasının paremetresi θ olsun. $0<\theta<\infty$ ve $0\leq u_i\leq 1$ olarak verilmesi durumunda; n-boyutlu clayton copulası denklem (2.20)'deki gibi ifade edilir [6, 39, 42, 56, 59, 61, 64, 65, 67, 78, 80, 98].

$$C_{\theta}^{Clayton}(\mathrm{u}_1,\ldots,u_n)=\left\{\sum_{i=1}^{n}\left({u_i}^{-\theta}\right)-n+1\right\}^{-1/\theta} \tag{2.20}$$

Eğer $\theta=0$ olursa, $\lim_{\theta\to 0} C_{\theta}^{Clayton}$ elde edilir. Kendall tau $\tau=\frac{\theta}{2+\theta}$'dir. Üst kuyruk bağımlılık katsayısı $\lambda_u=0$ ve alt kuyruk bağımlılık katsayısı $\lambda_l=2^{-1/\theta}$ dir. Clayton copulasının üreteci $\varphi(t)=\theta^{-1}(t^{-\theta}-1)$ dir. Eğer $\theta=0$ olursa, $\lim_{\theta\to 0}\varphi_{\theta}(t)=-\log t$ elde edilir [6, 27, 38, 39, 42, 56, 57, 61, 64, 65, 67, 70, 80, 98, 102]. Clayton Copula, alt kuyruk bağımlılığını modellemek için uygundur [6, 36]. Bu copula, negatif bağımlılık göstermez[6]. n-boyutlu clayton yoğunluk fonksiyonu denklem (2.21)'deki gibi hesaplanmaktadır [78, 88].

$$C_{\theta}^{ClaytonYoğunluk}(\mathrm{u}_1,\ldots,u_n)=\prod_{i=1}^{n}\{1+(i-1)\theta\}u_i^{-(\theta+1)}\left(\sum_{i=1}^{n}\left({u_i}^{-\theta}\right)-n+1\right)^{-\left(\frac{1}{\theta}+n\right)} \tag{2.21}$$

2.1.5.2. Gumbel Copula Fonksiyonu

Gumbel copulası, en çok kullanılan arşimedyan copulalarından biridir. Gumbel copulası; sağ kuyruk bağımlılığında oldukça güçlü olmasına rağmen, sol kuyruk bağımlılığında ise oldukça zayıftır [42]. Gumbel copulasının genel gösterimi C_θ^{Gumbel} şeklinde ifade edilir.
Gumbel copulasının paremetresi θ olsun. $1 \leq \theta < \infty$ ve $0 \leq u_i \leq 1$ olarak verilsin.
Bu durumda; n-boyutlu gumbel copulası denklem (2.22)'deki gibi hesaplanmaktadır [6, 39, 42, 56, 59, 61, 64, 65, 67, 78, 80, 98, 99].

$$C_\theta^{Gumbel}(\mathrm{u}_1, \dots, u_n) = \exp\left(-\left(\sum_{i=1}^{n}(-\ln u_i)^\theta\right)^{1/\theta}\right) \tag{2.22}$$

Burada; Kendall tau $\tau = 1 - \frac{1}{\theta}$ 'dir. Üst kuyruk bağımlılık katsayısı $\lambda_u = 2 - 2^{1/\theta}$ ve alt kuyruk bağımlılık katsayısı $\lambda_l = 0$'dır. Gumbel copulasının üreteci, $\varphi(t) = (-\log t)^\theta$ dir [6, 27, 38, 39, 42, 47, 56, 57, 61, 64, 65, 67, 70, 78, 80, 98, 102]. Gumbel copula, üst kuyruk bağımlılığını yüksek hassasiyetle yansıtır [36]. Bu copula, negatif bağımlılık göstermez [6].

2.1.5.3. Frank Copula Fonksiyonu

Frank copulası, en çok kullanılan arşimedyan copulalarından biridir. Frank copula, alt kuyruk ve üst kuyruk bağımlılık yapısında radyal olarak simetriktir [6, 64, 65] Frank copulasının kuyruk bağımlılığı zayıf olduğundan dolayı, bu copulanın sadece zayıf kuyruk bağımlılığı gösteren veriler için kullanılması daha uygundur [42]. Frank copulasının genel gösterimi C_θ^{Frank} şeklinde ifade edilir.
Frank copulasının paremetresi θ olsun. $0 < \theta < \infty$ ve $0 \leq u_i \leq 1$ olarak verilsin.
Bu durumda; n-boyutlu frank copulası denklem (2.23)'teki gibi hesaplanmaktadır [6, 39, 42, 56, 59, 61, 64, 65, 67, 78, 80, 98].

$$C_\theta^{Frank}(\mathrm{u}_1, \dots, u_n) = -\frac{1}{\theta}\ln\left(1 + \frac{\prod_{i=1}^{n}\left(e^{-\theta u_i} - 1\right)}{(e^{-\theta} - 1)^{n-1}}\right) \tag{2.23}$$

Eğer $\theta = 0$ olursa, $\lim_{\theta \to 0} C_{\theta}^{Frank}$ elde edilir. Kendall tau $\tau = 1 - 4\theta^{-1}\big(1 - D_1(\theta)\big)$'dir. $D_1(\theta)$, debye fonksiyonu olarak ifade edilir. $D_1(\theta) = \theta^{-1} \int_0^{\infty} \frac{t}{exp(t)-1} dt$. Üst kuyruk bağımlılık katsayısı $\lambda_u = 0$ ve alt kuyruk bağımlılık katsayısı $\lambda_l = 0$'dır. Frank copulasının üreteci, $\varphi(t) = -\ln(\frac{e^{-\theta t}-1}{e^{-\theta}-1})$ dir. Eğer $\theta = 0$ ise $\lim_{\theta \to 0} \varphi_{\theta}(t) = -\log t$ elde edilir [6, 27, 38, 39, 42, 56, 61, 64, 65, 67, 70, 80, 98, 102].

2.1.5.4. Independent (Independence) Copula Fonksiyonu

Independent copulası, önemli arşimedyan copulalarından biridir. Independent copulasının genel gösterimi $\Pi^{Independent}$ olarak ifade edilir.

$0 \leq u_i \leq 1$ olarak verilisin.
Bu durumda; n-boyutlu Independent copulası denklem (2.24)'teki gibi hesaplanmaktadır [6, 38, 98].

$$\Pi^{Independent}(\mathrm{u}_1, \dots, u_n) = \prod_{i=1}^{n} u_i \tag{2.24}$$

Burada; Kendall tau $\tau = 0$'dır. Üst kuyruk bağımlılık katsayısı $\lambda_u = 0$ ve alt kuyruk bağımlılık katsayısı $\lambda_l = 0$ olur. Independent copulasının üreteci, $\varphi(t) = -\log t$ dir [38, 98].

2.1.5.5. Arşimedyan Copulalarının Yoğunluk Fonksiyonu

n-boyutlu copula yoğunluk fonksiyonu $c(u_1, u_2, \dots, u_n) = \frac{\partial^n C(u_1, u_2, \dots, u_n)}{\partial u_1, \partial u_2, \dots, \partial u_n}$ yeniden yazılarak n-boyutlu arşimedyan copula yoğunluğu denklem (2.25)'teki gibi hesaplanır [6, 103].

$$\begin{aligned} c(u_1, u_2, \dots, u_n) &= \frac{\varphi(\varphi^{-1}(u_1) + \varphi^{-1}(u_1) + \cdots + \varphi^{-1}(u_n))}{\partial u_1, \partial u_2, \dots, \partial u_n} \\ &= \varphi^n(\varphi^{-1}(u_1) + \varphi^{-1}(u_1) + \cdots + \varphi^{-1}(u_n)) \prod_{i=1}^{n} (\varphi^{-1})'(u_i) \end{aligned} \tag{2.25}$$

Burada; herhangi bir arşimedyan copulanının yoğunluğunu (c) hesaplamak için o copulanın φ üreteç fonksiyonu kullanılarak elde edilir.

2.1.6. Paremetre Tahmin Etme Metotları

EML/ML, IFM ve CML metotları; copula ailelerinin ve marjinallerinin parametrelerini tahmin etmek için kullanılmaktadır[42, 104].

2.1.6.1. Tam Maksimum Olasılık/Maksimum Olasılık (EML/ML) Metodu

ELM/ML metodu, copula parametresi ve marjinallerin parametreleri eşzamanlı olarak tahmin etmektedir [6, 42, 44, 58, 61, 87, 99, 105]. Uygulamalarda yaygın olarak kullanılmamaktadır[6]. EML metodunun dezavantajı daha yüksek boyutlu dağılımlar için hesaplama işlemi fazla uzun sürmesi veya hesaplamada güçlük çekilmesidir [6, 47, 60, 89, 90]. Copula parametleri tahmini yapılırken marjinaller için dağılım varsayımsal olarak tahmin edilmektedir [47, 77, 90].

T zaman periyodunda N farklı değişkenin değerini içeren $x = (x_1^t, x_2^t, \dots, x_N^t)_{t=1}^T$ örneklere sahip olduğumuzu varsayalım [6, 67, 77, 80, 89]. Θ, parametre uzayı ve θ ise tahmin edilecek olan (kx1) boyutlu parametre vektörü olsun. $L_t(\theta)$ ve $l_t(\theta)$ sırasıyla t zamandaki gözlemler (observations) için olasılık (likelihood) ve log-olasılık (log-likelihood)'dır [47, 67].

F, ortak dağılım fonksiyonu olarak verilsin. f, F ortak dağılım yoğunluk fonksiyonu olsun. Çok değişkenli yoğunluk foksiyonu $f(x_1, x_2, \dots, x_n)$ ile copula yoğunluk foksiyonu (c) arasındaki ilişki denklem (2.26)'da gösterilmiştir [6, 45, 62, 63, 67, 77, 78, 80, 84, 89, 91, 93, 94].

$$c(F_1(x_1), F_2(x_2), \dots, F_n(x_n); \alpha) = \frac{f(x_1, x_2, \dots, x_n)}{\prod_{i=1}^n f_i(x_i)} \tag{2.26}$$

Burada; f_i, F_i marjinal dağılımın fonksiyonunun tek değişkenli yoğunluk fonksiyonudur. c, copula yoğunluk fonksiyonudur. α ise copula parametresidir. copula yoğunluk fonksiyonu denklem (2.27)'deki gibi hesaplanır [6, 42, 45, 58, 62, 64, 67, 78, 84, 89–91].

$$c(u_1, u_2, \dots, u_n) = \frac{\partial C(u_1, u_2, \dots, u_n)}{\partial u_1, \partial u_2, \dots, \partial u_n} \tag{2.27}$$

$\theta = (\theta_1, \theta_2, \dots, \theta_N, \alpha)$ tahmin edilecek olan parametre vektörü olsun. i=1,2,3…N. θ_i, F_i marjinal dağılımın parametre vektörü α ise copula parametresidir [67, 80].
log-olasılık fonksiyonu $l(\theta)$ denklem (2.28)'deki gibi hesaplanmaktadır[6, 47, 58, 60, 61, 67, 77, 78, 80, 89, 91].

$$l(\theta) = \sum_{t=1}^{T} \ln c(F_1(x_1^t; \theta_1), F_2(x_2^t; \theta_2), \dots, F_N(x_N^t; \theta_N); \alpha) + \sum_{t=1}^{T} \sum_{n=1}^{N} ln\, f_n\,(x_n^t; \theta_n) \qquad (2.28)$$

Burada; θ parametre vetörünün $\hat{\theta}_{EML/ML}$ parametre tahmin edicisi, denklem (2.29)'daki gibi hesaplanmaktadır [6, 47, 58, 61, 67, 68, 78, 80, 91].

$$\hat{\theta}_{EML/ML} = arg \max_{\theta \in \Theta} l(\theta) \qquad (2.29)$$

2.1.6.2. Marjinleri için Çıkarım İşlevleri (IFM) Metodu

EML metodunun çok zaman alması veya uygulanmasının mümkün olmadığı durumlarda IFM metodu kullanılmaktadır [58, 60, 77, 89, 94]. IFM metodu, copula parametresi ve marjinallerin parametresini ayrı ayrı tahmin eder [42, 44, 77, 80]. Birinci adımda marjinal dağılım parametre tahmini, ikinci adımda ise copula parametre tahmini yapılır [42, 47, 60, 61, 77, 80, 89, 90, 105]

log-olasılık fonksiyonu $l(\theta)$ denklem (2.30)'daki gibi hesaplanmaktadır [6, 58, 67, 77, 80, 90, 94].

$$l(\theta) = \sum_{t=1}^{T} \ln c(F_1(x_1^t; \theta_1), F_2(x_2^t; \theta_2), \dots, F_N(x_N^t; \theta_N); \alpha) + \sum_{t=1}^{T} \sum_{n=1}^{N} ln\, f_n\,(x_n^t; \theta_n) \qquad (2.30)$$

Burada; $\theta = (\theta_1, \theta_2, \dots, \theta_N)$ marjinallerin parametre vektörü ve α ise copula fonksiyonunun parametresini ifade eder. (θ, α) ise tahmin edilecek olan parametre vektörüdür. (θ, α) vektörü, iki adımda tahmin edilir.

1.Adım

IFM metodu aracılığıyla, tek değişkenli marjinallerin parametre tahmini denklem (2.31)'deki ifade ile bulunmaktadır [6, 42, 47, 58, 67, 77, 78, 80, 89].

i=1,2,3,…N için;

$$\hat{\theta}_i = \arg\max_{\theta_i} \sum_{t=1}^{T} \ln f_i(x_i^t; \theta_i) \tag{2.31}$$

2.Adım

Marjinaller için elde edilen tahminler $(\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N)$ kullanılarak, denklem (2.32)'deki α copula parametresi tahmin edilir [6, 42, 58, 67, 68, 77, 78, 80, 89].

$$\hat{\alpha}_{IFM} = arg\max_{\alpha} \sum_{t=1}^{T} \ln c(F_1(x_1^t; \hat{\theta}_1), F_2(x_2^t; \hat{\theta}_1), \dots, F_n(x_n^t; \hat{\theta}_N); \alpha) \tag{2.32}$$

$\hat{\theta}_{IFM}$ parametre tahmin edicisi, denklem (2.33)'teki gibi hesaplanır [6, 58, 61, 67, 78].

$$\hat{\theta}_{IFM} = (\hat{\theta}_1, \hat{\theta}_2, \dots \hat{\theta}_N, \hat{\alpha}_{IFM}) \tag{2.33}$$

2.1.6.3. Standart Maksimum Olasılık (CML) Metodu

EML/ML ve IFM'de tek değişkenli marjinallerin parametrelerinin tahmini gerekir. CML metodunda ise marjinallerin parametrik olmayan tahmini deneysel (empirical) marjinal dağılım fonksiyonu kullanılarak yapılmaktadır. CML metodu, copula parametresi ve marjinallerin parametresini ayrı ayrı tahmin eder [42, 99]. Birinci adımda deneysel marjinal dağılım fonksiyonu hesaplanır, ikinci adımda ise copula parametre tahmini yapılır [42, 61, 67, 68, 77, 99]. Hem EML hem de IFM tek değişkenli marjinlerinin parametrik formlarının dışsal dayatmasına (exogenous imposition) dayanmaktadır [42, 44, 67, 77]. CML metodu ise deneysel marjinal dönüşüm kavramına dayanmaktadır [42, 44, 47, 67, 77, 89].

T gözlemlerinin sayısı $x = (x_1^t, x_2^t, \dots, x_N^t)_{t=1}^T$ olsun. Deneysel dağılım fonksiyonu $\hat{F}_n(.)$; denklem (2.34)'teki gibi hesaplanır [6, 61, 67, 77, 78].

$$\hat{F}_n(.) = \frac{1}{T} \sum_{t=1}^{T} I\,(x_n^t \leq .), \qquad n = 1,2,3, \dots, N \tag{2.34}$$

Burada; $I(x_n^t \leq .)$, işaret (indicator) fonksiyonunu ifade etmektedir. CML metodu iki adımda hesaplanır.

1.Adım

Tek değişkenli deneysel marjinal dağılım fonksiyonu kullanılarak; denklem (2.35)'teki gibi $x = (x_1^t, x_2^t, \dots, x_N^t)_{t=1}^T$ veri kümesi, tekdüze değişkenlere (uniform variates) dönüşümü yapılır [6, 47, 67, 68, 77, 78, 80, 89, 99, 106].

$t = 1,2, \dots, T$ için; denklem (2.35)'deki sözde gözlemler (pseudo-observations) elde edilir.

$$\hat{u}_t = (\hat{u}_1^t, \hat{u}_2^t, \dots, \hat{u}_N^t) = \left[\hat{F}_1(x_1^t), \hat{F}_2(x_2^t), \dots, \hat{F}_N(x_N^t)\right] \tag{2.35}$$

2.Adım

Sözde gözlemler kullanılarak; α, copula parametresi denklem (2.36)'daki gibi tahmin edilir [6, 61, 67, 68, 77, 78, 106].

$$\hat{\alpha}_{CML} = arg \max_{\alpha} \sum_{t=1}^{T} \ln c(\hat{u}_1^t, \hat{u}_2^t, \dots, \hat{u}_N^t; \alpha) \tag{2.36}$$

$\hat{\theta}_{CML}$ parametre tahmin edicisi, denklem (2.37)'deki gibi hesaplanır [6, 61, 67].

$$\hat{\theta}_{CML} = \hat{\alpha}_{CML} \tag{2.37}$$

IFM ile CML arasındaki fark; IFM ilk önce dağıtımsal varsayımlar altında marjinal dağılımları tahmin eder ve daha sonra veri setini tekdüze değişkenlere dönüştürür. CML ise doğrudan veri seti üzerinde deneysel marjinal dağılımları kullanılarak veri setini tekdüze değişkenlere dönüştürür [47, 68, 77, 89].

2.1.7. Naive Bayes Sınıflandırıcısı

Naive bayes teoremi, veri belirsizliğini hesaplamak (üstesinden gelmek) için kullanılan bir yöntemdir [107–110]. Naive bayes, makine öğrenmesi ve veri madenciliğinde sıkça kullanılan bir sınıflandırma tekniğidir [34, 111–114]. Naive bayes sınıflandırıcısı, bayes kuralına ve olasılık

teoremine dayanmaktadır [115]. Bu sınıflandırıcı, veri setinin özelliklerinden sınıfı tahmin etmek için kullanılan denetimli öğrenme yöntemidir [112, 116, 117].

Naive bayes sınıflandırıcısı, güçlü bir performansa sahiptir. Bu güçlü performansının arkasındaki nedenlerden biri, tahmin ediciler (predictors) arasında koşullu bağımsızlık varsayımıdır [112, 114, 118]. Bu bağımsızlık varsayımı; bazen özelliklerin birbirleriyle ilişkili olduğu durumlarda doğruluk kaybına yol açmaktadır [34]. Bu sınıflandırıcı; basitliği, daha az hesaplama karmaşıklığı, daha az bellek gereksinimi ve iyi tahmin doğruluğu nedeniyle diğer sınıflandırıcılara kıyasla daha iyi performans göstermektedir [34, 109, 110, 114, 119–123].

$X = (X_1, X_2, , X_3, \dots, X_K)$, K boyutlu değişkenler vektörü olsun. Y, toplam sınıf sayısı olarak verilsin. $(S_1, S_2, , S_3, \dots, S_D)$ ise Y'nin D sınıf etiketleri (D class labels of Y) olsun. Naive bayes sınınflandırıcısı denklem (2.38)'deki gibi ifade edilebilir.

$$P(S_d|X) = \frac{P(S_d)P(X|S_d)}{P(X)} \tag{2.38}$$

Burada;

$P(S_d|X)$, sonraki olasılığı (posterior probability) veya bir S_d sınıfına ait olan X özellik değerinin olasılığı

$P(S_d)$, önceki sınıfın olasılığı (prior probability)

$P(X)$, önceki tahmin edicinin olasılığı veya özellik değerinin olasılığı (tüm sınıflar için bu sabit bir değerdir)

$P(X|S_d)$, S_d sınıfı olarak verilen X'nin olasılık fonksiyonu veya verilen bir S_d sınıfının X özellik değerinin olasılığını ifade etmektedir [5, 6, 34, 109, 110, 112–115, 117–119, 121, 124–131]. Naive bayes sınıflandırıcısı, her örneğe en yüksek koşullu olasılıklı sınıf değerini atar. Değişkenlerin sınıf değişkenine koşullu olarak bağımsız olduğu varsayımını kullanarak denklem (2.39)'daki ifade elde edilir.

$$P(S_d|X) = \frac{P(S_d)\prod_{k=1}^{K} P(X_k|S_d)}{P(X)} \tag{2.39}$$

Denklem (2.39) ifadesi, girdi verisi verildiğinde en iyi sınıfı tahmin etmek için yeterlidir. Çünkü $P(X)$ sabittir. Tahmin skorunun gerekli olduğu problemlerde; sınıf şartlı olasılığı (the class conditional probability) denklem (2.40) kullanılarak tahmin edilir.

$$P(S_d|X) = \frac{P(S_d)\prod_{k=1}^{K} P(X_k|S_d)}{\sum_{i=1}^{S} P(S_i)\prod_{k=1}^{K} P(X_k|S_i)} \tag{2.40}$$

Naive bayes sınıflandırıcısı, gerçek sınıf koşullu olasılıkları tahmin etmede zayıftır. Çünkü gerçek veri uygulamalarında bağımsızlık varsayımı genellikle ihlal edilmektedir [112, 114, 118, 124–127, 132, 133].

2.1.8. Copula Tabanlı Sınıflandırıcılar İnşa Etme

Copula tabanlı sınıflandırıcıları inşa etmede naive bayes teoreminden faydalanacağız. Naive bayes teoremi, denklem (2.38)'de verilmiştir. Denklem (2.38)'de verilen ifade sınıflandırma aracı olarak kullanılmaktadır.

Maksimum sonsal olasılık (MAP), naive bayes teoreminde kullanılan bir tahmin etme yöntemidir [134, 135]. MAP sınıflandırıcısı, $P(S_d|X)$ sonraki olasılığı karşılaştırılarak tasarlanabilir. Diğer bir deyişle; MAP, sonraki olasılığı maksimum eden sınıfı arayarak inşa edilebilir [136]. Ayrıca MAP, en iyi olasılığı seçmek için kullanılmaktadır [114]. Bir nesnenin $X = (X_1, X_2, , X_3, \dots, X_K)$ özellikleri D sınınfına aittir. Bu nesne sınıflandırılırken D sınıfı içerisinde en yüksek sonraki olasılığa sahip olan sınıfa atanır [5].

Sürekli özellikler için; bir gaussian copula fonksiyonu, olasılık fonksiyonundaki bağımlılık yapısının modellenmesi için kullanılabilir. Bu durumda; denklem (2.38)'de verilen ifade de gaussian copula yoğunluğu kullanılarak olasılık denklem (2.41)'deki gibi hesaplanabilir.

$$P(S_d|X) = \frac{\Phi(F_1(X_1), F_2(X_2), \dots, F_K(X_K)|S_d)P(S_d)\prod_{k=1}^{K} f_k(X_k|S_d)}{f(X)} \tag{2.41}$$

Burada; F_i, marjinal dağılım fonksiyonlarını ve f_i ise özelliklerin marjinal yoğunluğunu ifade etmektedir. Φ, gaussian copula yoğunluğunu ifade etmektedir [5, 6, 34, 118]. Denklem (2.41)'de gaussian copula yoğunluğu yerine herhangi bir (student's-t, clayton, frank, gumbel, independent) copula yoğunluğu yazılarak denklem (2.42)'deki gibi genelleştirilmesi yapılabilir [5, 6].

$$P(S_d|X) = \frac{\mathrm{c}(F_1(X_1), F_2(X_2), \dots, F_K(X_K)|S_d)P(S_d)\prod_{k=1}^{K} f_k(X_k|S_d)}{f(X)} \tag{2.42}$$

Burada; c, herhangi bir copula yoğunluğunu ifade etmektedir. Denklem (2.42)'de verilen ifade, naive bayes sınıflandırıcısı olarak ifade edilmektedir. Naive bayes sınınflandırıcısı, independent copula kullanılarak elde edilebilir. Bu yüzden, copulaya dayalı sınıflandırıcı onun genelleştirilmiş halidir. Denklem (2.42)'de verilen ifadeden yola çıkılarak, denklem (2.43)'teki gibi copula tabanlı bir MAP sınıflandırıcısı inşa edilebilir [5, 6, 34, 118, 135].

$$\widehat{S_D} = arg \max_{S_d \in Y} \mathrm{c}(F_1(X_1), F_2(X_2), \dots, F_K(X_K)|S_d) \prod_{k=1}^{K} f_k(x_k|S_d) P(S_d) \tag{2.43}$$

MAP sınıflandırıcı inşa edilirken, IFM metodunun uygulanması durumunda; herhangi bir (gaussian, student's-t, clayton, frank, gumbel, independent vb.) copula, IFM metodunun uygulanması sırasında tahmin edilen marjinaller, onların yoğunlukları ve önceki deneysel olasılıkları denklem (2.43)'te yerine yazılarak MAP sınıflandırıcı elde edilir. CML metodunun uygulanması durumunda ise; herhangi bir copula, deneysel kümülatif dağılım fonksiyonları, deneysel yoğunlukları ve önceki deneysel olasılıkları denklem (2.43)'te yerine yazılarak elde edilir [6].

2.2. Bağımlılık Ölçümleri

Bağımlılık, bir rastgele değişkenin büyük veya küçük değerlerinin diğer değişkenlerinin büyük veya küçük değerleriyle nasıl bir ilişki olduğunu açıklar. Pearson doğrusal korelasyonu, sıralama korelasyonu ve kuyruk bağımlığı olmak üzere üç tane bağımlılık ölçümü bulunmaktadır. Pearson doğrusal korelasyonu, elliptical dağılımlarda uygun bir bağımlılık ölçümdür. Elliptical dağılımlar dışında pearson doğrusal korelasyonun bir takım yanlışlıklara yol açtığı iyi bilinmektedir. Sıralama korelasyonu ve kuyruk bağımlığı ise herhangi bir bağımlılık yapısı için kullanılmaktadır. Kuyruk bağımlılığı, aşırı uçlarda (in the extremes) bağımlılığı dikkate alır [40].

2.2.1. Pearson Doğrusal Korelasyonu

Pearson doğrusal korelasyonu, en yaygın kullanılan bağımlılık ölçümü türüdür [46]. Pearson doğrusal korelasyonu, bir değişkenin diğer değişkenle doğrusal olarak ilişkili olduğu yönü ve dereceyi ölçmektedir [137].

X ve Y rastgele değişkenler için, doğrusal korelasyon katsayısı denklem (2.44)'teki gibi hesaplanmaktadır [46, 137, 138].

$$\rho_{(X,Y)} = r_{(X,Y)} = \frac{cov[X,Y]}{\sigma_X \sigma_Y} = \frac{cov[X,Y]}{\sqrt{Var(X)}\sqrt{Var(Y)}} \tag{2.44}$$

Burada; $cov[X,Y]$, X ve Y arasındaki kovaryans (covariance)'tır. σ_X, σ_Y, rasgele X ve Y değişkenlerinin standart sapmalarıdır. $Var(X)$ ve $Var(Y)$, sırasıyla X ve Y'nin varyanslarıdır. Pearson doğrusal korelasyonu ($\rho_{(X,Y)}$), $(-1,1)$ aralığında değerler almaktadır. $\rho_{(X,Y)} = 1$ olduğunda, X ve Y değişkenlerinin artan bir ilişkiye mükemmel bir şekilde bağımlı olduğu söylenir. $\rho_{(X,Y)} = -1$ olduğunda ise X ve Y değişkenleri azalan bir ilişkiye mükemmel bir şekilde bağımlıdır. Ayrıca, rastgele X ve Y değişkenler bağımsızsa, bu iki değişken arasındaki korelasyon sıfıra eşittir [137]. Bunlara ek olarak; pearson doğrusal korelasyonu, her zaman bağımlılık ölçümlerinin istenen özelliklerini karşılamamaktadır [137].

2.2.2. Sıralama Korelasyonu

(x_1, y_1) ve (x_2, y_2), $(X,Y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ sürekli rastgele değişkenler vektöründe iki gözlem olarak verilsin. Burada; n gözlemlerin sayısının ifade etmektedir. Eğer $x_2 > x_1$ ve $y_2 > y_1$ veya $x_2 < x_1$ ve $y_2 < y_1$ ise (x_1, y_1) ve (x_2, y_2) uyumlu (concordant) olduğu söylenilebilir. Benzer şekilde; eğer $x_2 > x_1$ ve $y_2 < y_1$ veya $x_2 < x_1$ ve $y_2 > y_1$ ise (x_1, y_1) ve (x_2, y_2) uyumsuz (discordant) olduğu söylenilebilir. Ayrıca, $(x_1 - x_2)(y_1 - y_2) > 0$ olması durumunda (x_1, y_1) ve (x_2, y_2) uyumlu ve $(x_1 - x_2)(y_1 - y_2) < 0$ olması durumunda ise (x_1, y_1) ve (x_2, y_2) uyumsuz olduğu söylenilebilir [38, 137, 138].

2.2.2.1. Kendall Tau Sıralama Korelasyonu

Kendall tau sıralama korelasyonu, rastgele X ve Y değişkenleri arasındaki uyumlu olasılığı ile uyumsuz olasılığı arasındaki farkı ölçen parametrik olmayan bir korelasyon ölçüsüdür [38, 46]. $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, (X,Y) sürekli rastgele değişkenler vektöründen n gözlemlerin bir rastgele örneği olarak verilsin. Belli bir örnekte (x_i, y_i) ve (x_j, y_j) gözlemlerinin $\binom{n}{2}$ farklı çiftleri vardır. Her bir çift ya uyumludur ya da uyumsuzdur. Belli bir örnek için Kendall tau denklem (2.45)'teki gibi hesaplanmaktadır[137].

$$\tau_{\text{Kendall } tau} = \frac{(C-D)}{n(n-1)/2} = (C-D)/\binom{n}{2} \tag{2.45}$$

Burada; C, uyumlu çiftlerin sayısını ve D ise uyumsuz çiftlerin sayısını ifade eder. Kendall tau sıralama korelasyonu; spearman rho gibi, temeldeki değişkenlerin monotonik doğrusal olmayan dönüşümleri altında değişmezdir.

(x_1, y_1) ve (x_2, y_2), ortak dağılım fonksiyonları H ve copulaları C olan sürekli rastgele bağımsız vektörler olarak verilsin. Bu durumda; Kendall tau yığın biçimi denklem (2.46)'daki gibi hesaplanmaktadır [38, 138–141].

$$\begin{aligned} \tau_{(X,Y)} = \tau_C &= P((x_1 - x_2)(y_1 - y_2) > 0) - P((x_1 - x_2)(y_1 - y_2) < 0) \\ &= 4\int_0^1 \int_0^1 C(u,v)dC(u,v) - 1 \end{aligned} \tag{2.46}$$

Burada; $\tau_{(X,Y)}$, (x_1, y_1) ve (x_2, y_2)'nin uyumlu ve uyumsuz olasılıkları arasındaki farkını ifade etmektedir [38, 46].

2.2.2.2. Spearman Rho Sıralama Korelasyonu

Spearman rho doğrusal korelasyonu, parametrik olmayan bir korelasyon ölçümüdür. Spearman rho tau denklem (2.47)'deki gibi hesaplanmaktadır[137].

$$\rho_S = 1 - \frac{6\sum d^2}{n(n^2-1)} \tag{2.47}$$

Burada; n, eşleştirilmiş sıraların (the paired ranks) sayısıdır. d ise eşleştirilmiş sıralar (the paired ranks) arasındaki farkı ifade etmektedir. Değişkenler en yüksek sırayı (the highest rank), en yüksek değere (the highest value) atanarak sıralanır. Spearman sıra korelasyonunun önemli avantajı, iki değişken arasındaki doğrusal olmayan bağımlılığı modelleyebilmesidir [40, 137].
(x_1, y_1) ve (x_2, y_2), ortak dağılım fonksiyonları H ve copulaları C olan sürekli rastgele bağımsız vektörler olarak verilsin. Bu durumda; Spearman rho yığın biçimi denklem (2.48)'deki gibi hesaplanmaktadır [38, 138–141].

$$\rho_{(X,Y)} = \rho_C == 3(P((x_1 - x_2)(y_1 - y_2) > 0) - P((x_1 - x_2)(y_1 - y_2) < 0)) \\ = 12 \int_0^1 \int_0^1 C(u,v) dudv - 3 \tag{2.48}$$

Burada; $\rho_{(X,Y)}$, (x_1, y_1) ve (x_2, y_2)'nin uyumlu ve uyumsuz olasılıkları arasındaki farkını ifade etmektedir [38, 46]. Kendall tau sıralama korelasyon katsayısı ile doğrusal korelasyon katsayısı arasındaki ilişki: $\rho_\tau(X,Y) = \frac{2}{\pi} arcsin(\rho)$ dir. Spearman rho sıralama korelasyon katsayısı ile doğrusal korelasyon katsayısı arsındaki ilişki: $\rho_S(X,Y) = \frac{6}{\pi} arcsin\left(\frac{\rho}{2}\right)$ dir. Burada; ρ, rastgele iki değişken arasındaki doğrusal korelasyon katsayısıdır [40].

2.2.3. Kuyruk Bağımlılığı

Kuyruk bağımlılığı, rastgele X ve Y değişkenlerinin aşırı değerleri arasındaki uyuma bakan bir bağımlılık ölçüsüdür [38, 137]. Diğer bir deyişle; kuyruk bağımlılığı, ortak dağılım fonksiyonunun sağ üst ve sol alt çeyreğinde X ve Y değişkenleri arasındaki bağımlılığı ölçer [38, 39, 141]. Kuyruk bağımlılığı ölçümlerinin en cazip özelliği, X ve Y değişkenlerin marjinal dağılımlarından bağımsız olmaları ve bu değişkenlerin monoton dönüşümleri altında değişmez olmalarıdır [38, 39, 137, 141].

X ve Y sürekli rastgele değişkenler olarak verilsin. F ve G sırasıyla X ve Y'nin marjinal dağılım fonksiyonları olsun. $\lambda_U, \lambda_L \in [0,1]$ olarak verilsin. Bu durumda; üst kuyruk bağımlılığı (λ_U) denklem (2.49)'daki gibi ifade edilmektedir [39, 40, 58, 59, 61, 67, 77].

$$\lambda_U = \lim_{t \to 1^-} P\left[Y > G^{(-1)}(t) \middle| X > F^{(-1)}(t)\right] \tag{2.49}$$

Alt kuyruk bağımlılığı (λ_L) ise denklem (2.50)'deki gibi ifade edilmektedir.[39, 40, 58, 59, 61, 67, 77].

$$\lambda_L = \lim_{t \to 0^+} P\left[Y \leq G^{(-1)}(t) \middle| X \leq F^{(-1)}(t)\right] \tag{2.50}$$

Burada; eğer $\lambda_U = 0$ ve $\lambda_L = 0$ ise, X ve Y sürekli değişkenler üst kuyruk ve alt kuyrukta asimptotik olarak bağımsız olduğu ifade edilir. Eğer $\lambda_U = \lambda_L$ ise X ve Y değişkenlerin simetrik

kuyruk bağımlılığına sahip olduğu ifade edilir. Eğer $\lambda_U \neq \lambda_L$ ise X ve Y değişkenlerin simetrik olmadığı ifade edilir [38, 58, 59, 67, 77, 137].

$X, Y, F, G, \lambda_U, \lambda_L$ (yukarıda verilmiş) , X ve Y'nin copulası C ve köşegen kesiti (diagonal section) δ_C olarak verilsin. Denklem (2.49) ve denklem (2.50)'de ifadeler sırasıyla denklem (2.51) ve denklem (2.52) deki gibi genelleştirilebilir [4, 38, 59, 77, 87, 137].

$$\lambda_U = \lim_{t\to 1^-} P\left[Y > G^{(-1)}(t) \middle| X > F^{(-1)}(t)\right] = 2 - \lim_{t\to 1^-} \frac{1 - C(t,t)}{1-t} = 2 - \delta'_C(1^-) \qquad (2.51)$$

$$\lambda_L = \lim_{t\to 0^+} P\left[Y \leq G^{(-1)}(t) \middle| X \leq F^{(-1)}(t)\right] = \lim_{t\to 0^+} \frac{C(t,t)}{t} = \delta'_C(0^+) \qquad (2.52)$$

Copulalarda bağımlılığı modellemek için sıklıkla kuyruk bağımlılıkları kullanılmaktadır. Bazı önemli copula aileleri için kuyruk bağımlılıkları Çizelge 2.3'te gösterilmiştir.

Çizelge 2.3. Bazı copula aileleri için kuyruk bağımlılıkları

Copula Ailesi	**Üst Kuyruk Bağımlılığı**	**Alt Kuyruk Bağımlılığı**
Gaussian (Normal)	0	0
Student's-t	$2t_{v+1}\left(-\sqrt{v+1}\sqrt{\frac{1-\rho}{1-\rho}}\right)$	$2t_{v+1}\left(-\sqrt{v+1}\sqrt{\frac{1-\rho}{1-\rho}}\right)$
Clayton	0	$2^{-1/\theta}$
Gumbel	$2-2^{1/\theta}$	0
Frank	0	0
Independent (Independence)	0	0

2.3. Yapay Sinir Ağları (YSA)

YSA, insan beyninin işleyişinden esinlenerek yeni bilgiler elde etme, yeni bilgileri oluşturabilme, yeni bilgileri kavrayabilme gibi yetenekleri kullanarak herhangi bir yardım almadan otomatik olarak gerçekleştirebilen bilgisayar sistemleri olarak ifade edilebilir [13, 25, 142–150]. Diğer bir deyişle; YSA, insan beyninin yaptığı işlemlerin bilgisayar ortamında matematiksel olarak modellenmesi işlemidir [13, 147, 151–155]. Bundan dolayı, YSA'lar üzerindeki çalışmalar ilk olarak insan beynini oluşturan nöronların modellenmesi ile başlamış olup, daha sonra bilişim sistemlerinin gelişimiyle beraber birçok alanda kullanılmıştır [144, 146, 156, 157].

İnsan beyninin çalışma prensibini taklit eden YSA sistemleri, herhangi bir canlı beyninin fonksiyonları göz önüne alındığında çok yetersiz kalmaktadır [150]. İnsan beyninde yaklaşık 100 milyar sinir hücresi bulunmaktadır. Bu sayının şuan bilgisayar ortamında modellenmesi imkânsız görünmektedir [149, 158, 159]. YSA'lar, karar verme hızı bakımından insan beyni ile henüz yarışacak durumda değildir. Ancak, YSA'lar öğrenme ve bu öğrenme yoluyla öğrendiği bilgileri saklaması açısından uygulama alanları giderek artmaktadır [144, 147, 149, 160].

2.3.1. YSA'nın Yapısı

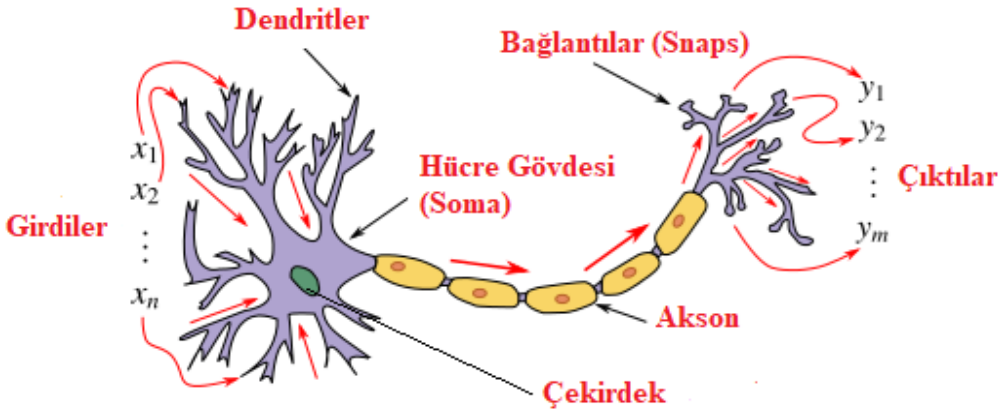
YSA'ların yapısını anlamak için ilk önce insan beyninin biyolojik yapısını anlamak gerekir.

2.3.1.1. İnsan Beynini Biyolojik Yapısı

Bir insanın beyninde yaklaşık olarak 100 milyar sinir hücresi vardır. Bu sinir hücrelerinin birbirleriyle yaptığı bağlantı sayısı da göz önünde bulundurulduğunda hücreler arası bağlantıların trilyonları bulacağı tahmin edilmektedir [149, 158, 159]. Bu sinir hücreleri, girdi bilgilerini duyu organlarından alarak daha sonra alıcı sinir hücreleri ile bu sinyalleri işleyip bir sonraki sinir hücresine aktarırlar [146, 147, 149, 160, 161].

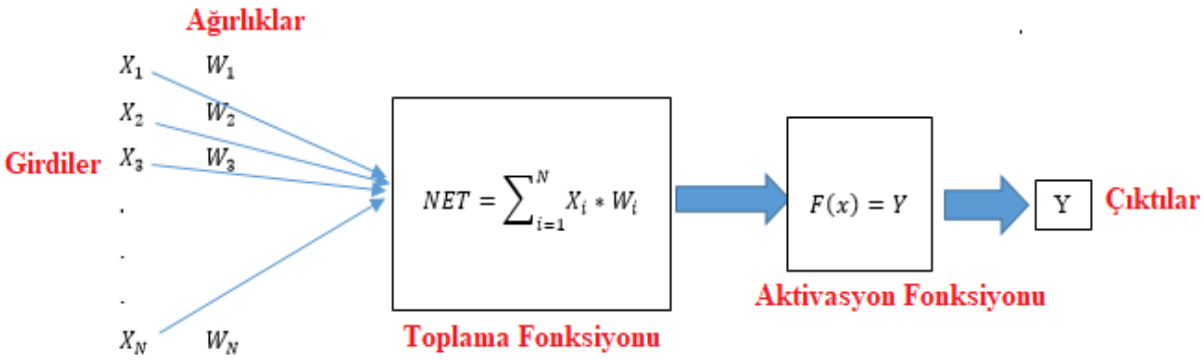
İnsan beyninin biyolojik yapısının temel yapı taşı olan nöronlar dört ana bölümden oluşmaktadır. Bunlar; dendritler, akson, hücre gövdesi (soma) ve bağlantılar (snaps)'dır. Dendritler, bağlı olduğu diğer nöronlardan veya duyu organlarından gelen sinyalleri somaya iletmekle görevlidirler. Soma, dendritler tarafından gelen sinyalleri bir araya toplayarak ve gerekli işlemleri yaparak aksona iletir. Elde edilen bu sinyaller akson tarafından işlenerek nöronun diğer

ucunda bulunan bağlantılar (snaps)'a gönderilir. Bağlantılar ise yeni üretilen sinyalleri diğer nöronlara iletir [144, 147, 149, 150, 156, 159, 160, 162–165]. Örnek bir insan sinir hücresinin çalışma mantığı Şekil 2.1'de verilmiştir.



Şekil 2.1. Örnek bir insan sinir hücresinin çalışma mantığı [163]

YSA, insan beyninden esinlendiği için yapısı da insan beyninin sinir yapısına benzemektedir. Aynı insan yapısındaki nöronlarda olduğu gibi YSA'lardaki nöronlar da aralarında bağ kurarak yapay sinir ağlarını oluştururlar. YSA'lardaki nöronlar da girdi olarak aldıkları sinyalleri işledikleri ve çıktı olarak diğer nöronlar ilettikleri bölümleri bulunmaktadır. Örnek bir yapay sinir hücresinin yapısı Şekil 2.2'de gösterilmiştir. Görüldüğü üzere bir yapay sinir hücresi girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak üzere beş bölümden oluşmaktadır [143, 147, 166–169].



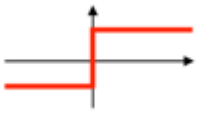
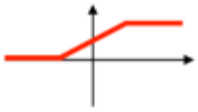
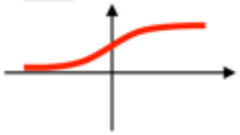
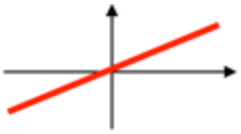
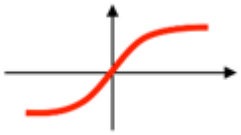
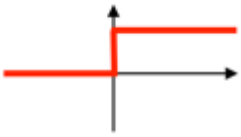
Şekil 2.2. Yapay sinir hücresinin yapısı

Girdiler: Yapay sinir hücresine dış dünyadan veya diğer bir hücreden girdiler verilebilir.
Ağırlıklar: Girdilerden gelen bilgiler bağlantıların ağırlıkları ile çarpılarak çekirdeğe gönderilir.

Toplama Fonksiyonu: Tüm girdiler ile bu girdilerin ağırlıkların çarpımının toplamından oluşan fonksiyondur. NET olarak ifade edilir.

Aktivasyon Fonksiyonu: Toplama fonksiyondan gelen bilgi girdi olarak aktivasyon fonksiyonuna verilir. Aktivasyon fonksiyonu da bu girdiyi işleyerek çıktı elde eder. Birçok aktivasyon fonksiyonu bulunmaktadır. Çizelge 2.4'te yaygın kullanılan bazı aktivasyon fonksiyonları verilmiştir. Bu aktivasyon fonksiyonları arasında seçim yapmak için genellikle doğrusal olmayan bir fonksiyon seçilir. Seçilen bu fonksiyonun türevinin kolay hesaplanabilir olması gerekir [169].

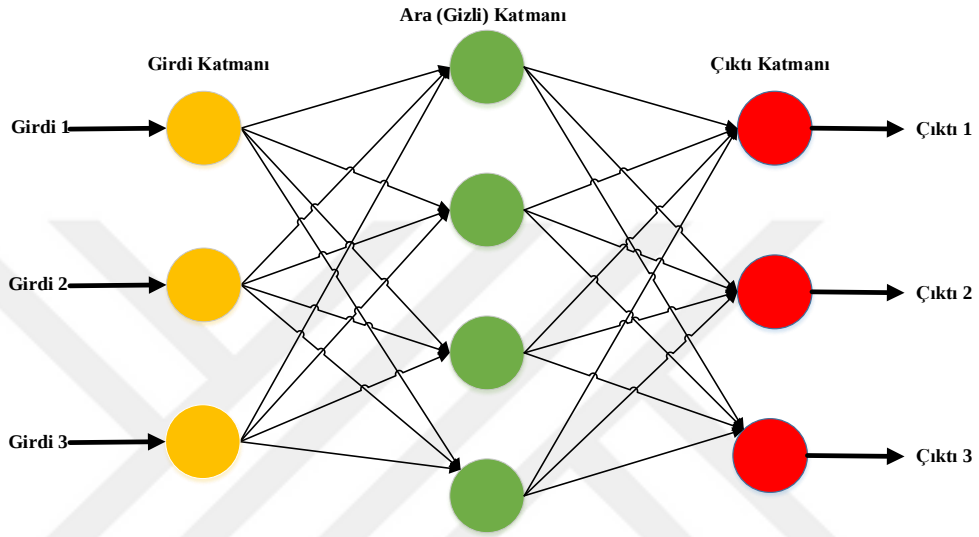
Çizelge 2.4. Bazı aktivasyon fonksiyonları [144, 147, 149, 170]

Aktivasyon Fonksiyonu	Denklemi	Grafiği
İşaret	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	
Parçalı Doğru	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	
Sigmoid	$\phi(z) = \frac{1}{1 + e^{-z}}$	
Doğrusal	$\phi(z) = z$	
Tanjant Hiperbolik	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	
Adım	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	

Hücrenin Çıktısı: Aktivasyon fonksiyonundan çıkan değer hücrenin çıktı değeri olarak ifade edilir. Bu çıktı değeri ya dış dünyaya verilir ya da ağın içinde kullanılabilir [144, 147, 149, 170, 171].

2.3.1.2. YSA'nın Katmanlı Yapısı

YSA'lar, tek bir katmandan oluşabileceği gibi birden fazla katmandan da oluşabilir. YSA'nın katmanlı yapısı; girdi katmanı, ara (gizli) katman(lar) ve çıktı katmanından oluşmaktadır [147, 166, 172, 173]. Ara katman sayısı istenilen miktarda artırılabilir. Şekil 2.3'te YSA'nın katmanlı yapısı gösterilmiştir.



Şekil 2.3. YSA'nın katmanlı yapısı

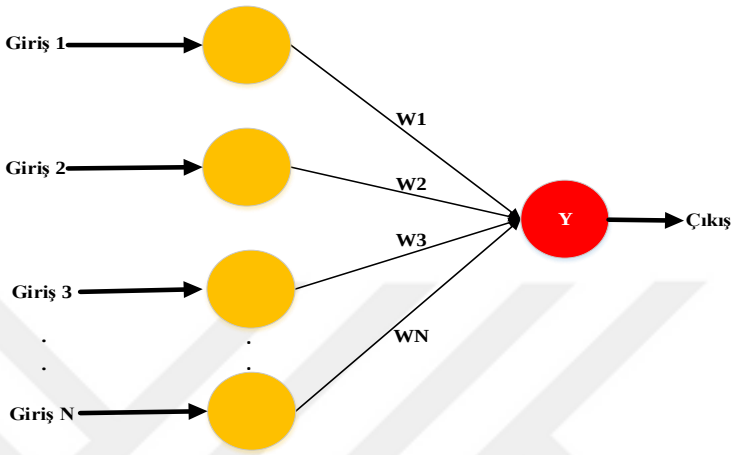
Girdi Katmanı: Dış dünyadan gelen bilgilerin girdi olarak alındığı ilk katmandır. Bu katmanda dış dünyadan gelen girdiler herhangi bir işleme tabi tutulmadan ara katman veya katmanlara işlenmek üzere gönderilir.

Ara Katman(lar): Girdi katmanından gelen bilgiler bu katman(lar) da işlenir. Ara katman sayısı istenilen miktarda artırılabilir. Örneğin; bazı yapay sinir ağlarında ara katman bulunmadığı halde bazılarında ise birden fazla ara katman bulunabilmektedir. Kısacası; ihtiyaca göre belirlenir. Ara katman ve bu katmanlardaki nöronların sayılarının artması hesaplama işleminin uzamasına neden olur. Fakat karmaşık problemlerin çözümünde kullanıldığında bu yaklaşım iyi sonuçlar verebilmektedir.

Çıktı Katmanı: Ara katmanlardan gelen bilgileri işleyerek ağın çıktılarının üretildiği katmandır. Bu katmanda üretilen çıktılar dış dünyaya verilir [12, 143, 144, 147, 149, 151, 152, 160, 167, 169, 171, 174–178].

2.3.1.3. Algılayıcı

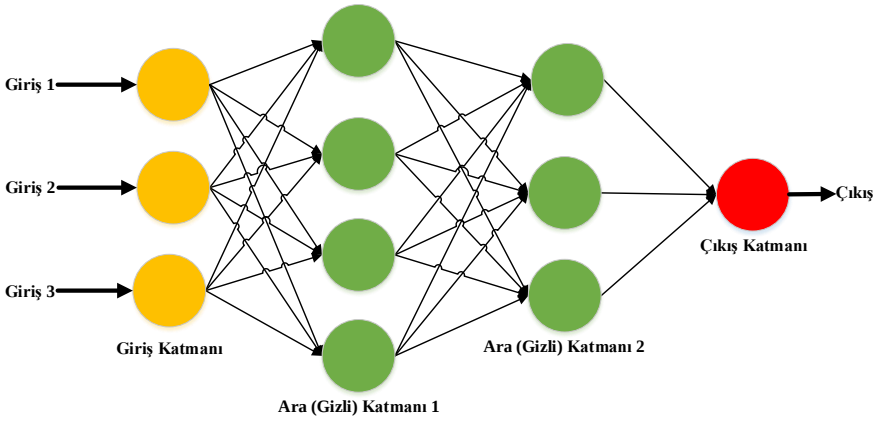
Algılayıcı, tek katmandan oluşmaktadır. Sadece giriş ve çıkışları vardır. Girişler ve çıkışlar birden fazla olabilir. YSA'nın en basit modelidir. Basit problemler için kullanışlıdır. Şekil 2.4'te tek katmanlı algılayıcı gösterilmiştir [144, 147, 171, 179].



Şekil 2.4. Tek katmanlı algılayıcı

2.3.1.4. Çok Katmanlı Algılayıcı

Çok katmanlı algılayıcı (ÇKA), birden fazla katmandan oluşmaktadır. Tek katmanlı algılayıcıya göre daha karmaşık problemleri çözebilmektedir. Buda onu kullanışlı kılan özelliğidir. ÇKA, giriş katmanı, ara katman(lar) ve çıkış katmanı olmak üzere üç seviyeden oluşmaktadır. Ara katman birden fazla katmandan da oluşabilir. Giriş katmanına dış dünyadan veriler alınır. Bu katmanda herhangi bir işlem yapılmaz. Alınan bu bilgiler ara katmana iletilir ve gerekli işlemler yapıldıktan sonra çıkış katmanına iletilir. Çıktı katmanı da ara katmandan gelen bu verileri işledikten sonra dış dünyaya çıkış olarak verir [18, 144, 147, 160, 171, 173, 179–182]. Şekil 2.5'te çok katmanlı algılayıcı gösterilmiştir.



Şekil 2.5. Çok katmanlı algılayıcı

Çizelge 2.5'te insan sinir hücresindeki her bir elamanın bilgisayar ortamında matematiksel modellenmiş hali olan YSA sinir hücresindeki karşılığı gösterilmiştir.

Çizelge 2.5. İnsan sinir hücresi ve yapay sinir hücresi [171, 180]

İnsan Sinir Hücresi	Yapay Sinir Hücresi
Nöron	İşlem Elemanı
Akson	Çıktı
Dendrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon Fonksiyonu
Sinaps	Ağırlıklar

YSA'ların avantajlarınn yanında bazı dezavantajlarıda bulunmaktadır. Çizelge 2.6'da YSA'ların temel avantaj ve dezavantajları gösterilmiştir.

Çizelge 2.6. YSA'ların temel avantaj ve dezavantajları [144, 147, 167, 171, 183–185]

YSA'ların Avantajları	YSA'ların Dezavantajları
Bilgiler ağın tamamında saklanır	Donanım bağımlıdır
Hata toleransına sahiptirler	Ağın davranışları açıklanamamaktadır
Örüntü tamamlama yapabilirler ve makine öğrenmesi gerçekleştirebilirler	Öğrenilecek problemin ağa gösterimi oldukça zordur
Kendi kendine öğrenebilme ve organize etme yetenekleri vardır	Ağın eğitiminin ne kadar süreceğine dair belli bir yöntem yoktur
Dağıtık belleğe sahiptirler	Ağ yapısının belirlenmesinde herhangi belli bir kural yoktur
Örnekleri kullanarak öğrenirler	Ağın parametresinin belirlenmesinde herhangi belli bir kural yoktur
Doğrusal olmayan problemleri çözebilirler	Sadece sayısal bilgiler ile çalışabilmektedirler
Eksik bilgi ile çalışabilmektedirler	

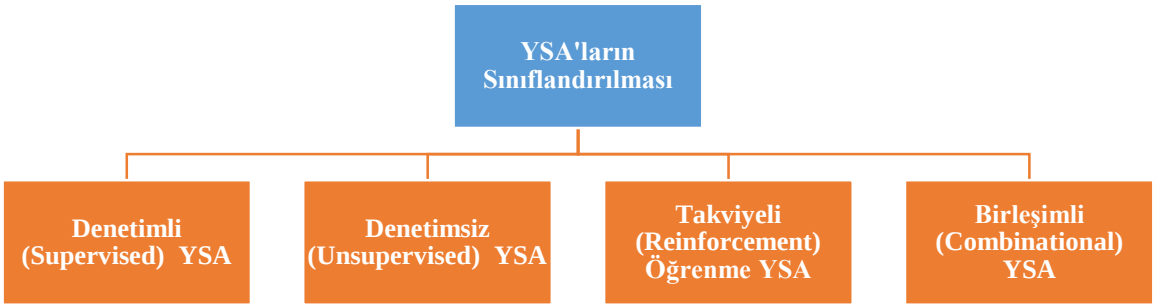
YSA'dan önce kullanılan birçok geleneksel algoritma mevcuttur. YSA'ların bu geleneksel algoritmalarla karşılaştırılması faydalı olacaktır. Çizelge 2.7'de bu karşılaştırma verilmiştir.

Çizelge 2.7. Geleneksel algoritmalar ile YSA'ların karşılaştırılması [167, 183]

Geleneksel Algoritmalar	YSA
Çok karmaşık problemleri çözemezler	Çok karmaşık problemleri çözebilirler
Hata payları yoktur.	Hata payları vardır.
Oldukça hızlıdır.	Yavaş ve donanıma bağımlıdır.
Bilgiler kesindir.	Deneyimden yararlanırlar.
Doğrusal olmayan problemleri çözemezler	Doğrusal olmayan problemleri çözebilirler
Eksik bilgi ile çalışmamaktadır	Eksik bilgi ile çalışabilmektedirler

2.3.2. YSA'ların Sınıflandırılması

YSA'ların sınıflandırılma şeması konunun daha iyi anlaşılması için Şekil 2.6'da gösterilmiştir.

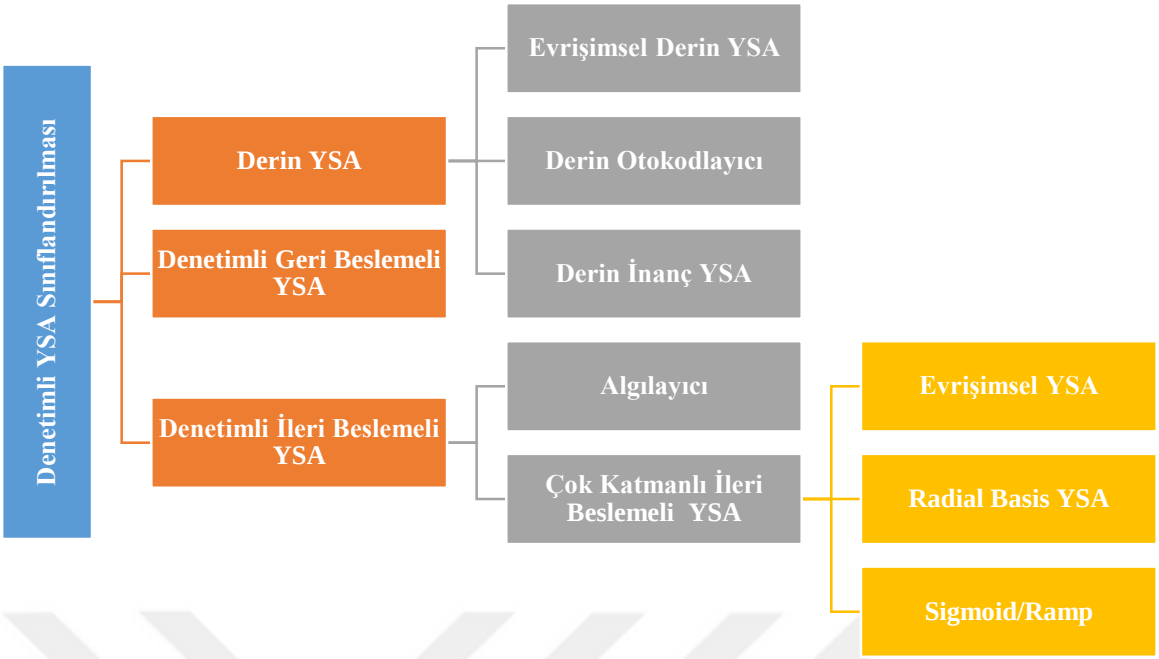


Şekil 2.6. YSA'ların sınıflandırılması [151]

YSA'ların verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenebilmesidir. Bu öğrenme işlemi; denetimli (supervised) öğrenme, denetimsiz (unsupervised) öğrenme, destekleyici (reinforcement) öğrenme ve birleşimli (combinational) öğrenme olmak üzere dört kategoriden oluşmaktadır [147, 151].

2.3.2.1. Denetimli Öğrenme

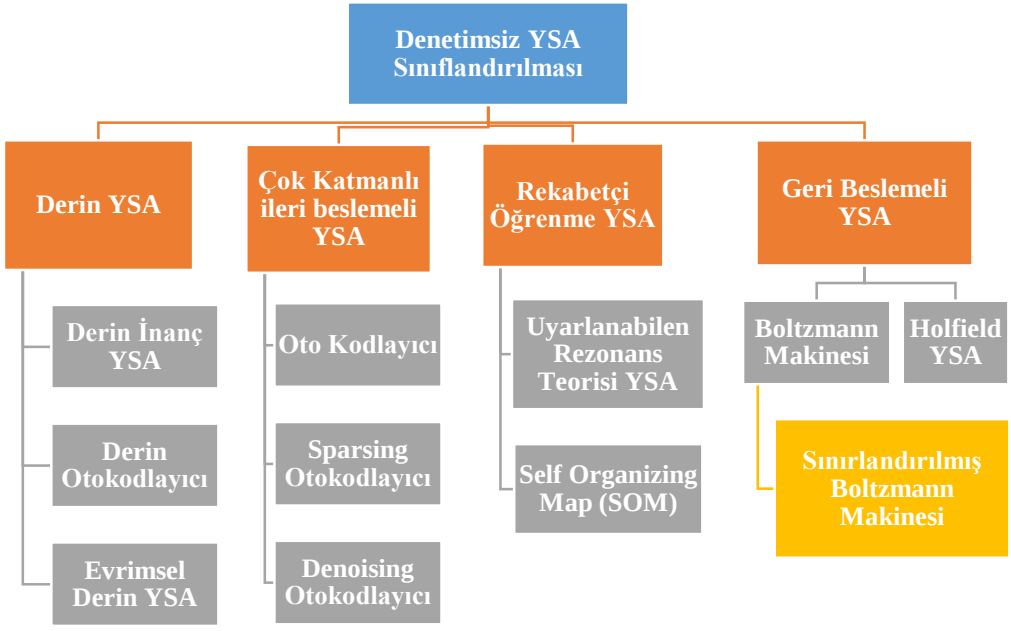
Denetimli öğrenme yönteminde ağa giriş ve çıkış değerleri aynı anda verilir. Burada çıkış değerleri önceden bellidir. Ağa verilen bu giriş değerleri istenilen çıkışları elde etmek için sürekli kendi ağırlıklarını günceller. Ağdan elde edilen çıktılar ile istenilen çıktılar (ağa çıkış olarak verilen değerler) arasındaki hata değeri hesaplanarak bulunur. Buradaki amaç; bu hata değerini en aza indirerek giriş olarak verilen değerleri sınıflandırmaktır [144, 147, 149, 151, 164, 167, 171, 186]. Şekil 2.7'de denetimli YSA sınıflandırılması gösterilmiştir.



Şekil 2.7. Denetimli YSA sınıflandırılması [151, 182]

2.3.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenme yönteminde ağa sadece giriş değerleri verilip çıkış değerleri verilmez. Bu yöntemde çıkış değerleri yoktur. Giriş değerlerine göre ağ her bir değeri kendi arasında gruplayarak veya kümeleyerek kurallarını oluşturur [144, 147, 149, 151, 164, 167, 171, 186]. Şekil 2.8'de denetimsiz YSA sınıflandırılması gösterilmiştir.



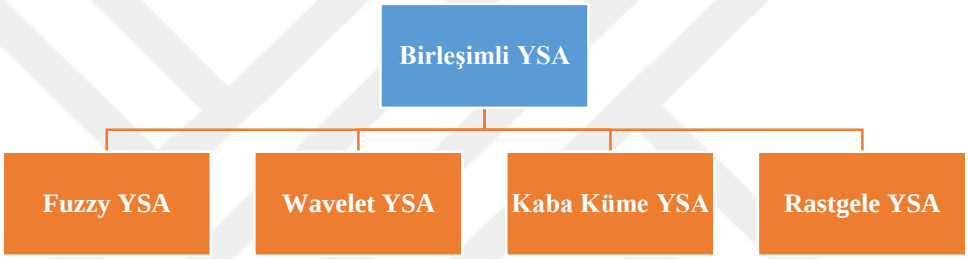
Şekil 2.8. Denetimsiz YSA sınıflandırılması [151, 182]

2.3.2.3. Takviyeli Öğrenme

Takviyeli öğrenme yönteminde ağın her bir adımında elde ettiği sonucun iyi mi/kötü mü olup olmadığına dair bilgiler verilir. Bu işlemi yaparken deneme-yanılma yönteminden faydalanır [144, 147, 151, 164, 167, 171].

2.3.2.4. Birleşimli Öğrenme

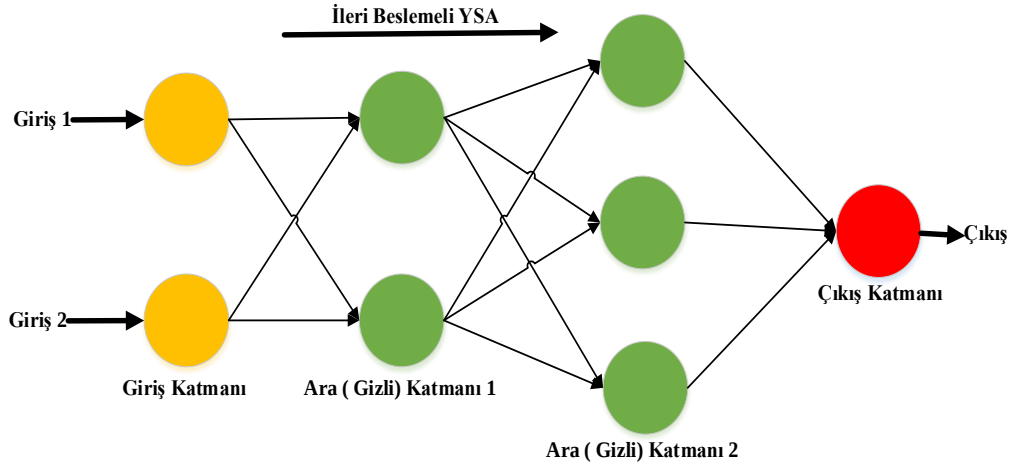
Birleşimli öğrenme daha iyi bir sistem elde etmek için diğer matematiksel modellerle YSA'ları birleştirerek kullanılan öğrenme yöntemidir [144, 151]. Şekil 2.9'da birleşimli YSA sınıflandırılması verilmiştir.



Şekil 2.9. Birleşimli YSA sınıflandırılması [151]

2.3.3. İleri Beslemeli YSA

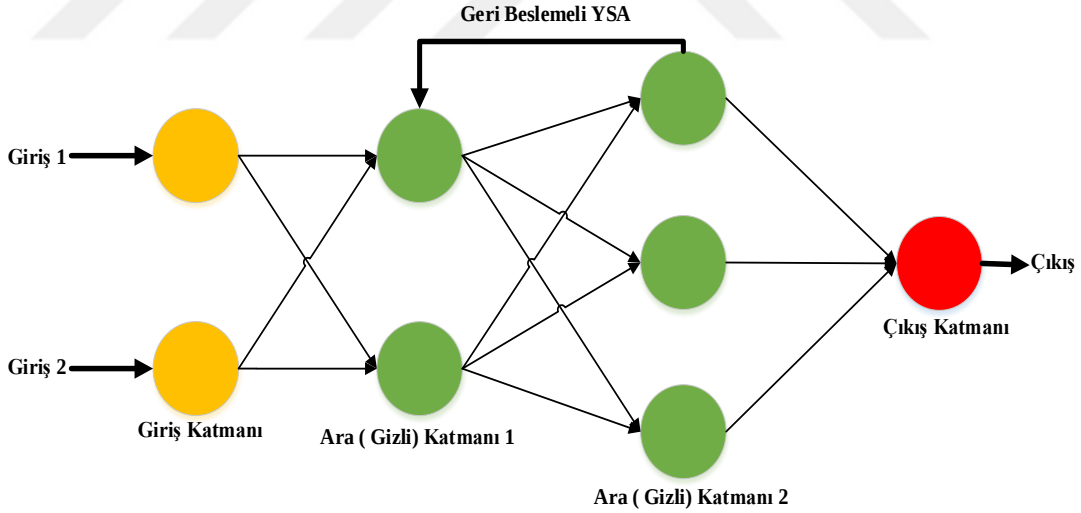
İleri beslemeli YSA'larda nöronlar girişten çıkışa doğru katmanlar halindedir. Bir katman kendinden sonraki katmanla bağlantısı bulunmaktadır. YSA'ya gelen bilgiler giriş katmanı, ara katman ve çıkış katmanından işlenerek dış dünyaya verilir [9, 17, 20, 144, 146, 147, 150–152, 156, 159, 162, 164, 167, 169, 171, 181, 184, 187–189]. Şekil 2.10'da ileri beslemeli YSA gösterilmiştir.



Şekil 2.10. İleri beslemeli YSA

2.3.4. Geri Beslemeli YSA

Geri beslemeli YSA'larda her bir hücre sadece kendinden sonra gelen hücrenin katmanına girdi olarak verilmez. Aynı zamanda, kendinden önceki katmanlarada girdi olarak verilebilir [146, 147, 151, 159, 162, 164, 167, 169, 171, 188]. Şekil 2.11'de Geri beslemeli YSA gösterilmiştir.

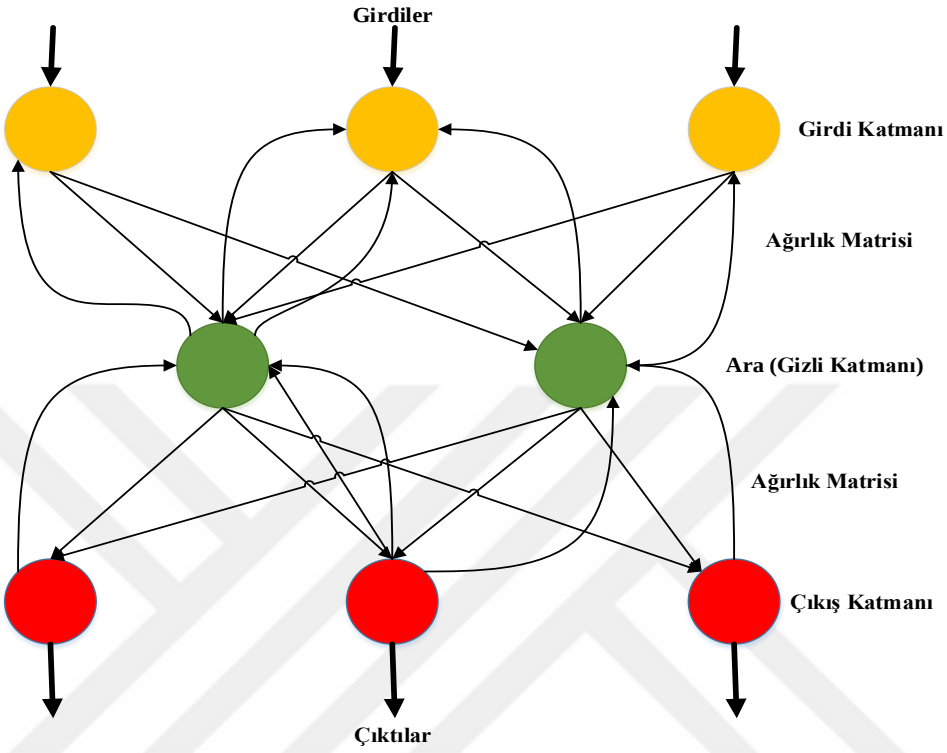


Şekil 2.11. Geri beslemeli YSA

2.3.5. Geri Yayılım Algoritması

Geri Yayılım Algoritması (GYA), denetimli öğrenme için ileri beslemeli ve çok katmanlı sinir ağlarının eğitiminde yaygın olarak kullanılan bir algoritmadır [190]. GYA, çıkış katmanında bulunan sinir hücrelerinden elde edilen hatayı geriye doğru yayarak ağırlıkların güncellenmesini

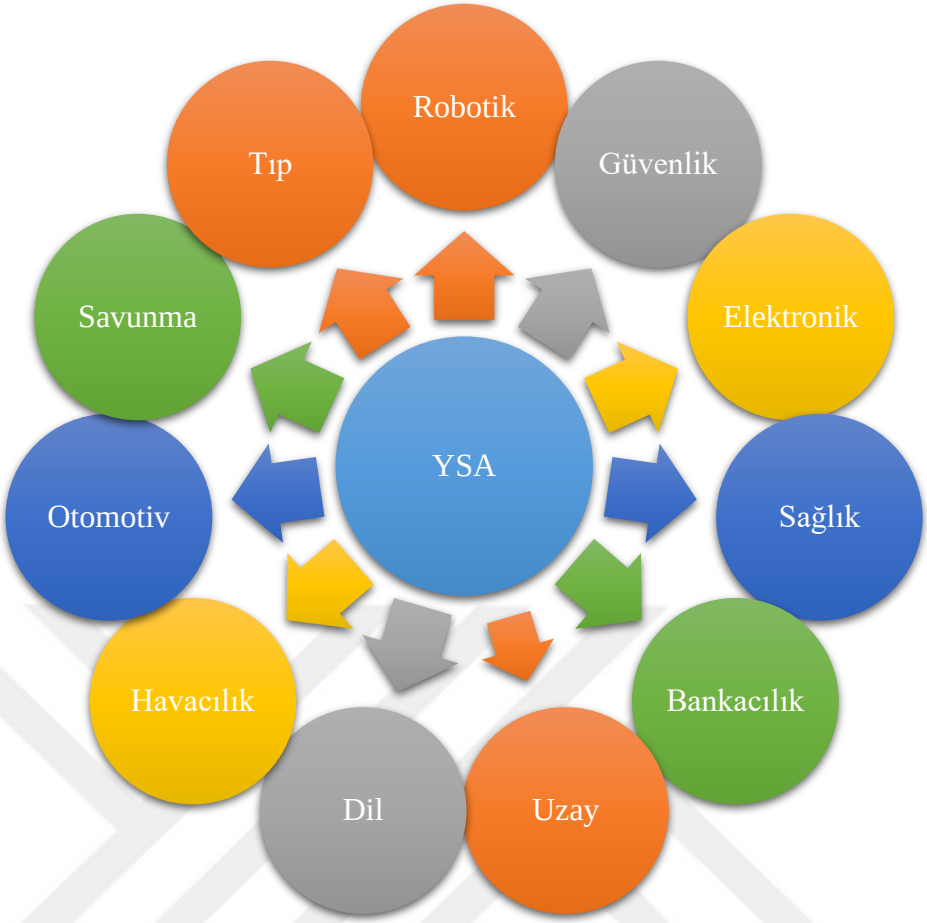
sağlayan algoritmadır [148, 159, 191]. GYA'nın asıl amacı ağırlıkları değiştirerek çıktı hatasını küçültmektir [17]. GYA, gradyan iniş ve delta kuralı gibi tekniklere dayanmaktadır [17, 142, 147, 149, 152, 156, 157, 164, 186, 190, 192–195]. Geri yayılım algoritması Şekil 2.12'de gösterilmiştir.



Şekil 2.12. Geri Yayılım Algoritması

2.3.6. YSA'ların Bazı Önemli Kullanım Alanları

YSA'lar başlıca; sınıflandırma, regrasyon, kümeleme, modelleme, veri ilişkilendirme, veri kavramsallaştırma, veri filtreleme ve tahmin uygulamalarında sıklıkla kullanılmaktadır [149, 151, 167, 169, 173, 195, 196]. Bu kullanım alanları gün geçtikçe artmaktadır [147, 159, 197]. YSA'ların bazı önemli kullanım alanları Şekil 2.13'te gösterilmiştir.



Şekil 2.13. YSA'ların bazı önemli kullanım alanları [144]

2.4. Destek Vektör Makinesi (DVM)

DVM, sınıflandırma ve regresyon problemlerinin çözümü için sıkça kullanılan bir denetimli öğrenme yöntemidir [11–13, 198–229]. DVM, oldukça yüksek genelleme yapabilme yeteneğine sahiptir [205, 206, 210, 214, 217, 219, 224, 226, 230–232]. DVM'nin amacı, iki sınıf arasındaki ayrılma marjinini maksimuma çıkarmak için doğrusal bir optimal hiper düzlem bulmaktır [19, 206, 217, 220, 230, 233–235]. Diğer bir deyişle; DVM, marjinini maksimum yapan en uygun ayırıcı düzlemi oluşturmaya çalışmaktadır [206, 207, 217, 220, 233–235]. DVM'nin en önemli avantajı yüksek oranda başarılı sonuçlar elde etme ve belleği iyi kullanabilmesidir [233, 236]. En önemli dezavantajı ise çok geç sonuç vermesidir [233, 237].

$K = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ bir eğitim seti olarak verilsin. $i = 1,2, \dots, N.\ x_i \in R^n$ ve $y_i \in \{-1,1\}$.

K'nın optimal ayrılabilir bir hiper düzlemi ($f(x) = w.x + b = 0$) olarak tanımlanabilir.

Burada;

$$f(x) = (w_0.x) + b_0 \tag{2.53}$$

$$w_0 = \sum_{j=1}^{N} y_j \alpha_j^0 x_j \tag{2.54}$$

$w_0 = (w_0^1, w_0^2, \dots, w_0^n)$ ve $x = (x^1, x^2, \dots, x^n)$ olarak verilsin. Bu iki vektörün iç çarpımı denklem (2.55)'te gösterilmiştir.

$$(w_0.x) = \sum_{i=1}^{n} w_0^i . x^i \tag{2.55}$$

Eğitim veri seti doğrusal (linear) ve doğrusal olmayan (non-linear) olmak üzere iki şekilde ayrılır [230].

Doğrusal olarak ayrılabilen durumlar için;

$$\begin{cases} \min\Phi(w,\xi) = \frac{1}{2}\left||w|\right|^2 + C\sum_{i=1}^{N}\xi_i \\ y_i[w.x_i + b] \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2,\dots,N \end{cases}$$

$$\begin{cases} \max Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j (x_i.x_j) \\ \sum_{j=1}^{N} y_j \alpha_j = 0; C \geq \alpha_i \geq 0, i = 1,2,\dots,N \end{cases}$$

Burada;

ξ_i, pozitif gevşeklik değişkenidir.

C, eğitim hatası için pozitif sabit değişkendir.

b_0, sabiti denklem (2.56)'daki gibi ifade edilir.

$$b_0 = y_i - \left(x_i . \sum_{j=1}^{N} y_j \alpha_j^0 x_j \right) \tag{2.56}$$

Denklem (2.54)'teki w_0, denklem (2.53)'te yerine yazılırsa denklem (2.57) elde edilir.

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i^0 (x_i . x) + b_0 \tag{2.57}$$

$$y_i (w_0 . x_i - b) \geq 1, i = 1,2, \dots, N.$$

Denklem (2.58)'deki optimal hiper düzlem karar fonksiyonu kullanarak iki sınıfın birbirinden doğrusal olarak ayrılıp ayrılmadığı kontrol edilebilir.

$$f(x) = sgn(w . x_i + b) = sgn(\sum_{i=1}^{N} \alpha_i y_i (x_i . x) + b) \tag{2.58}$$

Doğrusal olmayan durumlar için çekirdek (kernel) adı verilen fonksiyonlar kullanılmaktadır [222, 234, 238, 239].

K, çekirdek fonksiyonu ve $x, z \in X$ olsun.

$$K(x, z) = \langle \phi(x) . \phi(z) \rangle \tag{2.59}$$

DVM, denklem (2.59)'da da ifade edildiği gibi veri vektörlerini doğrusal olmayan bir eşleşme (ϕ) kullanarak giriş alanın (X)'ten yüksek boyutlu bir özellik alanına eşleşme yapmaktır [222, 233, 237].

ϕ, doğrusal olmayan eşleşme (nonlinear mapping).

$$\begin{cases} \max Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i . x_j) \\ \sum_{j=1}^{N} y_j \alpha_j = 0; C \geq \alpha_i \geq 0, i = 1,2, \dots, N . \end{cases}$$

Denklem (2.60)'daki optimal hiper düzlem karar fonksiyonu kullanılarak sınıfların birbirinden doğrusal olmayan olarak ayrılıp ayrılmadığı kontrol edilebilir [9, 198–214, 216, 217, 219–223, 225–230, 232–237, 240–251].

$$f(x) = sgn(w.x_i + b) = sgn(\sum_{i=1}^{N} \alpha_i y_i K(x_i.x) + b) \quad (2.60)$$

2.5. Topluluk Öğrenme (TÖ)

TÖ, sınıflandırma ve regresyon için sıkça kullanılan öğrenme yöntemidir [252, 253]. TÖ, birçok öğrenme algoritmasının birleştirilerek bir arada kullanılmasıdır [254–257]. Diğer bir deyişle; TÖ, birçok zayıf öğrenme algoritmasının birleştirilerek daha güçlü ve daha iyi sonuçlar elde eden bir öğrenme algoritmasının oluşturulması yöntemidir [11, 253, 254, 257–271]. Tek bir algoritmaya göre daha iyi sonuç vermektedir [254]. TÖ sınıflandırıcıları, en iyi sınıflandırma yöntemleri olarak bilinirler [272]. Ayrıca genelleme yetenekleri çok güçlüdür [262, 264, 265, 267, 268, 271, 273, 274]. İyi bir TÖ algoritması oluşturmak için, oldukça doğru (accurate) ve çeşitli (diversity) temel algoritmalar seçmek gerekir [262]. En yaygın kullanılan topluluk öğrenme yöntemleri Bagging ve Boosting'tır [258, 263, 264, 267, 275].

2.5.1. Bagging Öğrenme

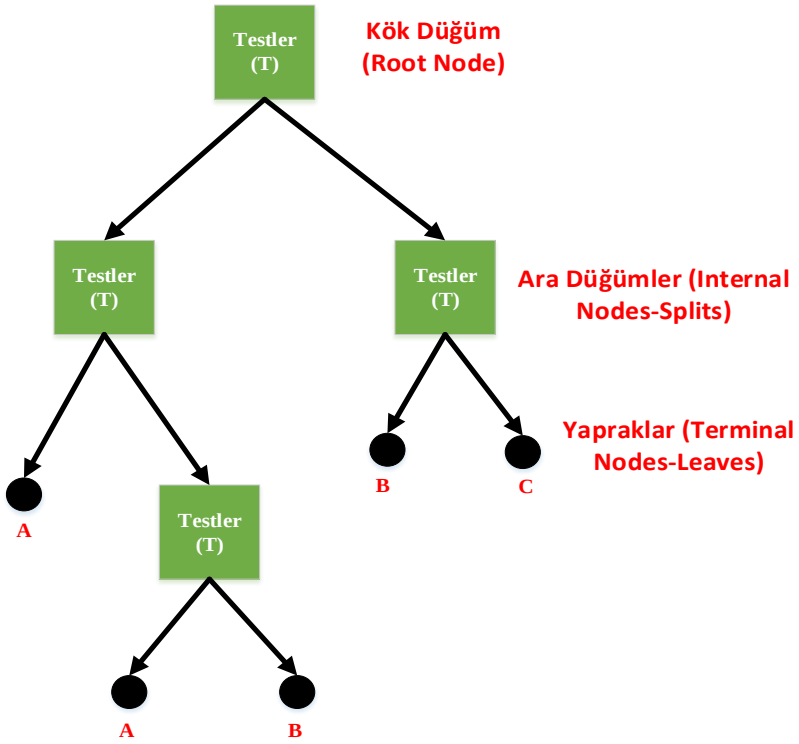
Bagging öğrenme, regresyon ve sınıflandırma için çokça kullanılan TÖ yöntemlerinden biridir [267, 275]. Eğitim setinden yeni eğitim setleri oluşturularak, temel öğrenicinin yeniden eğitilmesine olanak sağlayan bir yöntemdir [276]. Eğitim seti rastgele seçilir. Seçim yapıldıktan sonra veri, eğitim ve test olarak ikiye ayrılır. Daha sonra eğitim için ayrılan veri setinden rastgele seçim yapılır. Her eğitim seti aynı öğreniciye uygulanır ve elde edilen sonuçlar (sınıflandırma işleminde) oylanarak veya (regresyon işleminde) ortalaması alınarak birleştirilir [253, 257, 261, 262, 264, 277–279]. Bagging öğrenme yöntemi kararsız (unstable) modellerde oldukça etkilidir [253, 254, 279].

2.5.2. Boosting Öğrenme

Boosting öğrenme, regresyon ve sınıflandırma için çokça kullanılan TÖ yöntemlerinden biridir [253]. Boosting öğrenme, benzerlerine göre daha yaygın kullanılır. Hızlı çalışır ve az bellek kullanır [276]. Eğitim için oluşturulan veri setinden bir temel öğrenici rastgele seçilir. Öğrenme gerçekleştirilip, model test edilir. Test sonuçlarından yanlış sınıflandırılan örnekler belirlenir [251, 254, 267, 280, 281]. Tüm modeller sınıflandırma başarılarına göre ağırlıklandırılır ve daha sonra çıktılar oylama veya ortalama kullanılarak birleştirilir [253, 262, 276]. Örneğin, AdaBoost öğrenme algoritması en çok kullanılan boosting teknikleri arasındadır [251, 254, 261, 267, 276, 281–283].

2.6. Karar Ağacı (KA)

KA, sınıflandırma ve tahmin için sıkça kullanılan bir öğrenme yaklaşımıdır [13, 18, 284–292]. KA, kök düğümüne tüm eğitim örnekleri atanarak başlanır. Bu eğitim örnekleri, alt düğümlerin saflığını artırmak için alt düğümlere bölünür ve her bir alt düğüme atama işlemi gerçekleştirilir. Bu işlem yaprak düğümlerine ulaşıncaya kadar devam eder [289, 293–295]. Şekil 2.14'te KA sınıflandırıcısı gösterilmiştir.



Şekil 2.14. KA sınıflandırıcısı [296–298]

Şekil 2.14'te de görüldüğü üzere; her bir kutudaki testler (T), verilerin ardışık olarak küçük gruplara ayrıldığı düğümleri (nodes) ifade etmektedir. A,B,C ise her bir yaprak düğümündeki (leaf node) sınıf etiketlerini ifade etmektedir. KA'lar; kök düğümü (root node), ara düğümler (internal nodes) ve yapraklar (terminal nodes-leaves)'dan oluşmaktadır [286, 296, 299–301]. Bir KA'daki her bir düğüm sadece bir ebeveyn (parent) ve iki veya daha fazla torun (descendant)'dan oluşmaktadır [296, 299]. Terminal olmayan düğümler; amaçları temsil ederken, yapraklar (terminal düğüm) ise karar için kullanılır [300, 302–304]. Diğer bir deyişle, her düğüm bir özelliğinin (sayısal veya sembolik) testini temsil ederken, yaprak düğümü ise sınıflandırmayı temsil eder. Test örneği kökten başlayarak, her bir düğümdeki özellik değerlerini test ederek ve sınıflandırma sağlayan yaprak düğümüne ulaşıncaya kadar uygun dalda sınıflandırma yapılır [291]. KA'larının; düşük maliyetli olması, güvenli olması, anlaşılması ve yorumlanması kolay olması bakımından sıklıkla kullanılmaktadır [305]. KA'larda sınıflandırma için kullanılan eğitim verisindeki hangi alanların hangi sırada kullanılarak ağacın oluşturulacağı belirlenmelidir [239, 306, 307]. KA'larında belirsizlik ve kararsızlık önemli bir sorundur. Bunun üstesinden gelmek için yaygın olarak entropi ölçümü kullanılmaktadır [288, 291, 308, 309]. Entropi ölçümü ne kadar fazla ise çıkan sonuçlar da o oranda belirsiz ve kararsızdır. Bu yüzden, KA'larında entropi ölçüsü en az olan alanlar tercih edilmektedir [310, 311]. Entropi değeri denklem (2.61)'deki gibi hesaplanır [113, 288, 291, 298, 308–315].

$$Entropi(S) = \sum_{i}^{C} -p_i \log p_i \tag{2.61}$$

Burada;

C , hedef özellikteki değer sayısı (sınıfların sayısı) veya bir özelliğe atanan maksimum değer sayısı

p_i , i sınıfındaki örneklerin sayısı

Bir özelliğin bilgi kazancı denklem (2.62)'deki gibi hesaplanır [110, 288, 291, 298, 308–312, 314, 315].

$$Kazanç(S, A) = Entropi(S) - \sum_{V \in (A)} \frac{|S_V|}{S} Entropi(S_V) \tag{2.62}$$

Burada; A: Özellik, V: A özelliğinin bir olası değeri, S_V: V değeri için örneklerin sayısı, S: tüm veri örneklerinin sayısıdır.

Kazanç oranı denklem (2.63)'te ve Bölünmüş Bilgi ise denklem (2.64)'teki gibi hesaplanır [110, 308, 312, 314, 315].

$$Kazanç\ Oranı\ (Gain\ Ratio) = \frac{Kazanç(S,A)}{Bölünmüş\ Bilgi} \tag{2.63}$$

$$Bölünmüş\ Bilgi\ (Split\ Information) = -\sum_{i=1}^{C} \frac{S_i}{S} \log_2 \frac{S_i}{S} \tag{2.64}$$

KA'ların avantajlarının yanında bazı dezavantajları bulunmaktadır. Çizelge 2.8'de KA'ların bazı avantajları ve dezavantajları gösterilmiştir.

Çizelge 2.8. KA'ların bazı avantajları ve dezavantajları

KA'ların Bazı Avantajları	KA'ların Bazı Dezavantajları
Anlaşılabilir kurallarının olması	En uygun ağaç yapısını bulmakta zorluk çekmesi
Düşük maliyetli olması	Giriş değerleri yetersiz veya az olduğunda beklenen sonucu elde edememesi
Hızlı olması ve güvenli olması	Ağaç oluşturma ve ağaç budama karmaşıklığının fazla olması
Başarılı sonuçlar verebilmesi	Sınıf sayısı çok olduğu ve veri setindeki örneklerin az olduğu durumlarda iyi başarım elde edememesi

2.7. Temel Bileşen Analizi (TBA)

TBA, bilgisayar bilimlerinde boyut azaltmak, veri sıkıştırmak ve özellik çıkartmak için sıkça kullanılan bir yöntemdir [15, 316–327]. TBA; sinyal işleme, örüntü tanıma, görüntü işleme, yapay zekâ ve makine öğrenmesi tekniklerinde sıkça kullanılmaktadır [322, 328–330]. TBA, var olan verinin daha az sayıda boyutla ifade etmesi, fazla öneme sahip olmayan boyutların çıkarılması ve önemli olan boyutların kullanılması olayı olarak da ifade edilebilir [331, 332].
Bir dizi $x_1, x_2, x_3, \dots, x_m \in R^n (n < m)$ giriş vektörü verilmiş olsun [333].

$$\sum_{t=1}^{m} X_t = 0 \tag{2.65}$$

Bu vektörlerin kovaryans matrisi denklem (2.66)'daki gibi hesaplanır [317, 333–338].

$$C = \frac{1}{m}\sum_{t=1}^{m} x_t \, x_t{}^T \tag{2.66}$$

Temel bileşenler; C kovaryans matrisinin özdeğerleri çözülerek hesaplanır [333].

$$\lambda_t u_t = C u_t$$

Burada;

λ_t, C kovaryas matrisinin öz değerinden biridir.

u_t, öz vektörlerin özdeşidir.

Düşük boyutlu ham verileri elde etmek için en büyük k öz değerine denk gelen ilk k öz vektörü hesaplanır. k sayısını elde etmek için; en büyük k öz vektörlerinin yaklaşık hassasiyeti için denklem (2.67)'de verilmiş olan θ eşik değeri kullanılır [318, 333].

$$\frac{\sum_{t=1}^{k} \lambda_t}{\sum_{t=1}^{m} \lambda_t} \geq \theta \tag{2.67}$$

θ, eşik değerine bakarak; k öz vektörlerin sayısına karar verilir.

$$U = [\, u_1, u_2, u_3, \dots, u_k] \;\; ve \; \Lambda = [\, \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_k] \tag{2.68}$$

Denklem (2.68)'de verilmiş olan U; öz vektörleri, Λ ise öz değerleri göstermektedir [327, 334, 336, 339–341]. Düşük boyutlu ham özellik vektörleri denklem (2.69)'daki gibi hesaplanır [333, 341].

$$S = U^T x_t \tag{2.69}$$

2.8. Saldırı Tespit Sistemi (STS)

STS, ağ üzerinden yapılan her türlü saldırılara karşı bilişim sistemlerinin korunması olarak ifade edilebilir [3, 319, 342]. Diğer bir deyişle; STS, anında (zamanında) veya gerçek zamanlı olarak yapılan tüm saldırıları algılayarak ve ilgili birime uyarı veya mesaj yollayarak bilgilendirilmesi olarak ifade edilebilir [343]. STS'nin amacı normal ve anormal durumları tanımlamak veya herhangi bir şüpheli durum sezildiğinde (sms, mail veya diğer yollarla) sistem yöneticisini haberdar etmektir [342, 344]. STS kullanılarak, ağ üzerinden yapılan saldırılar tespit edilebilmekte ve ilgili mekanizmalar harekete geçirilerek engellenebilmektedir. STS'lerde geleneksel yöntemlerin yanı sıra, makine öğrenmesi, veri madenciliği, YSA ve derin öğrenme yöntemleri çokça kullanılmaya başlanmıştır. STS'ler, imza tabanlı STS ve anormallik tabanlı STS olmak üzere iki kategoriye ayrılır [11, 13, 14, 16, 20, 23, 25, 230, 266, 298, 312, 319, 342].

2.8.1. İmza Tabanlı STS

İmza tabanlı sistemlerde her davranışın bir imzası vardır. Daha önceden toplanmış olan saldırılar bir veri tabanında toplanmaktadır. Herhangi bir bilgi geldiğinde veri tabanına bakılarak saldırı olup olmadığına göre sınıflandırılır. Bu yüzden imza tabanlı sistemlerde yanlış alarm verme ihtimalleri yoktur. İmza tabanlı sistemlerinin dezavantajlarından biri veri tabanında olmayan bir saldırı meydana gelirse bu saldırıyı fark edemezler [11, 13, 14, 16, 20, 21, 23, 25, 230, 266, 269, 298, 312, 313, 319, 342, 345].

2.8.2. Anormallik Tabanlı STS

Bilişim sistemlerinde meydana gelen anormal durumları, normal durumlardan ayırt etmesi olarak ifade edilebilir. Normal durum, sistemin detaylı bir biçimde analiz edilmesi ile elde edilir. Normal durum kuralı belirlendikten sonra, bu kurallar dışındaki davranışları saldırı olarak tespit eder [11, 13, 14, 16, 20, 21, 23, 230, 298, 312, 342, 345]. Bu kurala göre, gelen bilgi normal veya anormal olarak sınıflandırılır. Anormallik tespitinin en önemli avantajlarından biri daha önceden bilmediği saldırıları yakalayabilmesidir [25, 266, 269]. Anormalik tespitinin en önemli dezavantajlarında biri ise yanlış alarm (false positive/negative) verebilmesidir [312, 313, 319, 342]. Yani, normalde saldırı olmayan bir davranışı saldırı olarak ya da saldırı olan bir olayı saldırı olarak algılamama olasılığı yüksektir [319, 345]. Çizelge 2.9'da imza tabanlı STS ile anormallik tabanlı STS karşılatırılması gösterilmiştir.

Çizelge 2.9. İmza tabanlı STS ile anormallik tabanlı STS karşılatırılması [13, 21, 346]

Anormallik Tabanlı STS	**İmza Tabanlı STS**
Kullanıcının davranışlarını modeller	Saldırganların davranışlarını modeller
Bütün saldırıların tespiti amaçlanır	Sadece veri tabanında olan saldırıların tespiti amaçlanır.
Yanlış alarm verme olasılığı yüksektir	Yanlış alarm verme olasılığı yoktur
Bilmediği bir saldırı meydana geldiğinde bu saldırıyı yakalayabilir	Bilmediği bir saldırı meydana geldiğinde bu saldırıyı yakalayamaz

2.9. KDD'99 (KDD Cup 1999) Veri Seti

DARPA tarafından desteklenen ilk çalışma Massachusetts Teknoloji Üniversitesi (MIT) Lincoln laboratuvarı tarafından 1998 yılında gerçekleştirilmiştir [12, 15, 17, 21, 25, 230, 298, 312, 342, 346–349]. DARPA, hem eğitim/öğrenme hem de test işlemlerini gerçekleştirmek amacıyla kullanılan bir veri setidir [298]. DARPA veri seti, bir takım ön işlemlerden geçirilerek (özellik çıkarma vb.) KDD'99 veri seti elde edilmiştir [3, 319, 346, 350]. KDD'99, son yıllarda STS'lerin araştırılmasında, yaygın olarak kullanılmaya başlanmıştır [319, 351]. KDD'99 veri setinin çokça kullanılmasının amacı saldırı tespiti için eğitim ve test işlemleri bakımından kolaylıklar sağlamasıdır [3, 344, 345].

STS'ler için DARPA veri setinin kullanılabilmesi için çok fazla önişleme ihtiyaç vardır. Bu tez çalışmasında; DARPA veri setinin ön işlemlerden geçirilerek elde edilen KDD'99 veri seti kullanılmıştır. KDD'99 veri seti kullanılarak, eğitim ve test sonuçları daha hızlı alınabilmektedir [3, 344, 345]. KDD'99 eğitim veri setinde 24 adet saldırı türü ve test veri setinde ise 14 saldırı türü olmak üzere toplamda 38 saldırı türü bulunmaktadır [3, 344–348]. KDD'99, 9 temel ve 32 adet türetilmiş olmak üzere toplamda 41 adet özellikten oluşan bir veri setidir [3, 9, 350, 352–356, 11, 14, 18–20, 25, 344, 345]. KDD'99 veri seti; temel özellikler (basic features), içerik özellikler (content features), zaman tabanlı trafik özellikler (time-based traffic features) ve sunucu tabanlı trafik özellikler (host-based traffic features) olmak üzere dört kategoriye ayrılmaktadır [3, 344, 345, 348, 357]. Bu tez çalışmasında: KDD100 (kddcup.data.gz), KDD10 (kddcup.data_10_percent.gz) ve KDDTEST (corrected.gz) olmak üzere üç farklı KDD'99 veri dosyası kullanılmıştır [343, 352]. KDD100, 4898431 örnekten, KDD10, 494021 örnekten ve KDDTEST ise 311029 örnekten oluşmaktadır [343, 346, 350, 352, 354]. Veri setinin temel özellikleri, bireysel TCP bağlantılarından sağlanmaktadır. TCP/IP bağlantısından çıkarılabilen tüm özellikleri içerir [348]. Çizelge 2.10'da KDD'99 veri setinin temel özellikleri gösterilmiştir.

Çizelge 2.10. KDD'99 veri setinin temel özellikleri [2, 348, 357]

Özellik No	Özellik Adı	Tanım	Tip
1	duration	Bağlantı uzunluğu (Saniye Sayısı)	Sürekli
2	protocol_type	Protokol tipi (örnek: udp, tcp)	Ayrık
3	service	Hedef üzerine ağ hizmeti (örnek: http, telnet)	Ayrık
4	src_bytes	Kaynaktan hedefe veri baytlarının sayısı	Sürekli
5	dst_bytes	Hedeften kaynağa veri baytlarının sayısı	Sürekli
6	flag	Bağlantının normal veya hata durumu	Ayrık
7	land	Bağlantı aynı sunucudan (from the same host) veya aynı portta (to the same port) ise 1 değilse 0	Ayrık
8	Wrong_fragment	Yanlış parça (fragment) sayısı	Sürekli
9	urgent	Acil paket sayısı	Sürekli

KDD'99 veri setinin içerik özellikleri alan (domain) bilgisi ile elde edilmektedir[348]. Çizelge 2.11'de KDD'99 veri setinin içerik özellikleri gösterilmiştir.

Çizelge 2.11. KDD'99 veri setinin içerik özellikleri [2, 348, 357]

Özellik No	Özellik Adı	Tanım	Tip
10	hot	"hot" gösterge sayısı	Sürekli
11	num_failed_logins	Başarısız giriş sayısı	Sürekli
12	Logged_in	Giriş başarılı ise 1 değilse 0	Ayrık
13	num_compromised	Riskli koşulların sayısı	Sürekli
14	root_shell	"Root Shell" elde edildiyse 1 değilse 0	Ayrık
15	su_attempted	"Su Root" komutu girildiğinde/denendiğinde 1 değilse 0	Ayrık
16	num_root	"Root" erişim sayısı	Sürekli
17	num_file_creations	Dosya oluşturma işlemlerinin sayısı	Sürekli
18	num_shells	Kabuk istemleri (Shell prompts) sayısı	Sürekli
19	num_access_files	Erişim kontrol dosyalarındaki işlem sayısı	Sürekli
20	num_outbound_cmds	Bir ftp oturumundaki giden komutların sayısı	Sürekli
21	is_hot_login	Giriş "hot" listesine ait ise 1 değilse 0	Ayrık
22	is_guest_login	Giriş misafir (guest) girişi ise 1 değilse 0	Ayrık

KDD'99 veri setinin zaman tabanlı trafik özellikleri, aynı sunucu ve aynı hizmet özellikleri kullanılarak elde edilen özelliklerdir [3, 344, 345, 348]. Aynı sunucu özellikleri, yalnızca son iki saniyede geçerli bağlantıyla aynı hedef ana bilgisayara sahip bağlantıları inceler ve protokol davranışı, hizmet vb. ile ilgili istatistikleri hesaplar [344, 348]. Aynı hizmet özellikleri, yalnızca son iki saniye içinde geçerli bağlantıyla aynı hizmete sahip bağlantıları inceler [3, 344, 345, 348]. Çizelge 2.12'de KDD'99 veri setinin zamana bağlı trafik özellikleri ve Çizelge 2.13'de KDD'99 veri setinin sunucu tabanlı trafik özellikleri gösterilmiştir.

Çizelge 2.12. KDD'99 veri setinin zaman tabanlı trafik özellikleri [2, 345, 348, 357]

Özellik No	Özellik Adı	Tanım	Tip
23	count	Son iki saniyedeki mevcut bağlantıyla aynı sunucu (same host) olan bağlantıların sayısı **Aynı sunucu bağlantıları (same host connections) aşağıda verilmiştir.**	Sürekli
24	serror_rate	"SYN" hatalarına sahip olan bağlantılarının yüzdesi	Sürekli
25	rerror_rate	"REJ" hatalarına sahip olan bağlantıların yüzdesi	Sürekli
26	same_srv_rate	Aynı hizmete olan bağlantıların yüzdesi	Sürekli
27	diff_srv_rate	Farklı hizmetlere olan bağlantıların yüzdesi	Sürekli
28	srv_count	Son iki saniyedeki mevcut bağlantıyla aynı hizmete (same service) olan bağlantıların sayısı **Aynı hizmet bağlantıları (same service connections) aşağıda verilmiştir.**	Sürekli
29	srv_serror_rate	"SYN" hataları olan bağlantıların yüzdesi	Sürekli
30	srv_rerror_rate	"REJ" hataları olan bağlantıların yüzdesi	Sürekli
31	srv_diff_host_rate	Farklı sunuculara olan bağlantıların yüzdesi	Sürekli

Çizelge 2.13. KDD'99 veri setinin sunucu tabanlı trafik özellikleri [357]

Özellik No	Özellik Adı	Tanım	Tip
32	destination_host_rerror_rate	"REJ" bayrağını etkinleştiren eşdeğer port numarasına karşılık gelen ilişkilendirme sayısı temsil eder	Sürekli
33	destination_host_serror_rate	(Özellik No:36) için tetiklenen bayrak sayısı	Sürekli
34	destination_host_srv_serror_rate	(Özellik No:39) için tetiklenen bayrak sayısı	Sürekli
35	destination_host_srv_rerror_rate	(Özellik No:39) için aktif "REJ" bayrak sayısı	Sürekli
36	destination_host_count	Aynı sunucu (same host) hedef IP adreslerini temsil etmek için kullanılır	Sürekli
37	destination_host_same_src_port_rate	(Özellik No:39) için eşdeğer kaynak port sayısı	Sürekli
38	destination_host_diff_srv_rate	(Özellik No:39) için farklı port sayısı	Sürekli
39	destination_host_srv_count	Eşdeğer port numarasına sahip ilişkilendirme sayısı.	Sürekli
40	destination_host_same_srv_rate	(Özellik No:36)'e ait eşdeğer hizmet için ilişkilendirme sayısı	Sürekli
41	destination_host_srv_diff_host_rate	(Özellik No:39) için farklı hedef IP adresleri için ilişkilendirme sayısı	Sürekli

KDD'99 veri setinde birçok saldırı tipi örneği mevcuttur. Çizelge 2.14'te KDD'99 veri setinin birkaç saldırı tipi örneği gösterilmiştir.

Çizelge 2.14. KDD'99 veri setinin birkaç saldırı tipi örneği [352]

Saldırı Tipi	Örnekler
normal	0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal
neptune	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,201,1,1.00,1.00,0.00,0.00,0.00,0.06,0.00,255,1,0.00,0.08,0.00,0.00,1.00,1.00,0.00,0.00,neptune
warezclient	1,tcp,ftp,SF,1267,2451,0,0,0,28,0,1,0,0,0,0,0,0,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,110,8,0.07,0.05,0.01,0.00,0.02,0.00,0.07,0.00,warezclient
satan	0,udp,private,SF,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.00,0.00,0.00,0.08,0.69,0.00,255,1,0.00,0.26,1.00,0.00,0.00,0.00,0.00,0.00,satan

KDD'99 veri seti altında bulunan KDD10 veri setinde toplamda 494021 veri örneği, KDDTEST veri setinde toplamda 311029 veri örneği ve KDD100 veri setinde ise toplamda 4898431 veri örneği bulunmaktadır. Bu veri setlerinde bulunan normal ve saldırı tiplerinin miktarları ve kategorileri Çizelge 2.15'te gösterilmiştir.

Çizelge 2.15. KDD10, KDDTEST ve KDD100 veri setlerinde bulunan normal ve saldırı tiplerinin miktarları ve kategorileri [16, 18, 230, 313, 358]

	KDD10		**KDDTEST**		**KDD100**	
Saldırı Tipi	**Miktarı**	**Kategori**	**Miktarı**	**Kategori**	**Miktarı**	**Kategori**
apache2	-	-	794	DoS	-	-
back	2203	DoS	1098	DoS	2203	DoS
buffer_overflow	30	U2R	22	U2R	30	U2R
ftp_write	8	R2L	3	R2L	8	R2L
guess_passwd	53	R2L	4367	R2L	53	R2L
httptunnel	-	-	158	R2L	-	-
imap	12	R2L	1	R2L	12	R2L
ipsweep	1247	probe	306	probe	12481	probe
land	21	DoS	9	DoS	21	DoS
loadmodule	9	U2R	2	U2R	9	U2R
mailbomb	-	-	5000	DoS	-	-
mscan	-	-	1053	probe	-	-
multihop	7	R2L	18	R2L	7	R2L
named	-	-	17	R2L	-	-
neptune	107201	DoS	58001	DoS	1072017	DoS
nmap	231	probe	84	probe	2316	probe
normal	97278	normal	60593	normal	972781	normal
perl	3	U2R	2	U2R	3	U2R
phf	4	R2L	2	R2L	4	R2L
pod	264	DoS	87	DoS	264	DoS
portsweep	1040	probe	354	probe	10413	probe
processtable	-	-	759	DoS	-	-
ps	-	-	16	U2R	-	-
rootkit	10	U2R	13	U2R	10	U2R
saint	-	-	736	probe	-	-
satan	1589	probe	1633	probe	15892	probe
sendmail	-	-	17	R2L	-	-
smurf	280790	DoS	164091	DoS	2807886	DoS
snmpgetattack	-	-	7741	R2L	-	-
snmpguess	-	-	2406	R2L	-	-
spy	2	R2L	-	-	2	R2L
sqlattack	-	-	2	U2R	-	-
teardrop	979	DoS	12	DoS	979	DoS
udpstorm	-	-	2	DoS	-	-
warezclient	1020	R2L	-	-	1020	R2L
warezmaster	20	R2L	1602	R2L	20	R2L
worm	-	-	2	R2L	-	-
xlock	-	-	9	R2L	-	-
xsnoop	-	-	4	R2L	-	-
xterm	-	-	13	U2R	-	-
Toplam	**494021**		**311029**		**4898431**	

Daha önceden belirtildiği üzere, KDD'99; KDD10, KDDTEST ve KDD100 olmak üzere üç veri setinden oluşmaktadır. KDD10, KDDTEST ve KDD100 veri setlerinin veri miktarları ile normal ve saldırı (DoS, Probe, U2R, R2L) tiplerinin yüzdelik oranları Çizelge 2.16'da gösterilmiştir.

Çizelge 2.16. KDD'99 veri setinde bulunan veri setlerinin veri miktarları ile normal ve saldırı (DoS, Probe, U2R, R2L) tiplerinin yüzdelik oranları [230, 352]

Veri Seti Adı	Veri Miktarı	Normal	Saldırı Tipleri			
			DoS	Probe	U2R	R2L
KDD10	494021	%19.69	%79.23	%0.83	%0.01	%0.22
KDDTEST	311029	%19.48	%72.29	%1.33	%0.02	%5.25
KDD100	4898431	%19.85	%79.27	%0.83	%0.001	%0.02

2.10. KDD'99 Veri Setinde Bulunan Saldırı Tipleri

KDD'99 veri setinde bulunan saldırı tipleri; DoS, U2R, R2L ve Probe olmak üzere dört kategoriye ayrılmıştır [9, 11–17, 19, 20, 23, 25, 298, 313, 342–345, 347, 348, 350, 353, 355, 358].

2.10.1. Hizmet Engelleme (DoS)

DoS saldırıları, sistem kaynaklarına gereğinden fazla istek gönderilmesi bazı bilgi işlem (computing) ve bellek (memory) gibi kaynakların çok aşırı yüklenmesini sağlamak amacıyla gerçekleştirilen saldırılardır [9, 25, 342, 345, 347, 350, 353, 355]. Sistemin tüm kaynaklarını tüketip hizmet verilemez hale getirmesi sağlanıp sisteme erişim engellenir [343, 347]. DoS saldırıları, en tehlikeli saldırılardan bir tanesidir [16, 359]. DoS saldırı tiplerine örnek olarak; back, pod, land, smurf, apache2, neptune ve mailbomb verilebilir [7, 9, 11, 12, 19, 25, 347, 355, 358].

2.10.2. Uzak Bir Makineden Yerel Ağda Oturum Açma (R2L)

R2L saldırıları, saldırganlar bir makineye ağ üzerinden paketler gönderilerek makineye erişimi olmamasına rağmen sistemin güvenlik açıklarından faydalanarak bu makinenin kullanıcısıymış gibi davranarak sistemi ele geçirerek yapılan saldırılardır [7, 9, 355, 359, 360, 12, 16, 25, 342, 343, 347, 350, 353]. R2L saldırı tipine örnek olarak; named, phf, xlock, multihop, ftp_write ve worm verilebilir [7, 11, 19, 22, 25, 347, 355, 358].

2.10.3. Kullanıcı Hesabını Admin Hesabına Yükseltme (U2R)

U2R saldırıları, sistemde sadece normal kullanıcı yetkisine sahip olmasına rağmen saldırganların adminin sahip olduğu (parola vb.) yetkilerini elde ederek yapılan saldırılardır [312, 342, 343, 350, 353, 355]. Bu saldırıyı düzenlerken sistemin güvenlik açıklarından faydalanır [7, 9, 12, 16, 25, 342, 347, 355, 359, 360]. U2R saldırı tipine örnek olarak; perl, rootkit, sqlattack, buffer_overflow ve leadmodule verilebilir [7, 11, 19, 25, 347, 355, 358].

2.10.4. Bilgi Tarama (Probe)

Bilgi tarama saldırıları, saldırı yapacağı sistemin güvenlik açıklarından faydalanarak sistemi tarayıp gerekli bilgileri elde eder. Daha sonra sisteme nasıl bir saldırı yapacağı belirlenir [7, 9, 12, 16, 25, 312, 342, 343, 345, 347, 350, 353, 355, 360]. Bu saldırı tipine örnek olarak; ipsweep, nmap, satan, portsweep ve saint verilebilir [7, 11, 19, 25, 347, 355, 358].

2.11. KDD10 ve KDD100 Veri Setlerinin Ön İşlem Aşamaları

Bu tez çalışmada, KDD'99 veri seti altında bulunan KDD10, KDD100 ve KDDTEST olmak üzere üç adet veri seti kullanılmıştır. KDD10, KDD100 ve KDDTEST veri setleri etiketli verilerden oluşmaktadır. Bundan dolayı KDD10, KDD100 ve KDDTEST veri setlerinde bulunan verilerin hangi saldırılara ait olduklarıda verilmiştir. KDD10 veri seti 494021 örnekten, KDD100 veri seti 4898431 ve KDDTEST veri seti ise 311029 örnekten oluşmaktadır. Bu veri setlerinde bulunan "protocol_type", "service" ve "flag" alanları metin (string) formatta olup, diğer alanlar ise sayısal formattadır. Veri setleri üzerinde işlem yapabilmek için, veri setlerinde bulunan tüm alanların sayısal formatta olması gerekmektedir. Bu yüzden metin formatta olan "protocol_type", "service" ve "flag" alanlarının her birine sayısal değerler verilmiştir. KDD10 ve KDD100 veri setlerinin örnek formatları Çizelge 2.17'de gösterilmiştir.

Çizelge 2.17. KDD10 ve KDD100 veri setlerinin örnek formatları

Özellik Adı	Örnek 1	Örnek 2
duration	0	0
protocol_type	icmp	tcp
service	ecr_i	http
flag	SF	SF
src_bytes	1032	181
.	.	.
.	.	.
.	.	.
dst_host_srv_rerror_rate	0	0
attack_type	smurf	normal

KDD10 ve KDD100 veri setlerindeki "attack_type" normal ve bütün saldırı tiplerinin sayısal formata dönüştürülmesi Çizelge 2.18'de gösterilmiştir.

Çizelge 2.18. KDD10 ve KDD100 veri setlerindeki "attack_type" normal ve bütün saldırı tiplerinin sayısal formata dönüştürülmesi

Saldırı Tipi	Sayısal Değeri
Normal	1
Bütün Saldırı Tipleri	2

Çizelge 2.18'de de görüldüğü gibi normal için 1 sayısal değeri verilirken, diğer bütün saldırı tiplerine de 2 sayısal değeri verilmiştir. Bunun yapılmasındaki amaç saldırı olup olmadığını tespit etmektir. KDD10 ve KDD100 veri setlerindeki "protocol_type" adlarının sayısal formata dönüştürülmesi ise Çizelge 2.19'da gösterilmiştir.

Çizelge 2.19. KDD10 ve KDD100 veri setlerindeki "protocol type" adlarının sayısal formata dönüştürülmesi

Protocol_Type	Sayısal Değeri
icmp	1
tcp	2
udp	3

KDD10 ve KDD100 veri setlerindeki "flag" adlarının sayısal formata dönüştürülmesi Çizelge 2.20'de gösterilmiştir.

Çizelge 2.20. KDD10 ve KDD100 veri setlerindeki "flag" adlarının sayısal formata dönüştürülmesi

Flag Adı	Sayısal Değeri
OTH	1
REJ	2
RSTO	3
RSTOS0	4
RSTR	5
S0	6
S1	7
S2	8
S3	9
SF	10
SH	11

KDD10 veri setindeki "service" adlarının sayısal formata dönüştürülmesi Çizelge 2.21'de gösterilmiştir.

Çizelge 2.21. KDD10 veri setindeki "service" adlarının sayısal formata dönüştürülmesi

Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri
IRC	1	finger	18	netbios_ns	35	sql_net	52
X11	2	ftp	19	netbios_ssn	36	ssh	53
Z39_50	3	ftp_data	20	netstat	37	sunrpc	54
auth	4	gopher	21	nnsp	38	supdup	55
bgp	5	hostnames	22	nntp	39	systat	56
courier	6	http	23	ntp_u	40	telnet	57
csnet_ns	7	http_443	24	other	41	tftp_u	58
ctf	8	imap4	25	pm_dump	42	tim_i	59
daytime	9	iso_tsap	26	pop_2	43	time	60
discard	10	klogin	27	pop_3	44	urh_i	61
domain	11	kshell	28	printer	45	urp_i	62
domain_u	12	ldap	29	private	46	uucp	63
echo	13	link	30	red_i	47	uucp_path	64
eco_i	14	login	31	remote_job	48	vmnet	65
ecr_i	15	mtp	32	rje	49	whois	66
efs	16	name	33	shell	50		
exec	17	netbios_dgm	34	smtp	51		

Çizelge 2.22'de KDD100 veri setindeki "service" adlarının sayısal formata dönüştürülmesi gösterilmiştir.

Çizelge 2.22. KDD100 veri setindeki "service" adlarının sayısal formata dönüştürülmesi

Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri	Service Adı	Sayısal Değeri
IRC	1	finger	19	name	37	smtp	55
X11	2	ftp	20	netbios_dgm	38	sql_net	56
Z39_50	3	ftp_data	21	netbios_ns	39	ssh	57
aol	4	gopher	22	netbios_ssn	40	sunrpc	58
auth	5	harvest	23	netstat	41	supdup	59
bgp	6	hostnames	24	nnsp	42	systat	60
courier	7	http	25	nntp	43	telnet	61
csnet_ns	8	http_2784	26	ntp_u	44	tftp_u	62
ctf	9	http_443	27	other	45	tim_i	63
daytime	10	http_8001	28	pm_dump	46	time	64
discard	11	imap4	29	pop_2	47	urh_i	65
domain	12	iso_tsap	30	pop_3	48	urp_i	66
domain_u	13	klogin	31	printer	49	uucp	67
echo	14	kshell	32	private	50	uucp_path	68
eco_i	15	ldap	33	red_i	51	vmnet	69
ecr_i	16	link	34	remote_job	52	whois	70
efs	17	login	35	rje	53		
exec	18	mtp	36	shell	54		

KDD100 veri setinde metin alanlar sayısal formata dönüştürüldükten sonra elde edilen giriş ve çıkış değerleri Çizelge 2.23'te gösterilmiştir.

Çizelge 2.23. Örnek KDD100 veri setinin sayısal format halindeki giriş ve çıkış değerleri

	Özellik Adı	**Örnek 1**	**Örnek 2**	**Örnek 3**	**Örnek 4**
Girişler (41 Özellik)	duration	0	0	0	0
	protocol_type	2	1	2	2
	service	25	16	50	21
	flag	10	10	6	10
	src_bytes	181	1032	0	0
	dst_bytes	5450	0	0	848
	land	0	0	0	0
	wrong_fragment	0	0	0	0
	urgent	0	0	0	0
	hot	0	0	0	0
	num_failed_logins	0	0	0	0
	logged_in	1	0	0	0
	num_compromised	0	0	0	0
	root_shell	0	0	0	0
	su_attempted	0	0	0	0
	num_root	0	0	0	0
	num_file_creations	0	0	0	0
	num_shells	0	0	0	0
	num_access_files	0	0	0	0
	num_outbound_cmds	0	0	0	0
	is_host_login	0	0	0	0
	is_guest_login	0	0	0	0
	count	8	511	63	2
	srv_count	8	511	1	2
	serror_rate	0	0	1	0
	srv_serror_rate	0	0	1	0
	rerror_rate	0	0	0	0
	srv_rerror_rate	0	0	0	0
	same_srv_rate	1	1	0.02	1
	diff_srv_rate	0	0	0.08	0
	srv_diff_host_rate	0	0	0	0
	dst_host_count	9	255	1	2
	dst_host_srv_count	9	255	1	2
	dst_host_same_srv_rate	1	1	1	1
	dst_host_diff_srv_rate	0	0	0	0
	dst_host_same_src_port_rate	0.01	1	1	1
	dst_host_srv_diff_host_rate	0	0	0	0
	dst_host_serror_rate	0	0	1	0
	dst_host_srv_serror_rate	0	0	1	0
	dst_host_rerror_rate	0	0	0	0
	dst_host_srv_rerror_rate	0	0	0	0
Çıkış (1 Özellik)	attack_type (Saldırı Tipleri)	12	19	10	22

2.12. Minimum Fazlalık Maksimum İlişkili (mRMR) Özellik Seçimi

mRMR, bir veri setinde bulunan özellikler arasında en ilişkili özellikleri seçmek ve gereksizliği azaltmaya yarayan bir yöntemdir [361–368]. mRMR, entropi (düzensizlik) tabanlı özellik seçme yöntemidir. Entropi, rastgele bir özellikteki belirsizliği hesaplamaktadır. Entropi, 0-1 arasında değer üretmektedir. Entropi hesaplaması denklem (2.70)'de gösterilmiştir [366, 369–372].

n toplam özellik sayısı olsun; belirli özellikler için $X_i(i \in \{1,2,\dots,n\})$ olsun.

$$H(X) = -\sum_{i=1}^{n} p(X_i)\log\big(p(X_i)\big) \tag{2.70}$$

Burada; X ayrık verilerden oluşan rastgele bir özellik , X_i bu özellikteki farklı verileri ve $p(X_i)$ ise farklı verilerin olasılık fonksiyonudur. $H(X)$, X'nin entropisi olarak ifade edilir.

mRMR, denklem (2.71)'deki gibi hesaplanır [370, 373].

$$f^{mRMR}(X_i) = I(C, X_i) - \frac{1}{|S|}\sum_{X_S \in S} I(X_S, X_i) \tag{2.71}$$

Burada; C, yanıt değişkeni (sınıf etiketi), S seçilen özellikler kümesi ve $|S|$ ise bu özellik kümesinin boyutu (özellik sayısı) olarak ifade edilir. $X_S \in S$ ise seçilen özellikler kümesinden bir özellik olarak ifade edilir. Eğer $X_i \notin S$ ise seçilen özellikler kümesinden henüz seçilmemiş bir özellik olarak ifade edilir. Ortaklık bilgisi (mutual information), bir özellik kümesi arasında doğrusal ve doğrusal olmayan bağımlılığın bir ölçüsüdür [374]. Diğer bir deyişle; ortaklık bilgisi, rastgele özellikler arasındaki bağımlılığı ölçmek için kullanılmaktadır [93]. Ortaklık bilgisi, iki özelliğin paylaştığı bilgi kalitesini (infomation quantity) ölçmeyi amaçlar [375]. mRMR, önemli ve benzer özellikleri elde etmek için ortaklık bilgisi kullanır [364]. Diğer bir deyişle, artıklık (redundancy) ve ilişki (relevance) hesaplamak için ortaklık bilgisi kullanılır [376, 377].

X ve Y rastgele iki ayrık özellik olmak üzere; bu özellikler arasındaki bilginin ölçümü ortaklık bilgisi kullanarak denklem (2.72)'deki gibi hesaplanmaktadır [362, 364, 369, 373, 378–384]. Ortaklık bilgisi $I(,.,)$, olarak ifade edilir.

$$I(Y,X) = \sum_{y\in\Omega_Y}\sum_{x\in\Omega_X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2.72}$$

Denklem (2.72)'deki $p(x,y)$, ortak olasılık yoğunluk fonksiyonunu, $p(x)$ ve $p(y)$ ise marjinal yoğunluk fonksiyonlarıdır. Ω_Y ve Ω_X , X ve Y'ye karşılık gelen örnek uzaylarıdır [373].

Eğer X ve Y rastgele iki sürekli özelliklerden oluştuğu kabul edilirse; bu iki özellik arasındaki ortaklık bilgisi denklem (2.73)'teki gibi hesaplanır [367, 369, 373, 374, 376–378, 385, 386].

$$I(Y,X) = \iint_{\Omega_Y\Omega_X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dxdy \tag{2.73}$$

En iyi kümeyi elde etmek için minimum artıklık (minimum redundancy) ve maksimum ilişki (maximum relevance) koşullarının sağlanması gerekmektedir [380]. Ayrık özellikler için; minimum artıklık koşulu denklem (2.74)'teki gibi hesaplanmaktadır [361, 362, 367, 374, 379–383, 386–388].

$$\min A(S), A = \frac{1}{|S|^2}\sum_{X_i,X_j\in S} I(X_i,X_j) \tag{2.74}$$

Ayrık özellikler için; maksimum ilişki koşulu ise denklem (2.75)'teki gibi hesaplanmaktadır [361, 362, 367, 374, 379–383, 386–388].

$$\max B(S,C), B = \frac{1}{|S|}\sum_{X_i,\in S} I(X_i;C) \tag{2.75}$$

Bir veri seti üzerinde mRMR çalıştırıldığında ortak bilgi farkı (mutual information difference-mid) ve ortak bilgi oranı (mutual information quotient-miq) olmak üzere iki yöntem kullanılır [361–363, 370, 373, 380].

Ayrık özellikler için;

Ortak bilgi farkı denklem (2.76)'daki gibi hesaplanır [373].

$$f^{mRMR\_mid}(X_i) = I(C, X_i) - \frac{1}{|S|}\sum_{X_S \in S} I(X_S, X_i) \quad (2.76)$$

C , yanıt değişkeni, S seçilen özellikler kümesi ve $|S|$ ise bu özellik kümesinin boyutu olarak ifade edilir. $X_S \in S$ ise seçilen özellikler kümesinden bir özellik olarak ifade edilir. Eğer $X_i \notin S$ ise seçilen özellikler kümesinden henüz seçilmemiş bir özellik olarak ifade edilir [362, 370, 373, 381, 384, 387, 389, 390]. Ortak bilgi oranı denklem (2.77)'deki gibi hesaplanır [373]

$$f^{mRMR\_miq}(X_i) = I(C, X_i) / \frac{1}{|S|}\sum_{X_S \in S} I(X_S, X_i) \quad (2.77)$$

C , yanıt değişkeni, S seçilen özellikler kümesi ve $|S|$ ise bu özellik kümesinin boyutu olarak ifade edilir. $X_S \in S$ ise seçilen özellikler kümesinden bir özellik olarak ifade edilir. Eğer $X_i \notin S$ ise seçilen özellikler kümesinden henüz seçilmemiş bir özellik olarak ifade edilir [362, 370, 373, 381, 384, 387, 389, 390].

2.13. KDD10 Veri Setine mRMR Yönteminin Uygulanması

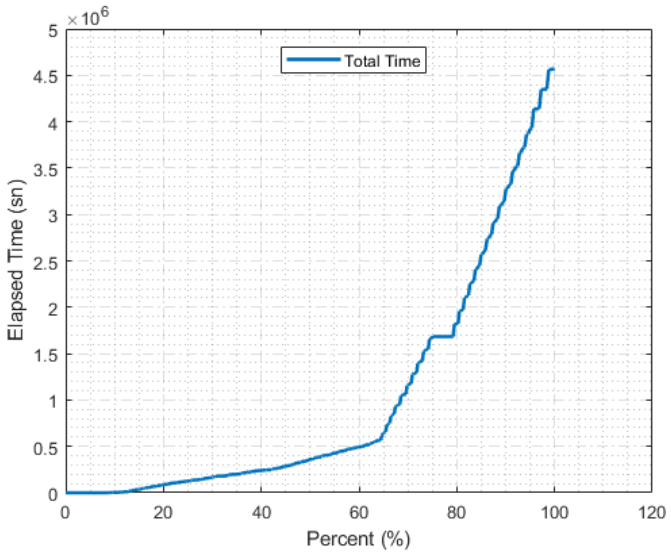
Çizelge 2.24'te de görüldüğü gibi KDD10 veri seti üzerine mRMR yöntemi uygulanmış olup, veri seti içerisinde de bulunan özellikler önem derecesine göre sıralanmıştır. Sıralanan bu özelliklere göre, copula fonksiyonları kullanılarak sınıflandırma işlemi yapılmıştır.

Çizelge 2.24. KDD10 veri setine mRMR_miq kriteri uygulandığında özelliklerin önem derecesine göre sıralanması

Özellik Adı	Özellik No	Özellik Adı	mRMR_miq Kriteri
duration	1	count	23
protocol_type	2	dst_bytes	6
service	3	duration	1
flag	4	dst_host_count	32
src_bytes	5	src_bytes	5
dst_bytes	6	srv_count	24
land	7	dst_host_srv_count	33
wrong_fragment	8	flag	4
urgent	9	service	3
hot	10	protocol_type	2
num_failed_logins	11	land	7
logged_in	12	wrong_fragment	8
num_compromised	13	urgent	9
root_shell	14	hot	10
su_attempted	15	num_failed_logins	11
num_root	16	logged_in	12
num_file_creations	17	num_compromised	13
num_shells	18	root_shell	14
num_access_files	19	su_attempted	15
num_outbound_cmds	20	num_root	16
is_host_login	21	num_file_creations	17
is_guest_login	22	num_shells	18
count	23	num_access_files	19
srv_count	24	num_outbound_cmds	20
serror_rate	25	is_host_login	21
srv_serror_rate	26	is_guest_login	22
rerror_rate	27	serror_rate	25
srv_rerror_rate	28	srv_serror_rate	26
same_srv_rate	29	rerror_rate	27
diff_srv_rate	30	srv_rerror_rate	28
srv_diff_host_rate	31	same_srv_rate	29
dst_host_count	32	diff_srv_rate	30
dst_host_srv_count	33	srv_diff_host_rate	31
dst_host_same_srv_rate	34	dst_host_same_srv_rate	34
dst_host_diff_srv_rate	35	dst_host_diff_srv_rate	35
dst_host_same_src_port_rate	36	dst_host_same_src_port_rate	36
dst_host_srv_diff_host_rate	37	dst_host_srv_diff_host_rate	37
dst_host_serror_rate	38	dst_host_serror_rate	38
dst_host_srv_serror_rate	39	dst_host_srv_serror_rate	39
dst_host_rerror_rate	40	dst_host_rerror_rate	40
dst_host_srv_rerror_rate	41	dst_host_srv_rerror_rate	41

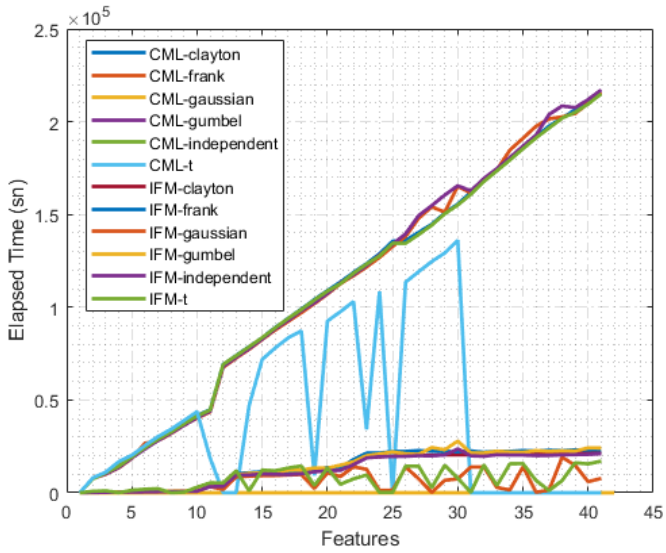
2.14. Özellik Seçimi

Çizelge 2.24'te görüldüğü üzere KDD10 veri seti üzerinde mRMR_miq özellik seçme kriteri kullanılarak özellikler önem sırasına göre sıralandıktan sonra ilk (23.) özellik baz alınarak özellik seçimine başlanılmıştır. İlk özellik alınıp bu özelliliğe göre diğer özelliklerin ilişki durumuna bakılmıştır. Örneğin, ilk başta 1.ci özellik (23.) seçilmiştir. Daha sonra 1.ci özellik ile 2. özellik (6.) alınarak sınıflandırma işleminde doğruluk oranı hesaplanmıştır. Bundan sonraki her bir özellik eklendiğinde bu yöntem kullanılarak doğruluk oranları elde edilmiştir. Bu durum son özelliğe (41.özelliğe) kadar devam etmiştir. Her bir veri seti için en iyi üç başarım oranı ve kullanılan özellikler elde edilerek sınıflandırma işlemi tamamlanmıştır. Şekil 2.15'te KDD10 veri seti üzerinde özellik seçimi yapılırken toplam geçen süre (elapsed time) ve hesaplanan yüzdelik (percent) arasındaki ilişki gösterilmiştir.



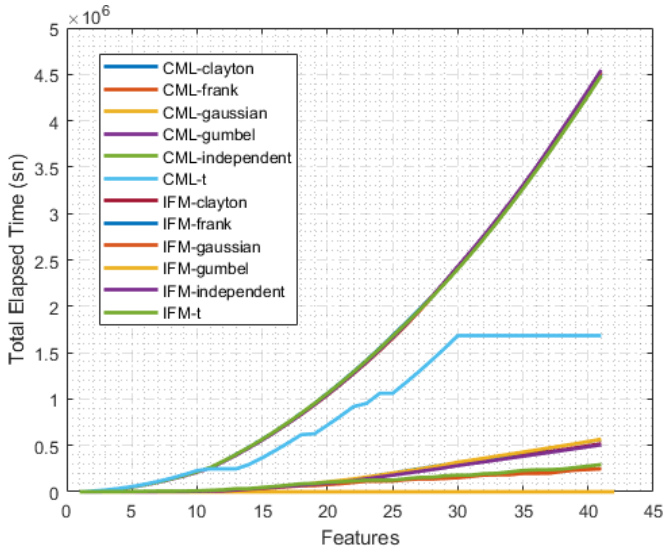
Şekil 2.15. KDD10 veri seti üzerinde özellik seçimi yapılırken toplam geçen süre'ye bağlı olarak toplam hesaplanan yüzdelik arasındaki ilişki

Şekil 2.15'te de görüldüğü üzere, KDD10 veri seti üzeride sınıflandırma yapılırken kullanılan bütün özellikler için geçen toplam süre hesaplanmıştır. Hesaplama yapılırken yukarıda da ifade edildiği gibi özellikler ayrı ayrı eklenerek doğruluk oranları hesaplanmıştır. Eklenen özellik sayısı artıkça buna bağlı olarak hesaplama süreside artmaktadır. Özellikle, hesaplanması yapılan özelliklerin yüzdelik oranı % 80'den sonra daha fazla zaman harcadığı anlaşılmaktadır. KDD10 veri seti üzerinde her bir copula ailesi ile IFM/CML metotlarının kullanımına göre iki özellik arasında geçen süre hesaplanması Şekil 2.16'da gösterilmiştir.



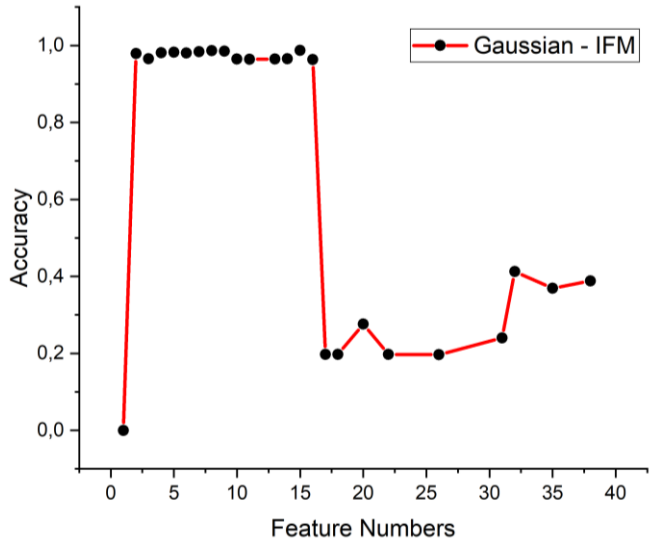
Şekil 2.16. KDD10 veri seti üzerinde her bir copula ailesi ile IFM/CML metotlarının kullanımına göre iki özellik arasında geçen süre hesaplaması

Şekil 2.16'da da görüldüğü üzere, KDD10 veri seti üzeride sınıflandırma yapılırken kullanılan her adımda iki özellik arasındaki geçen süre hesaplanmıştır. Hesaplama yapılırken Şekil 2.16'da da ifade edildiği gibi özellikler ayrı ayrı eklenerek doğruluk oranları hesaplanmıştır. KDD10 veri seti üzerinde her bir copula ailesi ile IFM/CML metotlarının kullanımına göre özellikler (features)'in toplam geçen süre (total elapsed time)'ye göre değişimi Şekil 2.17'de gösterilmiştir.



Şekil 2.17. KDD10 veri seti üzerinde her bir copula ailesi ile IFM/CML metotlarının kullanımına göre özelliklerin toplam geçen süre'ye göre değişimi

Şekil 2.17'de de görüldüğü üzere, KDD10 veri seti üzerinde sınıflandırma yapılırken yeni özellik eklendikçe geçen süre daha önce hesaplanan sürelere eklenerek her bir copula ailesi için toplam geçen süre hesaplanmıştır. KDD10 veri seti üzerinde bütün özellikler kullanıldığında en iyi performansı elde eden copula ailesi IFM metodunu kullanarak gaussian copulası olmuştur. KDD10 veri setinin 41 özelliği için en iyi performansları elde eden gaussian copula ailesinin başarım oranları Şekil 2.18'de gösterilmiştir.



Şekil 2.18. KDD10 veri setinin 41 özelliği için en iyi performansı gösteren gaussian copula ailesinin başarım oranları

Şekil 2.18'de de görüldüğü üzere, KDD10 veri seti üzerinde gaussian copulası ve IFM metodu özellik sayısına bağlı olarak elde edilen doğruluk (accuracy) oranları hesaplanmıştır. Gaussian copulasının doğruluk oranlarına bakıldığında en iyi başarım oranlarını 2.özellik ile 15.ci özellik arasında elde edildiği gözlemlenmiştir.

3. BULGULAR VE TARTIŞMA

3.1. Makine Öğrenme Metotlarını Kullanarak Saldırı Tespiti

3.1.1. Uygulama 1: KDD10, KDDTEST

Uygulama 1'de; KA, TÖ ve DVM olmak üzere üç adet sınıflandırma algoritması kullanılmıştır. KA sınıflandırma algoritmasından; fine tree, medium tree ve coarse tree olmak üzere üç adet sınıflandırıcı kullanılmıştır. TÖ sınıflandırma algoritmasından; boosted trees, bagged trees ve rusboosted trees olmak üzere üç adet sınıflandırıcı kullanılmıştır. DVM sınıflandırma algoritmasından ise; linear DVM, quadratic DVM, cubic DVM, fine gaussian DVM, medium gaussian DVM ve coarse gaussian DVM olmak üzere altı adet sınıflandırıcı kullanılmıştır. Bu on iki farklı sınıflandırma algoritması kullanılarak KDD10, KDDTEST ve KDD10+KDDTEST veri setleri üzerinde, her bir sınıflandırıcının başarım değerleri elde edilmiştir. Bu başarım değerleri elde edilirken TBA'nin aktif ve pasif durumuna göre başarım oranları elde edilmiştir. Tüm eğitimler MATLAB programının Classification Learner Toolbox (CLT)'ı kullanılarak yapılmıştır. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmıştır.

Etiketli olan KDD10 ve KDDTEST veri setleri 42 özellikten oluşmaktadır. Bu uygulama için; Çizelge 3.1'de gösterilen KDD10 ve KDDTEST veri setlerinin giriş ve çıkış değerleri kullanılarak başarım oranları elde edilmiştir. Veri setlerinin eğitim ve test işlemleri sırasında herhangi bir ön işlem yapılmamıştır. Eğitim ve test aşamasında CLT'ye veri setlerinin ilk 41 özelliği giriş olarak ve son özellik ise çıkış olarak verilmiştir. KDD10 ve KDDTEST veri setleri, 4 çekirdekli Intel Core i5-4590S işlemci, 4 GB Ram ve Intel HD Graphics 4600 bilgisayarda eğitilmiştir. KDD10+KDDTEST veri seti ise veri miktarının fazla olmasından dolayı Intel Xeon E5620 (2 işlemci 8 çekirdek), 16 GB Ram ve NVIDIA Quadro K2000 ekran kartı olan iş istasyonunda eğitilmiştir.

Çizelge 3.1. Örnek KDD10 ve KDDTEST veri setlerinin giriş ve çıkış özellikleri

	Özellik Adı	**Örnek 1**	**Örnek 2**	**Örnek 3**	**Örnek 4**
Girişler (41 Özellik)	duration	0	0	0	289
	protocol_type	tcp	icmp	tcp	tcp
	service	http	ecr_i	private	ftp
	flag	SF	SF	REJ	SF
	src_bytes	181	1032	0	157
	dst_bytes	5450	0	0	595
	land	0	0	0	0
	wrong_fragment	0	0	0	0
	urgent	0	0	0	0
	hot	0	0	0	2
	num_failed_logins	0	0	0	0
	logged_in	1	0	0	1
	num_compromised	0	0	0	0
	root_shell	0	0	0	0
	su_attempted	0	0	0	0
	num_root	0	0	0	0
	num_file_creations	0	0	0	0
	num_shells	0	0	0	0
	num_access_files	0	0	0	0
	num_outbound_cmds	0	0	0	0
	is_host_login	0	0	0	0
	is_guest_login	0	0	0	1
	count	8	511	106	1
	srv_count	8	511	15	1
	serror_rate	0	0	0	0
	srv_serror_rate	0	0	0	0
	rerror_rate	0	0	1	0
	srv_rerror_rate	0	0	1	0
	same_srv_rate	1	1	0.14	1
	diff_srv_rate	0	0	0.07	0
	srv_diff_host_rate	0	0	0	0
	dst_host_count	9	255	255	18
	dst_host_srv_count	9	255	15	15
	dst_host_same_srv_rate	1	1	0.06	0.83
	dst_host_diff_srv_rate	0	0	0.05	0.11
	dst_host_same_src_port_rate	0.01	1	0	0.06
	dst_host_srv_diff_host_rate	0	0	0	0
	dst_host_serror_rate	0	0	0	0
	dst_host_srv_serror_rate	0	0	0	0
	dst_host_rerror_rate	0	0	1	0
	dst_host_srv_rerror_rate	0	0	1	0
Çıkış (1 Özellik)	attack_type (Saldırı Tipleri)	normal	smurf	neptune	warezmaster

KDD10 veri seti 494021 örnekten, KDDTEST veri seti 311029 örnekten ve KDD10+KDDTEST veri seti ise 805050 örnekten oluşmaktadır. Çizelge 3.2'de KDD10, KDDTEST ve KDD10+KDDTEST veri setlerinde bulunan normal ve saldırı tiplerinin miktarları ve yüzdelik oranları verilmiştir.

Çizelge 3.2. KDD10, KDDTEST ve KDD10+KDDTEST veri setlerinde bulunan normal ve saldırı tiplerinin miktarları ve yüzdelik oranları

	KDD10		KDDTEST		KDD10+KDDTEST	
Saldırı Tipi	**Miktarı**	**Yüzdelik Oranı (%)**	**Miktarı**	**Yüzdelik Oranı (%)**	**Miktarı**	**Yüzdelik Oranı (%)**
apache2	-	-	794	0.2552	794	0.0986
back	2203	0.4459	1098	0.3530	3301	0.4100
buffer_overflow	30	0.0060	22	0.0070	52	0.0064
ftp_write	8	0.0016	3	0.0009	11	0.0013
guess_passwd	53	0.0107	4367	1.4040	4420	0.5490
httptunnel	-	-	158	0.0507	158	0.0196
imap	12	0.0024	1	0.0003	13	0.0016
ipsweep	1247	0.2524	306	0.0983	1553	0.1929
land	21	0.0042	9	0.0028	30	0.0037
loadmodule	9	0.0018	2	0.0006	11	0.0013
mailbomb	-	-	5000	1.6075	5000	0.6210
mscan	-	-	1053	0.3385	1053	0.1307
multihop	7	0.0014	18	0.0057	25	0.0031
named	-	-	17	0.0054	17	0.0021
neptune	107201	21.6996	58001	18.6481	165202	20.5207
nmap	231	0.0467	84	0.0270	315	0.0391
normal	97278	19.6910	60593	19.4814	157871	19.6100
perl	3	0.0006	2	0.0006	5	0.0006
phf	4	0.0008	2	0.0006	6	0.0007
pod	264	0.0534	87	0.0279	351	0.0435
portsweep	1040	0.2105	354	0.1138	1394	0.1731
processtable	-	-	759	0.2440	759	0.0942
ps	-	-	16	0.0051	16	0.0019
rootkit	10	0.0020	13	0.0041	23	0.0028
saint	-	-	736	0.2366	736	0.0914
satan	1589	0.3216	1633	0.5250	3222	0.4002
sendmail	-	-	17	0.0054	17	0.0021
smurf	280790	56.8376	164091	52.7574	444881	55.2612
snmpgetattack	-	-	7741	2.4888	7.741	0.9615
snmpguess	-	-	2406	0.7735	2406	0.2988
spy	2	0.0004	-	-	2	0.0002
sqlattack	-	-	2	0.0006	2	0.0002
teardrop	979	0.1981	12	0.0038	991	0.1230
udpstorm	-	-	2	0.0006	2	0.0002
warezclient	1020	0.2064	-	-	1020	0.1267
warezmaster	20	0.0040	1602	0.5150	1622	0.2014
worm	-	-	2	0.0006	2	0.0002
xlock	-	-	9	0.0028	9	0.0011
xsnoop	-	-	4	0.0012	4	0.0004
xterm	-	-	13	0.0041	13	0.0016
Toplam	**494021**	**100**	**311029**	**100**	**805050**	**100**

Kullanmış olduğumuz her bir sınıflandırma yönteminin varsayılan özellikleri kullanılmıştır. Bu uygulamada kullanılan TÖ sınıflandırıcıların varsayılan özellikleri Çizelge 3.3'te, DVM sınıflandırıcıların varsayılan özellikleri Çizelge 3.4'te ve KA sınıflandırıcıların varsayılan özellikleri Çizelge 3.5'te gösterilmiştir.

Çizelge 3.3. TÖ sınıflandırıcıların varsayılan özellikleri

Main Technique	Used Technique	Ensemble Method	PCA	Learner Type	Maximum Numbers of Splits	Number of Learners	Learning Rate
Ensemble	Boosted Trees	Adaboost	Off/On	Decision Tree	20	30	0.1
Ensemble	Bagged Trees	Bag	Off/On	Decision Tree	-	30	-
Ensemble	RUSBoosted Trees	RUSBoost	Off/On	Decision Tree	20	30	0.1

Çizelge 3.4. DVM sınıflandırıcıların varsayılan özellikleri

Main Technique	Used Technique	PCA	Kernel Function	Kernel Scale	Box Constraint Level	MultiClass Method	Standardize Data
SVM	Linear SVM	Off/On	Linear	Automatic	1	One-vs-One	True
SVM	Quadratic SVM	Off/On	Quadratic	Automatic	1	One-vs-One	True
SVM	Cubic SVM	Off/On	Cubic	Automatic	1	One-vs-One	True
SVM	Fine Gaussian SVM	Off/On	Gaussian	1.6	1	One-vs-One	True
SVM	Medium Gaussian SVM	Off/On	Gaussian	6.4	1	One-vs-One	True
SVM	Coarse Gaussian SVM	Off/On	Gaussian	26	1	One-vs-One	True

Çizelge 3.5. KA sınıflandırıcıların varsayılan özellikleri

Main Technique	Used Technique	PCA	Maximum Numbers of Splits	Split Criterion	Surrogate Decision Splits
Decision Tree	Fine Tree	Off/On	100	Gini's Diversity Index	Off
Decision Tree	Medium Tree	Off/On	20	Gini's Diversity Index	Off
Decision Tree	Coarse Tree	Off/On	4	Gini's Diversity Index	Off

Tez çalışmasında kullanılan on iki adet sınıflandırıcının değerlendirme metrikleri Çizelge 3.6'da hata matrisi kullanılarak elde edilmiştir.

Çizelge 3.6. Hata matrisi

Hata Matrisi (Confusion Matrix)		**Tahmin Edilen Sınıf (Predicted Class)**	
		Positive	Negative
Gerçek Sınıf (Actual Class)	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

True Positive (TP): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde de pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.

False Negative (FN): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir.

False Positive(FP): Gerçekte negatif (normal) olan ve tahmin edildiğinde pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.

True Negative (TN): Gerçekte negatif (normal) olan ve tahmin edildiğinde de negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir [13, 18, 23, 342, 344, 376, 377]. Denklem (3.78)'de bu tez çalışmasında kullanılan başarım ölçütü ve denklem (3.79)'da ise liteartürde çokça rastlanılan diğer başarım ölçütleri verilmiştir [13, 18, 23, 342, 356, 376, 377, 385].

$$Doğruluk\ (Accuracy) = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.78}$$

$$Kesinlik\ (Precision) = \frac{TP}{TP + FP}$$

$$Duyarlılık\ (Sensitivity) = \frac{TP}{TP + FN}$$

$$Özgünlük\ (Specificity) = \frac{TN}{TN + FP} \tag{3.79}$$

$$Negatif\ Tahmin\ Değeri\ (Negative\ Predictive\ Value) = \frac{TN}{TN + FN}$$

Çizelge 3.7'de 12 adet sınıflandırıcının KDD10 veri seti üzerindeki performansı, Çizelge 3.8'de 12 adet sınıflandırıcının KDDTEST veri seti üzerindeki performansı ve Çizelge 3.9'da ise 12 adet sınıflandırıcının KDD10+KDDTEST veri seti üzerindeki performansı gösterilmiştir.

Çizelge 3.7. 12 adet sınıflandırıcının KDD10 veri seti üzerindeki performansı

Sınıflandırıcılar	**TBA Pasif**		**TBA Aktif**	
	Doğruluk (%)	**Eğitim ve Test Süresi (Sn)**	**Doğruluk (%)**	**Eğitim Süresi ve Test (Sn)**
Fine Tree	99.90	832	99.90	763
Medium Tree	99.50	10290	99.50	745
Coarse Tree	98.30	895	98.20	688
Boosted Trees	99.80	637	99.60	599
Bagged Trees	**99.99**	**447**	**99.90**	**369**
RUSBoosted Trees	91.80	400	84.70	360
Linear SVM	99.90	2769	89.60	12228
Quadratic SVM	99.90	6920	82.20	65404
Cubic SVM	86.30	62429	86.80	130300
Fine Gaussian SVM	99.80	15394	99.40	1784
Medium Gaussian SVM	99.90	4178	99.40	1807
Coarse Gaussian SVM	99.90	2918	99.20	2629

Yapılan uygulama sonucunda; Çizelge 3.7'de göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %99.99 ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %86.30 ile Cubic SVM sınıflandırıcısı elde etmiştir Öte yandan, TBA aktif durumda iken en iyi performansları TÖ sınıflandırıcısından olan %99.90 ile Bagged Trees sınıflandırıcısı ile KA sınıflandırıcısından olan %99.90 ile Fine Tree elde etmiştir. Bagged Trees sınıflandırıcısı ile Fine Tree sınıflandırıcısı aynı başarım oranına sahip olmasına rağmen, eğitim ve test süresi bakımından en iyi sonucu 369 saniye ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %82.20 ile Quadratic SVM sınıflandırıcısı elde etmiştir.

Çizelge 3.8. 12 adet sınıflandırıcının KDDTEST veri seti üzerindeki performansı

Sınıflandırıcılar	TBA Pasif		TBA Aktif	
	Doğruluk (%)	Eğitim ve Test Süresi (Sn)	Doğruluk (%)	Eğitim ve Test Süresi (Sn))
Fine Tree	97.30	23	96.80	31
Medium Tree	95.10	14	94.90	23
Coarse Tree	90.90	11	90.90	23
Boosted Trees	96.50	600	95.80	481
Bagged Trees	**97.90**	**297**	**97.60**	**237**
RUSBoosted Trees	73.00	377	55.20	368
Linear SVM	96.30	9508	95.60	14826
Quadratic SVM	96.90	16756	77.00	25586
Cubic SVM	78.90	125060	71.10	74122
Fine Gaussian SVM	96.60	15306	95.80	5309
Medium Gaussian SVM	96.30	13022	95.70	5520
Coarse Gaussian SVM	96.30	10493	95.30	8058

Yapılan uygulama sonucunda; Çizelge 3.8'de göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %97.90 ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise TÖ öğrenme sınıflandırıcısından olan %73.00 ile RUSBoosted Trees sınıflandırıcısı elde etmiştir. Öte yandan, TBA aktif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %97.60 ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %71.10 ile Cubic SVM sınıflandırıcısı elde etmiştir

Çizelge 3.9. 12 adet sınıflandırıcının KDD10+KDDTEST veri seti üzerindeki performansı

Sınıflandırıcılar	TBA Pasif		TBA Aktif	
	Doğruluk (%)	Eğitim ve Test Süresi (Sn)	Doğruluk (%)	Eğitim ve Test Süresi (Sn)
Fine Tree	98.40	63	98.30	101
Medium Tree	97.00	44	96.80	62
Coarse Tree	95.20	38	94.90	76
Boosted Trees	98.20	2081	98.10	1612
Bagged Trees	**100.00**	**1192**	**98.80**	**952**
RUSBoosted Trees	80.90	1299	65.00	1212
Linear SVM	98.50	11720	81.10	57348
Quadratic SVM	98.50	89777	76.80	273860
Cubic SVM	75.60	668140	72.60	486570
Fine Gaussian SVM	98.60	75506	97.30	8523
Medium Gaussian SVM	98.50	20729	97.20	8861
Coarse Gaussian SVM	98.50	13795	97.00	13070

Yapılan uygulama sonucunda; Çizelge 3.9'da göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %100.00 ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %75.60 ile Cubic SVM sınıflandırıcısı elde etmiştir. Öte yandan, TBA aktif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %98.80 ile Bagged Trees sınıflandırıcısı elde etmiştir. En kötü performansı ise TÖ sınıflandırıcısından olan %65.00 ile RUSBoosted Trees sınıflandırıcısı elde etmiştir.

Çizelge 3.7, Çizelge 3.8 ve Çizelge 3.9 incelendiğinde TÖ sınıflandırıcılarından olan Bagged Trees sınıflandırıcısı en iyi performansı elde etmiştir. Bunun sebebi, TÖ sınıflandırıcıların birden fazla algoritmayı birleştirilerek başarım oranını mümkün olduğunca yükseltilmesi olarak ifade edilebilir.

3.1.2. Uygulama 2: KDD100

Uygulama 2'de; KDD100 veri seti üzerinde on iki adet sınıflandırıcı kullanılarak, her bir sınıflandırıcının performans değerleri elde edilmiştir. Eğitimler MATLAB programının CLT'ı kullanılarak yapılmıştır. Bu uygulamada; Çizelge 3.1'de gösterilen KDD10 ve KDDTEST veri setlerinin giriş ve çıkış değerleri kullanılarak başarım oranları elde edilmiştir.

Eğitimler için 10 çekirdekli 2 adet Intel Xeon(R) CPU E52687Wv3@ 3.10 GHz işlemcisi, 64 GB Ram ve Nvidia Quadro P5000 GPU'su olan bir HP-Z840 iş istasyonu kullanılmıştır. Kullanılan her bir sınıflandırma yönteminin varsayılan özellikleri kullanılmıştır. Bu uygulamada kullanılan TÖ sınıflandırıcıların varsayılan özellikleri Çizelge 3.3'te, DVM sınıflandırıcıların varsayılan özellikleri Çizelge 3.4'te ve KA sınıflandırıcıların varsayılan özellikleri Çizelge 3.5'te gösterilmiştir. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmıştır. Sınıflandırıcıların değerlendirme metrikleri Çizelge 3.6'da hata matrisi kullanılarak elde edilmiştir. Çizelge 3.10'da KDD100 veri setinde bulunan saldırı tiplerinin miktarları ve yüzdelik oranları gösterilmiştir.

Çizelge 3.10. KDD100 veri setinde bulunan saldırı tiplerinin miktarları ve yüzdelik oranları

Saldırı Tipi	Miktarı	Yüzdelik Oranı (%)
back	2203	0.04500
buffer_overflow	30	0.00061
ftp_write	8	0.00016
guess_passwd	53	0.00110
imap	12	0.00024
ipsweep	12481	0.25480
land	21	0.00042
loadmodule	9	0.00018
multihop	7	0.00014
neptune	1072017	21.88490
nmap	2316	0.04730
normal	972781	19.8590
perl	3	0.00006
phf	4	0.00008
pod	264	0.00540
portsweep	10413	0.21260
rootkit	10	0.00020
satan	15892	0.32440
smurf	2807886	57.32220
spy	2	0.00004
teardrop	979	0.02000
warezclient	1020	0.02080
warezmaster	20	0.00040
Toplam	**4898431**	**100**

Çizelge 3.11'de 12 adet sınıflandırıcının KDD100 veri seti üzerindeki performansı gösterilmiştir.

Çizelge 3.11. 12 adet sınıflandırıcının KDD100 veri seti üzerindeki performansı

Sınıflandırıcılar	Doğruluk (%)	Eğitim ve Test Süresi (sn)
Coarse Tree	99.10	284
Medium Tree	99.80	324
Fine Tree	99.90	447
RUSBoosted Trees	92.20	2808
Boosted Trees	99.90	9996
Bagged Trees	**100.00**	**12947**
Linear SVM	100.00	36254
Coarse Gaussian SVM	100.00	60641
Medium Gaussian SVM	100.00	166390
Fine Gaussian SVM	100.00	734420
Quadratic SVM	99.90	820600
Cubic SVM	59.90	2378700

Yapılan uygulama sonucunda; Çizelge 3.11'de de göründüğü gibi, 12 adet sınıflandırıcı arasında TÖ sınıflandırıcılarından olan Bagged Trees ile DVM sınıflandırıcılarından olan Linear SVM, Fine Gaussian SVM, Medium Gaussian SVM ve Coarse Gaussian SVM başarım oranları %100 olarak gerçekleşmiştir. Bu beş sınıflandırıcının başarım oranları %100 ile aynı olmasına rağmen, eğitim ve test süreleri bakımımdan en iyi performansı 12947 saniye ile Bagged Trees

sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcılarından olan %59.90 ile Cubic SVM sınıflandırıcısı elde etmiştir. DVM tabanlı sınıflandırıcılar çok daha iyi doğruluk oranlarını yakalamalarına karşın, test ve eğitim süreleri noktasında yüksek değerler ile geri kalmaktadırlar. Zira STS'ler için kısa sürelerde sınıflandırmanın yapılması oldukça önemlidir.

3.1.3. Uygulama 3: KDD10

Uygulama 3'te; KDD10 veri seti üzerinde YSA öğrenme yöntemi kullanılarak saldırı tespiti yapılmıştır. Eğitimler MATLAB programının Neural Net Fitting Toolbox'ı kullanılarak gerçekleştirilmiştir. Eğitimler için 4 çekirdekli Intel(R) Core(TM) i5-4590S CPU@ 3.0 GHz işlemci ve 4 GB Ram'e sahip olan bir HP-ProOne 600 markalı bilgisayar kullanılmıştır. Çizelge 3.12'de KDD10 veri seti üzerinde YSA öğrenme yöntemi uygulanarak elde edilen başarım oranları gösterilmiştir.

Çizelge 3.12. KDD10 veri seti üzerinde YSA öğrenme yöntemi uygulanarak elde edilen başarım oranları

Kullanılan Yöntem	Kullanılan Algoritma	Ara Katmandaki Nöron Sayısı	Toplam Veri Miktarı (%100)	Eğitim (Training) için Veri Miktarı (%70)	Doğrulama (Validation) için Veri Miktarı (%15)	Test (Testing) için Veri Miktarı (%15)	MSE	1-MSE (%)
YSA	Levenberg-Marquardt	5	494021	345815	74103	74103	0.0982	90.18
YSA	Levenberg-Marquardt	8	494021	345815	74103	74103	0.0526	94.74
YSA	Levenberg-Marquardt	10	494021	345815	74103	74103	0.0698	93.02
YSA	Levenberg-Marquardt	12	494021	345815	74103	74103	0.0595	94.05
YSA	Levenberg-Marquardt	15	494021	345815	74103	74103	0.0552	94.48
YSA	**Levenberg-Marquardt**	**20**	**494021**	**345815**	**74103**	**74103**	**0.0469**	**95.31**

Çizelge 3.12'de de görüldüğü üzere ara katmandaki nöron sayısına bağlı olarak YSA'nın başarım oranları elde edilmiştir. YSA'nın performansını ölçmek için ortalama kare hatası (MSE) başarım ölçme kriteri kullanılarak elde edilmiştir. MSE'den elde edilen başarım değeri 0 yakın olması performasın iyi olduğu anlamına gelmektedir. 1'e yakın olması durumda da ise performasın kötü olduğu anlamına gelmektedir. Performans kıyaslanması yapıldığında en iyi başarım oranı ara katmanda 20 adet nöron kullanılarak %95.31 olarak elde edilmiştir. En kötü başarım ise ara

katmanda 5 adet nöron kullanılarak %90.18 olarak elde edilmiştir. Çizelge 3.12'de de görüleceği üzere ara katmandaki nöron sayısı arttıkça genellikle başarımın arttığı görülmektedir.

3.2. Copula Fonksiyonlarını Kullanarak Saldırı Tespiti

Şekil 3.1'de KDD'99 veri seti kümesine copula-tabanlı sınıflandırıcıların uygulama aşamaları gösterilmiştir.

Şekil 3.1. KDD'99 veri seti kümesine copula-tabanlı sınıflandırıcıların uygulama aşamaları

3.2.1. Uygulama 4: KDD10

Uygulama 4'te; gumbel, independent, clayton, gaussian, student's-t ve frank copula aileleri ile CML, IFM metotları kullanılarak saldırı tespiti yapılmıştır. KDD10 veri setinin %1'lik, %5'lik, %10'luk ve %50'lik oranları Intel Xeon E5620 (2 işlemci 8 çekirdek), 16 GB Ram ve NVIDIA Quadro K2000 ekran kartı olan iş istasyonunda eğitilmiştir. KDD10 veri setinin %100'lük oranı ise veri miktarlarının fazla olmasından dolayı; 10 çekirdekli 2 adet Intel Xeon(R) CPU E52687Wv3@ 3.10 GHz işlemcisi, 64 GB Ram ve Nvidia Quadro P5000 GPU'su olan bir HP-Z840 iş istasyonunda eğitilmiştir. Eğitimler MATLAB ortamında yapılmıştır. Sınıflandırma aşamasında 10-kat çapraz doğrulama tekniği kullanılmıştır. Sınıflandırıcıların değerlendirme metrikleri için Çizelge 3.6'da hata matrisi kullanılarak elde edilmiştir. KDD10 veri setinde her bir normal ve saldırı tipinin belli yüzdelikleri alınmıştır. KDD10 veri setinin %1'lik kısmı normalde

4940 veriden oluşmaktadır. Ama bu çalışmada her bir saldırı tipinin %1 kısmı alınarak 4956 veri kullanılmıştır. Buradaki amaç; veri seti eğitilirken bütün saldırı tiplerinden örneklerin bulunmasını sağlamaktır. Eğer her bir saldırı tipinin aynı oranda yüzdelik dilimleri alınmasaydı bazı saldırı tipleri eğitim ve test işleminde kullanılamayacaktır. Bu durum, veri setinin %100, %50, %10 ve %5 içinde geçerlidir. Çizelge 3.13'te KDD10 veri setinde bulunan her bir saldırı tipinin veri setinin belirlenen %'lik dilimlerdeki miktarları gösterilmiştir.

Çizelge 3.13. KDD10 veri setinde bulunan her bir saldırı tipinin miktarları

Saldırı Tipi	**KDD10 (%100)**	**KDD10 (%50)**	**KDD10 (%10)**	**KDD10 (%5)**	**KDD10 (%1)**
back	2203	1102	221	111	23
buffer_overflow	30	15	3	2	1
ftp_write	8	4	1	1	1
guess_passwd	53	27	6	3	1
imap	12	6	2	1	1
ipsweep	1247	624	125	63	13
land	21	11	3	2	1
loadmodule	9	5	1	1	1
multihop	7	4	1	1	1
neptune	107201	53601	10721	5361	1073
nmap	231	116	24	12	3
normal	97278	48639	9728	4864	973
perl	3	2	1	1	1
phf	4	2	1	1	1
pod	264	132	27	14	3
portsweep	1040	520	104	52	11
rootkit	10	5	1	1	1
satan	1589	795	159	80	16
smurf	280790	140395	28079	14040	2808
spy	2	1	1	1	1
teardrop	979	490	98	49	10
warezclient	1020	510	102	51	11
warezmaster	20	10	2	1	1
Toplam	**494021**	**247016**	**49411**	**24713**	**4956**

KDD10 veri setinde bulunan 23 adet saldırı tipinin her birinin aynı yüzdelik oranları alınarak veri miktarı hesaplanmıştır. Örneğin; back saldırı tipinin veri miktarı KDD10 veri setinde 2203 örnekten oluşmaktadır. Back saldırı tipi için 2203 örneğin %1'lik kısmı 22.03 örnek olarak hesaplanmaktadır. Ondalık çıkan sayılar bir üst sayıya yuvarlanmıştır. Örneğin; back saldırı tipinden 22.03 örnek alınması gerekirken bir üst sayıya yuvarlandığından dolayı 23 örnek alınmıştır. Aynı durum diğer tüm saldırı tipi içinde geçerlidir. Çizelge 3.14'te KDD10 veri setinde bulunan her bir saldırı tipinin %1'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları gösterilmiştir.

Çizelge 3.14. KDD10 veri setinde bulunan her bir saldırı tipinin %1'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
gumbel	**IFM**	**973**	**3920**	**63**	**0**	**98.73**	**"23 6 1 32 5 24 33 4 3 2"**
independent	IFM	973	3920	63	0	98.73	"23 6 1 32 5 24 33 4 3 2 7"
gumbel	IFM	973	3920	63	0	98.73	"23 6 1 32 5 24 33 4 3 2 7"
gaussian	IFM	973	3920	63	0	98.73	"23 6 1 32 5 24 33 4 3 2 10 12 22"
independent	IFM	973	3919	64	0	98.71	"23 6 1 32 5 24 33 4 3 2"
gaussian	IFM	973	3919	64	0	98.71	"23 6 1 32 5 24 33 4 3 2 10 12"
clayton	IFM	973	3918	65	0	98.69	"23 6 1 32 5 24 33 4 3 2 7"
gaussian	IFM	973	3918	65	0	98.69	"23 6 1 32 5 24 33 4 3 2 10 12 22 29 31"

Çizelge 3.14'te de görüldüğü gibi, en iyi başarım oranını %98.73 ile gumbel, independent ve gaussian copulaları IFM metodu kullanılarak elde edilmiştir. Gumbel copula ailesi için bu başarım "23 6 1 32 5 24 33 4 3 2" ve "23 6 1 32 5 24 33 4 3 2 7" özellik setleri kullanılarak elde edilmiştir. Bu iki özellik seti arasında az özellikle aynı başarımı gösteren tercih edilmelidir. İndependent copulası bu başarım oranını "23 6 1 32 5 24 33 4 3 2 7" özellikleri ile elde ederken, gaussian copula ailesi ise "23 6 1 32 5 24 33 4 3 2 10 12 22" özellikleri ile elde etmiştir. Bu durumda gumbel copula ailesi, independent copula ailesi ve gaussian copula ailesine göre daha az özellik kullanılarak en iyi performansı elde etmiştir. KDD10 veri setinin %1'lik kısmı için gumbel copula ailesi, IFM metodu ve "23 6 1 32 5 24 33 4 3 2" özellikleri tercih edilmelidir. Çizelge 3.15'te KDD10 veri setinde bulunan her bir saldırı tipinin %5'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları gösterilmiştir.

Çizelge 3.15. KDD10 veri setinde bulunan her bir saldırı tipinin %5'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
independent	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
gaussian	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
frank	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
clayton	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
gumbel	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
t	**IFM**	**4788**	**19707**	**142**	**76**	**99.12**	**"23 6"**
independent	IFM	4853	19637	212	11	99.10	"23 6 1 32 5 24 33 4 3"
gumbel	IFM	4794	19692	157	70	99.08	"23 6 1"

Çizelge 3.15'te de görüldüğü gibi, en iyi başarım oranını %99.12 ile gumbel, independent, clayton, gaussian, student_t ve frank copula aileleri "23 6" özellikleri ile IFM metodu kullanılarak elde edilmiştir. KDD10 veri setinin %5'lik kısmı için gumbel, independent, clayton, gaussian, student_t ve frank copula ailelerinden herhangi biri, IFM metodu ve "23 6" özellikleri tercih

edilmelidir. Çizelge 3.16'da KDD10 veri setinde bulunan her bir saldırı tipinin %10'luk oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları gösterilmiştir.

Çizelge 3.16. KDD10 veri setinde bulunan her bir saldırı tipinin %10'luk oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
independent	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
gaussian	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
frank	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
clayton	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
gumbel	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
t	**IFM**	**9585**	**39373**	**310**	**143**	**99.08**	**"23 6"**
independent	IFM	9585	39373	310	143	99.08	"23 6 1"
gaussian	IFM	9585	39373	310	143	99.08	"23 6 1"
frank	IFM	9585	39373	310	143	99.08	"23 6 1"
clayton	IFM	9585	39373	310	143	99.08	"23 6 1"
gumbel	IFM	9585	39373	310	143	99.08	"23 6 1"
t	IFM	9584	39374	309	144	99.08	"23 6 1"
gumbel	IFM	9685	39260	423	43	99.06	"23 6 1 32 5 24 33 4 3"
independent	IFM	9617	39309	374	111	99.02	"23 6 1 32 5 24 33 4 3"

Çizelge 3.16'da da görüldüğü gibi, en iyi başarım oranını %99.08 ile gumbel, independent, clayton, gaussian, student_t ve frank copula aileleri "23 6" ve "23 6 1" özellik setleri ve IFM metodu kullanılarak elde edilmiştir. Gumbel, independent, clayton, gaussian, student_t ve frank copula aileleri için özellik sayısı daha az olan "23 6" özellikleri tercih edilmelidir. KDD10 veri setinin %10'lik kısmı için gumbel, independent, clayton, gaussian, student_t ve frank copula ailelerinden herhangi biri, IFM metodu ve "23 6" özellikleri tercih edilmelidir. Çizelge 3.17'de KDD10 veri setinde bulunan her bir saldırı tipinin %50'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları gösterilmiştir.

Çizelge 3.17. KDD10 veri setinde bulunan her bir saldırı tipinin %50'lik oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
gumbel	**CML**	**48174**	**196287**	**2090**	**465**	**98.97**	**"23 6 1 32 5 24 33 4 3 2 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 26 27 28 29 30"**
gaussian	IFM	48128	196301	2076	511	98.95	"23 6 1 32 5 24 33 4 3 2 7 10 11 12"
gumbel	CML	48250	196171	2206	389	98.95	"23 6 1 32 5 24 33 4 3 2 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 26 27 28"

Çizelge 3.17' de de görüldüğü gibi, en iyi başarım oranını %98.97 ile gumbel copula ailesi "23 6 1 32 5 24 33 4 3 2 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 26 27 28 29 30" özellikler ile CML metodu kullanılarak elde etmiştir. Gaussian copula ailesi ise %98.95 başarım oranını "23

6 1 32 5 24 33 4 3 2 7 10 11 12" özellikleri ile IFM metodu kullanılarak elde etmiştir. KDD10 veri setinin %50'lik kısmı için gumbel copula ailesi, CML metodu ve "23 6 1 32 5 24 33 4 3 2 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 26 27 28 29 30" özellikleri tercih edilmelidir. Çizelge 3.18'de KDD10 veri setinde bulunan her bir saldırı tipinin %100'lük oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları gösterilmiştir.

Çizelge 3.18. KDD10 veri setinde bulunan her bir saldırı tipinin %100'lük oranları kullanılarak en iyi üç performansı gösteren copula ailelerinin başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
gaussian	**IFM**	**96325**	**391193**	**5550**	**953**	**98.68**	**"23 6 1 32 5 24 33 4 3 2 7 9 10 11"**
gaussian	IFM	95710	391601	5142	1568	98.64	"23 6 1 32 5 24 33 4"
gaussian	IFM	96183	390323	6420	1095	98.48	"23 6 1 32 5 24 33 4 3"

Çizelge 3.18'de de görüldüğü gibi, en iyi başarım oranını %98.68 ile "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikleri ve IFM metodu kullanılarak gaussian copulası ailesi elde etmiştir. KDD10 veri setinin %100'lük kısmı için gaussian copula ailesi, IFM metodu ve "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikleri tercih edilmelidir.

3.2.2. Uygulama 5: KDD100

Uygulama 5'te; gumbel, independent, clayton, gaussian, student's-t ve frank copula aileleri ile CML, IFM metotları kullanılarak saldırı tespiti yapılmıştır. Çizelge 3.18'de de görüldüğü üzere, KDD10 veri seti üzerinde en iyi üç başarımı elde eden "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellik kümesi, "23 6 1 32 5 24 33 4" özellik kümesi ve "23 6 1 32 5 24 33 4 3" özellik kümesi bu örnek çalışmada kullanılmıştır. KDD100 veri setinde bulunan bu özellik kümeleri seçilerek yukarıda bahsedilen altı copula ailesi kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. KDD100 veri seti 10 çekirdekli 2 adet Intel Xeon(R) CPU E52687Wv3@ 3.10 GHz işlemcisi, 64 GB Ram ve Nvidia Quadro P5000 GPU'su olan bir HP-Z840 iş istasyonunda eğitilmiştir. Eğitimler MATLAB ortamında yapılmıştır. Sınıflandırma aşamasında 10-kat çapraz doğrulama tekniği kullanılmıştır. Sınıflandırıcıların değerlendirme metrikleri Çizelge 3.6'da hata matrisi kullanılarak elde edilmiştir. Student's-t copulasının serbestlik derecesinin (v) çok büyük olmasında dolayı ölçümler yapılırken hata vermiştir. Bundan dolayı; Çizelge 3.20, Çizelge 3.21 ve Çizelge 3.22'de gösterilmemiştir. Çizelge 3.19'da KDD100 veri seti üzerinde en iyi performansı elde eden özelliklerin numaraları ve isimleri gösterilmiştir.

Çizelge 3.19. KDD100 veri seti üzerinde en iyi performansı elde eden özelliklerin numaraları ve isimleri

Özellik Numarası	Özellik Adı
23	count
6	dst_bytes
1	duration
32	dst_host_count
5	src_bytes
24	srv_count
33	dst_host_srv_count
4	flag

Çizelge 3.20'de "23 6 1 32 5 24 33 4" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları gösterilmiştir.

Çizelge 3.20. "23 6 1 32 5 24 33 4" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
independent	IFM	932928	3905436	20214	39853	98.77	"23 6 1 32 5 24 33 4"
gaussian	**IFM**	**972741**	**3896885**	**28765**	**40**	**99.41**	**"23 6 1 32 5 24 33 4"**
clayton	IFM	872190	3909171	16479	100591	97.61	"23 6 1 32 5 24 33 4"
frank	IFM	906564	3906204	19446	66217	98.25	"23 6 1 32 5 24 33 4"
gumbel	IFM	933022	3905382	20268	39759	98.77	"23 6 1 32 5 24 33 4"

Çizelge 3.20'de de görüldüğü gibi, en iyi başarım oranını %99.41 ile "23 6 1 32 5 24 33 4" özellikleri ve IFM metodu kullanılarak gaussian copula ailesi elde etmiştir. En kötü başarım oranı ise %97.61 ile "23 6 1 32 5 24 33 4" özellikleri ve IFM metodu kullanılarak clayton copula ailesi elde etmiştir. KDD100 veri setinde bulunan "23 6 1 32 5 24 33 4" özellikler için gaussian copula ailesi ve IFM metodu tercih edilmelidir. Çizelge 3.21'de "23 6 1 32 5 24 33 4 3" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları gösterilmiştir.

Çizelge 3.21. "23 6 1 32 5 24 33 4 3" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
independent	IFM	933635	3905622	20028	39146	98.79	"23 6 1 32 5 24 33 4 3"
gaussian	**IFM**	**972676**	**3894142**	**31508**	**105**	**99.35**	**"23 6 1 32 5 24 33 4 3"**
clayton	IFM	831992	3912384	13266	140789	96.86	"23 6 1 32 5 24 33 4 3"
frank	IFM	906560	3906189	19461	66221	98.25	"23 6 1 32 5 24 33 4 3"
gumbel	IFM	933740	3905665	19985	39041	98.80	"23 6 1 32 5 24 33 4 3"

Çizelge 3.21'de de görüldüğü gibi, en iyi başarım oranını %99.35 ile "23 6 1 32 5 24 33 4 3" özellikleri ve IFM metodu kullanılarak gaussian copula ailesi elde etmiştir. En kötü başarım oranı ise %96.86 ile "23 6 1 32 5 24 33 4 3" özellikleri ve IFM metodu kullanılarak clayton copula ailesi elde etmiştir. KDD100 veri setinde bulunan "23 6 1 32 5 24 33 4 3" özellikler için gaussian copula ailesi ve IFM metodu tercih edilmelidir. Çizelge 3.22'de "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları gösterilmiştir.

Çizelge 3.22. "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikleri kullanılarak copula ailelerinin KDD100 veri seti üzerindeki başarım oranları

Copula Ailesi	Metot	TP	TN	FP	FN	Doğruluk (%)	Kullanılan Özellikler
independent	IFM	922035	3907600	18050	50746	98.60	"23 6 1 32 5 24 33 4 3 2 7 9 10 11"
gaussian	**IFM**	**972757**	**3895289**	**30361**	**24**	**99.38**	**"23 6 1 32 5 24 33 4 3 2 7 9 10 11"**
clayton	IFM	910658	3908739	16911	62123	98.39	"23 6 1 32 5 24 33 4 3 2 7 9 10 11"
frank	IFM	658964	3909941	15709	313817	93.27	"23 6 1 32 5 24 33 4 3 2 7 9 10 11"
gumbel	IFM	926372	3907381	18269	46409	98.68	"23 6 1 32 5 24 33 4 3 2 7 9 10 11"

Çizelge 3.22'de de görüldüğü gibi, en iyi başarım oranını %99.38 ile "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikleri ve IFM metodu kullanılarak gaussian copula ailesi elde etmiştir. En kötü başarım oranı ise %93.27 ile "23 6 1 32 5 24 33 4" özellikleri ve IFM metodu kullanılarak frank copula ailesi elde etmiştir. KDD100 veri setinde bulunan "23 6 1 32 5 24 33 4 3 2 7 9 10 11" özellikler için gaussian copula ailesi ve IFM metodu tercih edilmelidir.

4. SONUÇ

Bu tez çalışmasında, günümüzde en çok kullanılan makine öğrenme sınıflandırıcıları ile copula tabanlı sınıflandırıcılar kullanılarak saldırı tespiti gerçekleştirilmiştir. Makine öğrenme sınıflandırıcıları olarak; KA, TÖ ve DVM sınıflandırıcıları tercih edilmiştir. Bu üç sınıflandırma tekniği kullanılarak KDD'99 veri kümesi altında bulunan; KDD10, KDD100, KDDTEST ve KDD10+KDDTEST veri seti üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmış olup, en iyi başarım oranları KDD10 veri setinde %99.99, KDDTEST veri setinde %97.90, KDD10+KDDTEST veri setinde %100 ve KDD100 veri setinde ise %100 doğrulukla TÖ sınıflandırıcılarından olan bagged trees sınıflandırıcısı elde etmiştir. Çizelge 4.1'de makine öğrenmesi tabanlı sınıflandırıcıların farklı veri seti miktarları için başarım kıyaslaması gösterilmiştir.

Çizelge 4.1. Makine öğrenmesi tabanlı sınıflandırıcıların farklı veri seti miktarları için başarım kıyaslaması

Kullanılan Yöntem	En iyi Algoritma	Kullanılan Veri Seti	Başarım Oranı (%)
TÖ	Bagged Trees	KDD10	99.99
TÖ	Bagged Trees	KDDTEST	97.90
TÖ	Bagged Trees	KDD10+KDDTEST	100.00
TÖ	Bagged Trees	KDD100	100.00
YSA	Levenberg-Marquardt	KDD10	95.31

Çizelge 4.1'de de görüldüğü üzere, veri setinin çeşitli versiyonlarına göre makine öğrenme sınıflandırıcıları kullanılarak saldırı tespiti yapılmıştır. Genel olarak veri setinin miktarına bakılmaksızın TÖ sınıflandırıcılarından olan bagged trees sınıflandırıcısının daha iyi performans elde ettiği görülmüştür. Çizelge 4.2'de copula-tabanlı sınıflandırıcıların farklı veri seti miktarları için başarım kıyaslaması gösterilmiştir.

Çizelge 4.2. Copula-tabanlı sınıflandırıcıların farklı veri seti miktarları için başarım kıyaslaması

Kullanılan Yöntem	En iyi Algoritma	Kullanılan Veri Seti	Başarım Oranı (%)
Copula	Gumbel-IFM	KDD10 (%1)	98.73
Copula	Independent-IFM Gaussian-IFM Frank-IFM Clayton-IFM Gumbel-IFM (Student's-t)-IFM	KDD10 (%5)	99.12
Copula	Independent-IFM Gaussian-IFM Frank-IFM Clayton-IFM Gumbel-IFM (Student's-t)-IFM	KDD10 (%10)	99.08
Copula	Gumbel-CML	KDD10 (%50)	98.97
Copula	Gaussian-IFM	KDD10 (%100)	98.68
Copula	**Gaussian-IFM**	**KDD100**	**99.41**

Çizelge 4.2'de de görüldüğü üzere, veri setinin çeşitli versiyonlarına göre copula tabanlı sınıflandırıcılar kullanılarak saldırı tespiti yapılmıştır. Veri setinin miktarı az alındığında gumbel copula tabanlı sınıflandırıcı daha iyi performans elde etmiştir. Veri miktarı artıkça gaussian copula tabanlı sınıflandırıcı ön plana çıkmaktadır. Çizelge 4.3'te daha önce literatürde STS'ler ile ilgili yapılan bazı çalışmaların başarım oranları gösterilmiştir.

Çizelge 4.3. Daha önce literatürde STS'ler ile ilgili yapılan bazı çalışmaların başarım oranları

Literatürdeki Bazı Çalışmalar	Kullanılan Yöntem	Kullanılan Veri Seti	Başarım Oranı (%)
A.Dastanpour ve ark[12]	GA+YSA	KDD'99	100.00
J.Esmaily ve ark[18]	YSA	KDD'99	99.71
Önerilen çalışma	**Copula**	**KDD'99**	**99.41**
W.Wang ve ark[15]	TBA	DARPA	98.80
Y.B.Bhavsar ve ark[19]	DVM	NSL-KDD	98.57
Ş.Sağıroğlu ve ark[3]	YSA	KDD'99	97.92
B.W.Masduki ve ark[2]	DVM	KDD'99	96.08
G.Poojitha ve ark[20]	YSA	KDD'99	94.93
S.Kumar ve ark[16]	YSA	KDD'99	91.90
H.A.Sonawane ve ark[10]	SA	KDD'99	90.20
M.Govindarajan ve ark[11]	RTF+DVM	NSL-KDD	85.19
B.Huyot ve ark[4]	Copula	DARPA	79.00

Çizelge 4.3'te de görüldüğü üzere, STS'lerde birçok farklı yöntem kullanılarak saldırı tespiti gerçekleştirilmiştir. A.Dastanpour ve ark[12], yaptıkları çalışmada KDD'99 veri setinin 18 özelliğini kullanarak %100 başarım elde etmişlerdir. J.Esmaily ve ark[18] , yaptıkları çalışmada KDD'99 veri setinin tüm (41) özelliklerini kullanarak %99.71 başarım elde etmişlerdir. Bu tez çalışmasında ise Çizelge 3.20'de görüldüğü gibi 8 özellik kullanılarak %99.41 oranında başarım

elde edilmiştir. Copula tabanlı sınıflandırıcılardan elde edilen sonuçlar, daha önce yapılan çalışmalar ile kıyaslandığında oldukça kayda değer sonuçlar elde edilmiştir. Böylece, copula tabanlı sınıflandırıcıların makine öğrenme sınıflandırıcılarına alternatif olabileceği kanısına varılmıştır.

Sonuç olarak; bu tez çalışmasında gumbel, independent, clayton, gaussian, student's-t ve frank copula tabanlı sınıflandırıcıları tercih edilmiş olup, bu copula tabanlı sınıflandırıcıların saldırı tespit sistemlerinde kullanılabilirliği araştırılmıştır. Copula tabanlı sınıflandırıcılar kullanılarak, KDD10 ve KDD100 veri setleri üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Sınıflandırma aşamasında 10-kat çapraz doğrulama tekniği kullanılmıştır. KDD10 veri seti üzerinde tüm copula sınıflandırıcıları %99.12 gibi iyi bir başarım elde ederken, KDD100 veri seti üzerinde ise en iyi başarım oranını %99.41 ile gaussian copula tabanlı sınıflandırıcı elde etmiştir. Çizelge 4.2'de de görüldüğü üzere, copula tabanlı sınıflandırıcılar diğer yöntemler ile kıyaslandığında gayet iyi değerler elde etmiştir.

Sonraki çalışmalarda bu copula ailelerine ek olarak farklı copula aileleri kullanılarak saldırı tespiti başarımları incelenecektir. Ayrıca YSA ile copula tabanlı yaklaşımların birlikte kullanılabilirliği araştırılacaktır.

5. KAYNAKLAR

[1] Burukanlı M, Budak Ü, Çıbuk M, 2019. Saldırı Tespit Sistemlerinde Makine Öğrenme Metotlarının Kullanımı. Uluslararası Bilim ve Mühendislik Sempozyumu, 20-22 Haziran 2019, Siirt, Türkiye, s: 1052–1057.

[2] Masduki BW, Ramli K, Saputra FA, Sugiarto D, 2015. Study on Implementation of Machine Learning Methods Combination for Improving Attacks Detection Accuracy on Intrusion Detection System (IDS). 2015 Int. Conf. Qual. Res, 10-13 Aug. 2015, Lombok, Indonesia, s: 56–64.

[3] Sağıroğlu Ş, Yolaçan EN, Yavanoğlu U, 2011. Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi. J Fac Eng Arch Gazi Univ, 26 (2): 325–340.

[4] Huyot B, Mabiala Y, Marcotorchino J-F, 2014. Online Unsupervised Anomaly Detection in Large Information Systems Using Copula Theory. 2014 IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst, 27-29 Nov. 2014, Shenzhen, China, s: 679–684.

[5] Salinas Gutiérrez R, Hernández Aguirre A, Rivera Meraz MJJ, Villa Diharce ER, 2010. Using Gaussian Copulas in Supervised Probabilistic Classification. 355-372, in: Soft Computing for Intelligent Control and Mobile Robotics (eds: Castillo C, Kacprzyk J, Pedrycz W). Springer-Verlag Berlin and Heidelberg GmbH & Co. KG Press, Heidelberg.

[6] Scavnicky M, 2013. A study of Applying Copulas in Data Mining. Master's Thesis, Charles University in Prague Faculty of Mathematics and Physics, Prague.

[7] Mukkamala S, Janoski G, Sung A, 2002. Intrusion Detection Using Neural Networks and Support Vector Machines. Proc. 2002 Int. Jt. Conf. Neural Networks. IJCNN'02 (Cat. No.02CH37290), 12-17 May 2002, Honolulu, HI, USA, USA, s: 1702–1707.

[8] Moradi M, Zulkernine M, 2004. A Neural Network Based System for Intrusion Detection and Classification of Attacks. Proc. IEEE Int. Conf. Adv. Intell. Syst. Appl, November 2004, Luxembourg-Kirchberg, Canada, s: 15–18.

[9] Mukkamala S, Sung AH, 2003. Artificial Intelligent Techniques for Intrusion Detection. SMC'03 Conf. Proceedings. 2003 IEEE Int. Conf. Syst. Man Cybern. Conf. Theme - Syst. Secur. Assur. (Cat. No.03CH37483), 8-8 Oct. 2003, Washington DC, USA, s: 1266–1271.

[10] Sonawane HA, Pattewar TM, 2015. A Comparative Performance Evaluation of Intrusion Detection Based on Neural Network and PCA. 2015 Int. Conf. Commun. Signal Process, 2-4 April 2015, Melmaruvathur, India, s: 841–845.

[11] Govindarajan M, Chandrasekaran RM, 2012. Intrusion Detection using an Ensemble of

Classification Methods. Lect. Notes Eng. Comput. Sci, 24-26 October 2012, San Francisco, USA, s: 459–464.

[12] Dastanpour A, Ibrahim S, Mashinchi R, Selamat A, 2014. Comparison of Genetic Algorithm Optimization on Artificial Neural Network and Support Vector Machine in Intrusion Detection System. 2014 IEEE Conf. Open Syst, 26-28 October 2014, Subang, Malaysia, s: 72–77.

[13] Jalil KA, Kamarudin MH, Masrek MN, 2010. Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion. 2010 Int. Conf. Netw. Inf. Technol (ICNIT), 11-12 June 2010, Manila, Philippines, s: 221–226.

[14] Golovko VA, Vaitsekhovich LU, Kochurko PA, Rubanau US, 2007. Dimensionality Reduction and Attack Recognition using Neural Network Approaches. 2007 Int. Jt. Conf. Neural Networks, 12-17 Aug. 2007,Orlando, FL, USA, s: 2734–2739.

[15] Wang W, Battiti R, 2006. Identifying Intrusions in Computer Networks with Principal Component Analysis. First Int. Conf. Availability, Reliab. Secur, 20-22 April 2006, Vienna, Austria, Austria, s: 270–279.

[16] Kumar S, Yadav A, 2014. Increasing Performance Of Intrusion Detection System Using Neural Network. 2014 IEEE Int. Conf. Adv. Commun. Control Comput. Technol, 8-10 May 2014, Ramanathapuram, India, s: 546–550.

[17] Haddadi F, Khanchi S, Shetabi M, Derhami V, 2010. Intrusion Detection and Attack Classification Using Feed-Forward Neural Network. 2010 Second Int. Conf. Comput. Netw. Technol, 23-25 April 2010, Bangkok, Thailand, s: 262–266.

[18] Esmaily J, Moradinezhad R, Ghasemi J, 2015. Intrusion Detection System Based on Multi-Layer Perceptron Neural Networks and Decision Tree. 2015 7th Conf. Inf. Knowl. Technol, 26-28 May 2015, Urmia, Iran, s: 1–5.

[19] Bhavsar YB, Waghmare KC, 2013. Intrusion Detection System using Data Mining Technique: Support Vector Machine. Int J Emerg Technol Adv Eng, 3 (3): 581–586.

[20] Poojitha G, Kumar KN, Reddy PJ, 2010. Intrusion Detection using Artificial Neural Network. 2010 Second Int. Conf. Comput. Commun. Netw. Technol, 29-31 July 2010, Karur, India, s: 1–7.

[21] Shum J, Malki HA, 2008. Network Intrusion Detection System Using Neural Networks. Proc. - 4th Int. Conf. Nat. Comput (ICNC), 18-20 Oct. 2008, Jinan, China, s: 242–246.

[22] Ahmad I, Abdullah AB, Alghamdi AS, 2010. Remote to Local Attack Detection Using Supervised Neural Network. 2010 Int. Conf. Internet Technol. Secur. Trans, 8-11 Nov. 2010, London, UK, s: 1–6.

[23] Ali KM, W V, Rababaa MSA, 2009. The Affect of Fuzzification on Neural Networks Intrusion Detection System. 2009 4th IEEE Conf. Ind. Electron. Appl (ICIEA), 25-27 May 2009, Xi'an, China, s: 1236–1241.

[24] Hodo E, Bellekens X, Hamilton A, Dubouilh P-L, Iorkyase E, Tachtatzis C, Atkinson R, 2016. Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System. 2016 Int. Symp. Networks, Comput. Commun, 11-13 May 2016, Yasmine Hammamet, Tunisia, s: 1–6.

[25] Dias LP, Cerqueira JJF, Assis KDR, Almeida RC, 2017. Using Artificial Neural Network in Intrusion Detection Systems to Computer Networks. 2017 9th Comput. Sci. Electron. Eng, 27-29 Sept. 2017, Colchester, UK, s: 145–150.

[26] Çelebioğlu S, 2003. Arşimedyen Kapulalar ve Bir Uygulama. Selçuk Üniversitesi Fen Fakültesi Fen Derg, 22(1):43–52.

[27] Gülöksüz ÇT, 2015. Dolar Kuru ile Tüketici Fiyat Endeksi Arasındaki İlişkinin Archimedean Kapula ile Modellenmesi. Bankacılık ve Sigortacılık Araştırmaları Derg, 2 (7): 53–62.

[28] Tosunoğlu F, Can İ, 2015. Erzurum İli Kuraklıkların İki Değişkenli Frekans Analizi : Kopula Fonksiyonlarının Kullanımı. http://www.imo.org.tr/resimler/ekutuphane/pdf/17684_54_08.pdf (Erişim Tarihi: 12/04/2020).

[29] Arslan S, Çelebioğlu S, Öztürk F, 2012. İki Boyutlu Arşimedyen Kopulalarda İstatiksel Sonuç Çıkarımı ve Bir Uygulama. Gazi Üniversitesi İktisadi ve İdari Bilim Fakültesi Derg, 14 (2): 1–18.

[30] Sathe S, 2006. A Novel Bayesian Classifier using Copula Functions. https://arxiv.org/abs/cs/0611150 (Erişim Tarihi: 17/05/2020).

[31] Qian D, Wang B, Qing X, Zhang T, Zhang Y, Wang X, Nakamura M, 2017. Drowsiness Detection by Bayesian-Copula Discriminant Classifier Based on EEG Signals during Daytime Short Nap. IEEE Trans Biomed Eng, 64 (4): 743–754.

[32] Slechan L, Górecki J, 2015. On the Accuracy of Copula-Based Bayesian Classifiers: An Experimental Comparison with Neural Networks. 485-493, in: Computational Collective Intelligence (eds: Nunez M, Nguyen NT, Camacho D, Trawinski B). Springer International Press, Madrid.

[33] Ozdemir O, Allen TG, Choi S, Wimalajeewa T, Varshney PK, 2018. Copula Based Classifier Fusion under Statistical Dependence. IEEE Trans Pattern Anal Mach Intell, 40 (11): 2740–2748.

[34] Chen Y, 2014. A Copula-Based Supervised Learning Classification for Continuous and Discrete Data. J Data Sci, 13: 769–790.

[35] Hammami N, Bedda M, Farah N, 2013. Probabilistic Classification Based on Gaussian Copula for Speech Recognition: Application to Spoken Arabic Digits. 2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 26-28 Sept. 2013, Poznan, Poland, s: 312–317.

[36] He Y, Deng J, Li H, 2017. Short-Term Power Load Forecasting with Deep Belief Network and Copula Models. 2017 9th Int. Conf. Intell. Human-Machine Syst. Cybern, 26-27 Aug. 2017, Hangzhou, China, s: 191–194.

[37] Karataş AM, 2018. Modeling of daily maximum and minimum temperature changes in Bitlis province using Copula Method. BEÜ Fen Bilim Derg, 7 (2): 268–275.

[38] Nelsen RB, 2006. An Introduction to Copulas. Springer Science+Business Media, Inc. Press. Portland.

[39] Karagül BZ, 2013. Hayat Dışı Sigortalarda Doğrusal Olmayan Bağımlılığın Kopulalar İle Dinamik Finansal Analizi. Yüksek Lisans Tezi, Haccettepe Üniversitesi Fen Bilimler Enstitüsü, Ankara.

[40] Schmidt T, 2006. Coping with Copulas. 3-34, in: Copulas: From theory to application in finance (eds: Rank J). Risk Books Publisihing, Berkeley.

[41] Sklar A, 2020. The Name of Copulas. https://en.wikipedia.org/wiki/Abe_Sklar (Erişim Tarihi: 04/05/2020).

[42] Lu J, Tian W, Zhang P, 2008. The Archimedean Copulas Measure of the Risk Characteristic for the Tail Dependent Asset Returns. 2008 Int. Conf. Manag. Sci. Eng. 15th Annu. Conf. Proc, 10-12 Sept. 2008, Long Beach, CA, USA, s: 173–181.

[43] Sklar A, 1973. Random Variables , Joint Distribution Functions , and Copulas. Kybernetika, 9 (6): 449–460.

[44] Lu J, Tian WJ, Zhang P, 2008. The Extreme Value Copulas Analysis of the Risk Dependence for the Foreign Exchange Data. 2008 4th Int. Conf. Wirel. Commun. Netw. Mob. Comput, 12-14 Oct. 2008, Dalian, China, s: 1–6.

[45] Du J, Li H, 2019. A Rosenblatt Transformation Method Based on Copula Function for Solving Structural Reliability. 2019 Int. Conf. Qual. Reliab. Risk, Maintenance, Saf. Eng. (QR2MSE), August 6-9, 2019, Zhangjiajie, Hunan, China, s: 590–596.

[46] Embrechts P, Lindskog F, Mcneil A, 2003. Modelling Dependence with Copulas and Applications to Risk Management. 329-384, in: Handbook of Heavy Tailed Distributions in Finance (eds: Racher ST) Elsevier Science B.V Press, Amsterdam.

[47] Bouyé E, Durrleman V, Nikeghbali A, Riboulet G, Roncalli T, 2000. Copulas for Finance-A Reading Guide and Some Applications. SSRN Electronic Journal. Korea.

[48] Qu L, Chen H, Tu Y, 2011. Nonparametric Copula Density Estimation in Sensor Networks. 2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks, 16-18 Dec. 2011, Beijing, China, s: 1–8.

[49] Marti G, Nielsen F, Donnat P, 2016. Optimal Copula Transport for Clustering Multivariate Time Series. 2016 IEEE Int. Conf. Acoust. Speech Signal Process, 20-25 March 2016, Shanghai, China, s: 2379–2383.

[50] Yundai X, Yue Y, 2019. Analysis of Aggregated Wind Power Dependence Based on Optimal Vine Copula. 2019 IEEE Innov. Smart Grid Technol. - Asia (ISGT Asia), 21-24 May 2019, Chengdu, China, China, s: 1788–1792.

[51] Behan D, Cox S, 2007. A Procedure for Simulation with Constructed Copulas. https://web.actuaries.ie/sites/default/files/erm-resources/rsrch-final-instr-copula.pdf (Erişim Tarihi: 22/05/2020).

[52] Huang M, Wang Q, Li Y, Ao L, 2011. An Approach for Improvement of Avionics Reliability Assessment Based on Copula Theory. Proc. 2011 9th Int. Conf. Reliab. Maintainab. Saf, 12-15 June 2011, Guiyang, China, s: 179–183.

[53] Surana A, Pinto A, 2010. Analysis of Stochastic Automata Networks Using Copula Functions. 2010 48th Annu. Allert. Conf. Commun. Control. Comput, 29 Sept.-1 Oct. 2010, Allerton, IL, USA, s: 1699–1706.

[54] Iyengar SG, Varshney PK, Damarla T, 2009. A Parametric Copula Based Framework for Multimodal Signal Processing. 2009 IEEE Int. Conf. Acoust. Speech Signal Process, 19-24 April 2009, Taipei, Taiwan, s: 1893–1896.

[55] Iyengar SG, Varshney PK, Damarla T, 2011. A Parametric Copula-Based Framework for Hypothesis Testing Using Heterogeneous Data. IEEE Trans Signal Process, 59 (5): 2308–2319.

[56] Dong Y, Zhang S, Fan G, Zhang L, Yi L, Lin M, 2012. Application of Copula Function in the Reliability Analysis of the Electrical System and the Power Device of Certain-type Armored Vehicle. 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), 25-27 May 2012, Zhangjiajie, China, s: 386–389.

[57] Arslan S, 2013. Arşimedyen Kapulalar Üzerine Bir Çalışma. Doktora tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

[58] Alhan A, 2008. Bağımsızlık Kapulasını İçeren Kapula Aileleri, Kapula Tahmin Yöntemleri Ve İstanbul Menkul Kıymetler Borsasında Sektörler Arası Bağımlılık Yapısı. Doktora Tezi,

Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

[59] Karataş AM, 2018. Modeling of Daily Maximum and Minimum Temperature Changes in Bitlis Province Using Copula Method. BEÜ Fen Bilim Derg, 7 (2): 268–275.

[60] Kim G, Silvapulle MJ, Silvapulle P, 2007. Comparison of Semiparametric and Parametric Methods for Estimating Copulas. Comput Stat Data Anal, 51 (6): 2836–2850.

[61] Manner H, 2007. Estimation and Model Selection of Copulas with an Application to Exchange Rates. Maastricht research school of Economics of TEchnology and ORganizations (METEOR) Press. Maastricht.

[62] Mou P, Tao F, Jia C, Ma W, 2013. A Copula-Based Function Model in Fuzzy Reliability Analysis on The Planetary Steering Gear. 2013 Int. Conf. Qual. Reliab. Risk, Maintenance, Saf. Eng, 15-18 July 2013, Chengdu, China, s: 375–378.

[63] He H, Subramanian A, Shen X, Varshney PK, 2013. A Coalitional Game for Distributed Estimation in Wireless Sensor Networks. 2013 IEEE Int. Conf. Acoust. Speech Signal Process, 26-31 May 2013, Vancouver, BC, Canada, s: 4574–4578.

[64] Jadhav S, Daruwala R, 2016. 3-D modeling of statistical dependencies using Copulas for Wireless Sensor Network. Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking (WiSPNET), 23-25 March 2016, Chennai, India, s: 1886–1889.

[65] Sezgin EE, 2019. Finansal Bağımlılık Analizi: Vine ve CD Vine Copula Yaklaşımları. Yüksek Lisans Tezi, Bitlis Eren Üniversitesi ve Fırat üniversitesi Fen Bilimleri Enstitüsü, Bitlis.

[66] Avutman Ö, 2011. Yatırım Fonu Stratejileri Arasindaki Bağımlılığın Copula ile Modellenmesi ve Bir Uygulama. Yüksek Lisans Tezi, Marmara Üniversitesi Sosoyal Bilimler Enstitüsü ,İstanbul.

[67] Galiani SS, 2003. Copula Functions and Their Application in Pricing and Risk Managing Multiname Credit Derivative Products. Master's Thesis, Department of Mathematics King's College London The Strand , London.

[68] Yan J, 2007. Enjoy the Joy of Copulas: With a Package copula. J Stat Softw, 21(4):1–21.

[69] Brunel N, Pieczynski W, 2005. Unsupervised Signal Restoration Using Hidden Markov Chains with Copulas. Signal Processing, 85 (12): 2304–2315.

[70] Yapakçı G, 2007. Kopulalar Teorisinin Finansta Uygulaması. Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

[71] Aslan S, Çelebioğlu S, Öztürk F, 2012. İki Boyutlu Arşimedyen Kopulalarda İstatistiksel Sonuç Çıkarımı ve Bir Uygulama. Gazi Üniversitesi İktisadi ve İdari Bilim Fakültesi Derg, 14 (2): 1–18.

[72] Andersen L, Sidenius J, 2005. Extensions to the Gaussian Copula: Random Recovery and Random Factor Loadings. J Credit Risk, 1 (1): 29–70.

[73] Çatal D, Albayrak RS, 2013. Ri̇ske Maruz Değer Hesabında Karışım Kopula Kullanimi: Dolar - Euro Portföyü. J Yasar Univ 2013, 8(31):5187–5202.

[74] Mehdizadeh M, Ghazi R, Ghayeni M, 2018. Power System Security Assessment with High Wind Penetration Using the Farms Models Based on their Correlation. IET Renew Power Gener, 12 (8): 893–900.

[75] Sánchez JF, Úbeda-Flores M, 2014. A Characterization of the Orthogonal Grid Constructions of Copulas. IEEE Trans Fuzzy Syst, 22 (4): 1045–1047.

[76] Hájek P, Mesiar R, 2008. On Copulas, Quasicopulas and Fuzzy Logic. Soft Comput, 12 (12): 1239–1243.

[77] Wulp G Van Der, 2003. Using Copulas in Risk Management. Master's Thesis, Tilburg University Department of Econometrics, Tilburg.

[78] Giacomini E, 2005. Risk Management with Copulae. Master's Thesis, Humboldt-Universit¨at zu Berlin Institute for Statistics and Econometrics CASE - Center for Applied Statistics and Economics, Berlin.

[79] Karakas AM, Karakas M, Dogan M, 2017. Archimedean Copula Estimation Parameter with Kendall Distribution Function. Cumhur Sci J, 38 (4): 619–625.

[80] Romano C, 2002. Calibrating and Simulating Copula Functions: An Application To the Italian Stock Market. Risk Manag Funct Capital Viale U Tupini, 180 : 1–26.

[81] Sklar M, 1959. Fonctions de repartition an dimensions et leurs marges. Publ inst Stat univ Paris, 8 : 229–231.

[82] Karakaş AM, 2017. Modelling temperature measurement data by using copula functions. Bitlis Eren Univ J Sci Technol, 7 (1): 27–32.

[83] Fatahi AA, Dokouhaki P, Moghaddam BF, 2011. A Bivariate Control Chart Based on Copula Function. 2011 IEEE Int. Conf. Qual. Reliab (ICQR), 14-17 Sept. 2011, Bangkok, Thailand, s: 292–296.

[84] He H, Varshney PK, 2016. A Coalitional Game for Distributed Inference in Sensor Networks with Dependent Observations. IEEE Trans Signal Process, 64 (7): 1854–1866.

[85] Xi Z, Wang P, 2012. A Copula Based Sampling Method for Residual Life Prediction of Engineering Systems under Uncertainty. 2012 IEEE Conf. Progn. Heal. Manag, 18-21 June 2012, Denver, CO, USA, s: 1–9.

[86] Gholizadeh MH, Amindavar H, 2011. An Analytic Approach in Joint Delay and Doppler Estimation Using Copula. 2011 IEEE Int. Conf. Acoust. Speech Signal Process, 22-27 May

2011, Prague, Czech Republic, s: 4248–4251.

[87] Li Xixi, Wang Qiang, Jia Suling, 2017. Analysis of Topological Properties of Complex Network of Chinese Stock Based on Copula Tail Correlation. 2017 Int. Conf. Serv. Syst. Serv. Manag, 16-18 June 2017, Dalian, China, s: 1–6.

[88] Carrera D, Santana R, Lozano JA, 2016. Vine Copula Classifiers for the Mind Reading Problem. Prog Artif Intell, 5(4):289–305.

[89] Durrleman V, Nikeghbali A, Roncalli T, 2000. Which Copula is the Right One? https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1032545 (Erişim Tarihi: 15/05/2020).

[90] Joe H, 2005. Asymptotic Efficiency of The Two-Stage Estimation Method for Copula-Based Models. J Multivar Anal, 94 (2): 401–419.

[91] Watada J, 2013. A Kernel Density Estimation-Maximum Likelihood Approach to Risk Analysis of Portfolio. 2013 IEEE 8th Int. Symp. Intell. Signal Process, 16-18 Sept. 2013, Funchal, Portugal, s: 37–42.

[92] Zhu Q, Wang S, Chen Z, He Y, Xu Y, 2019. A Virtual Sample Generation Method Based on Kernel Density Estimation and Copula Function for Imbalanced Classification. 2019 IEEE 8th Data Driven Control Learn. Syst. Conf, 24-27 May 2019, Dali, China, China, s: 969–975.

[93] Tran CD, Rudovic OO, Pavlovic V, 2017. Unsupervised Domain Adaptation with Copula Models. 2017 IEEE 27th Int. Work. Mach. Learn. Signal Process, 25-28 Sept. 2017, Tokyo, Japan, s: 1–6.

[94] Joe H, Xu JJ, 1996. The Estimation Method of Inference Functions for Margins for Multivariate Models. Techical Report, Vancouver, s: 1-21.

[95] Frahm G, Junker M, Szimayer A, 2003. Elliptical Copulas: Applicability and Limitations. Stat Probab Lett, 63 (3): 275–286.

[96] Anonim, 2011. Elliptical copulas Normal and T Copula. https://www.vosesoftware.com (Erişim Tarihi: 20/02/2020).

[97] Demarta S, McNeil AJ, 2007. The t Copula and Related Copulas. Int Stat Rev, 73 (1): 111–129.

[98] Anonim, 2017. Probability Distributions. http://www.nematrian.com/Pages/ProbabilityDistributionsCombined.pdf (Erişim Tarihi: 13/04/2020).

[99] Borowicz JM, Norman JP, 2006. The Effects of Parameter Uncertainty in Dependency Structures. ICA Proc, : 1–20.

[100] Anonim, 2020. Copula (Probability Theory).

https://en.wikipedia.org/wiki/Copula_(probability_theory) (Erişim Tarihi: 18/05/2020).

[101] Bacigal T, Mesiar R, Najjari V, 2015. Generators of Copulas and Aggregation. Inf Sci (Ny), 306 : 81–87.

[102] Karakaş AM, Doğan M, 2017. Archimedean Copula Parameter Estimation with Kendall Distribution Function. J Inst Sci Technol, 7 (3): 187–198.

[103] Hofert M, Mächler M, McNeil A, 2013. Archimedean Copulas in High Dimensions: Estimators and Numerical Challenges Motivated by Financial Applications. J la Société Française Stat Rev Stat appliquée, 154 (1): 25–63.

[104] Cui M, Krishnan V, Hodge BM, Zhang J, 2019. A Copula-Based Conditional Probabilistic Forecast Model for Wind Power Ramps. IEEE Trans Smart Grid, 10 (4): 3870–3882.

[105] Yi Ting F, Xiong Wei W, 2011. Econometric Analysis of the Relationships among the Financial Markets. 2011 Int. Conf. Manag. Sci. Eng. 18th Annu. Conf. Proc, 13-15 Sept. 2011, Rome, Italy, s: 964–969.

[106] Siburg KF, Stoimenov P, Weiß GNF, 2015. Forecasting Portfolio-Value-at-Risk with Nonparametric Lower Tail Dependence Estimates. J Bank Financ, 54 : 129–140.

[107] Setiawan A, Soeheri, Panggabean E, Elhias MA, Ikorasaki F, Riski B, 2018. Efficiency of Bayes Theorem in Detecting Early Symptoms of Avian Diseases. 2018 6th Int. Conf. Cyber IT Serv. Manag, 7-9 Aug. 2018, Parapat, Indonesia, Indonesia, s: 1–5.

[108] Sembiring NSB, Ginting E, Fauzi M, Yudi, Tambunan F, Haryanto EV, 2019. An Expert System To Diagnose Herpes Zoster Disease Using Bayes Theorem. 2019 7th Int. Conf. Cyber IT Serv. Manag, 6-8 Nov. 2019, Jakarta, Indonesia, Indonesia, s: 1–3.

[109] Ikorasaki F, Akbar MB, 2018. Detecting Corn Plant Disease with Expert System Using Bayes Theorem Method. 2018 6th Int. Conf. Cyber IT Serv. Manag, 7-9 Aug. 2018, Parapat, Indonesia, Indonesia, s: 1–3.

[110] Leman D, 2018. Expert System Diagnose Tuberculosis Using Bayes Theorem Method and Shafer Dempster Method. 2018 6th Int. Conf. Cyber IT Serv. Manag, 7-9 Aug. 2018, Parapat, Indonesia, Indonesia, s: 1–4.

[111] Jahromi AH, Taheri M, 2017. A Non-Parametric Mixture of Gaussian Naive Bayes Classifiers based on Local Independent Features. 2017 Artif. Intell. Signal Process. Conf, 25-27 Oct. 2017, Shiraz, Iran, s: 209–212.

[112] Netti K, Radhika Y, 2015. A Novel Method for Minimizing Loss of Accuracy in Naive Bayes Classifier. 2015 IEEE Int. Conf. Comput. Intell. Comput. Res, 10-12 Dec. 2015, Madurai, India, s: 1–4.

[113] Farid DM, Zhang L, Rahman CM, Hossain MA, Strachan R, 2014. Hybrid Decision Tree

and Naive Bayes Classifiers for Multi-Class Classification Tasks. Expert Syst Appl, 41 (4): 1937–1946.

[114] Islam MJ, Wu QMJ, Ahmadi M, Sid Ahmed MA, 2007. Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers. 2007 Int. Conf. Converg. Inf. Technol. (ICCIT), 21-23 Nov. 2007, Gyeongju, South Korea, s: 1541–1546.

[115] Murphy KP, 2006. Naive Bayes classifiers. https://www.ic.unicamp.br/~rocha/teaching/2011s1/mc906/aulas/naive-bayes.pdf (Erişim Tarihi: 02/04/2020).

[116] Yang FJ, 2018. An Implementation of Naive Bayes Classifier. 2018 Int. Conf. Comput. Sci. Comput. Intell, 12-14 Dec. 2018, Las Vegas, NV, USA, USA, s: 301–306.

[117] Leung KM, 2007. Naive Bayesian Classifier. http://cis.poly.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf (Erişim Tarihi: 10/04/2020).

[118] Boullé M, 2007. Compression-based Averaging of Selective Naive Bayes Classifiers. J Mach Learn Res, 8 : 1659–1685.

[119] Gao C, Cheng Q, He P, Susilo W, Li J, 2018. Privacy-Preserving Naive Bayes Classifiers Secure against the Substitution-Then-Comparison Attack. Inf Sci (Ny), 444 : 72–88.

[120] Yang Y, Webb GI, 2001. Proportional k-Interval Discretization for Naive-Bayes Classifier. 564-575, in: European Conference on Machine Learning (eds: Raedt LD, Flach P). Springer Press, Freiburg.

[121] Almeida TA, Almeida J, Yamakami A, 2011. Spam Filtering: How the Dimensionality Reduction Affects the Accuracy of Naive Bayes Classifiers. J Internet Serv Appl, 1 (3): 183–200.

[122] Chandrasekar P, Qian K, 2016. The Impact of Data Preprocessing on the Performance of a Naive Bayes Classifier. 2016 IEEE 40th Annu. Comput. Softw. Appl. Conf, 10-14 June 2016, Atlanta, GA, USA, s: 618–619.

[123] Dai W, Xue GR, Yang Q, Yu Y, 2007. Transferring Naive Bayes Classifiers for Text Classification. Proc Natl Conf Artif Intell, 7 : 540–545.

[124] Dezert J, Tchamova A, Han D, Wickramarathne T, 2019. A Simplified Formulation of Generalized Bayes' Theorem. 2019 22th Int. Conf. Inf. Fusion, 2-5 July 2019, Ottawa, Canada, s: 1–8.

[125] Li X, Zhang W, He L, 2019. A Novel Nonparametric Estimation for Conditional Copula Functions Based on Bayes Theorem. IEEE Access, 7 : 186182–186192.

[126] Cassandra C, Sari R, 2018. Agricultural Expert System Design Based on Bayes Theorem.

2018 Int. Conf. Inf. Manag. Technol, 3-5 Sept. 2018, Jakarta, Indonesia, s:315–320.

[127] Faigman DL, Baglioni AJ, 1988. Bayes' Theorem in the Trial Process - Instructing Jurors on the Value of Statistical Evidence. Law Hum Behav, 12 (1): 1–17.

[128] Muangnak N, Pukdee W, Hengsanunkun T, 2010. Classification Students with Learning Disabilities Using Naive Bayes Classifier and Decision Tree. 6th Int. Conf. Networked Comput. Adv. Inf. Manag, 16-18 Aug. 2010, Seoul, South Korea, s: 189–192.

[129] Nurnberger A, Borgelt C, Klose A, 1999. Improving Naive Bayes Classifiers Using Neuro-Fuzzy Learning. ICONIP'99. ANZIIS'99 ANNES'99 ACNN'99. 6th Int. Conf. Neural Inf. Process. Proc. (Cat. No.99EX378), 16-20 Nov. 1999, Perth, WA, Australia, Australia, s: 154–159.

[130] Domingos P, Pazzani M, 1997. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Mach Learn, 29 (2–3): 103–130.

[131] Yang Y, Webb GI, 2002. A Comparative Study of Discretization Methods for Naive-Bayes Classifiers. Knowl Acquis, 2002 : 159–173.

[132] Ramoni M, Sebastiani P, 2001. Roboust Bayes Classifiers. Artif Intell, 125(1–2):209–226.

[133] Dezert J, Tchamova A, Han D, 2018. Total Belief Theorem and Generalized Bayes' Theorem. 2018 21st Int. Conf. Inf. Fusion, 10-13 July 2018, Cambridge, UK, s: 1040–1047.

[134] Anonim, 2020. Maximum A Posteriori Estimation. https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation (Erişim Tarihi: 04/05/2020).

[135] Bogunovic H, Pozo JM, Cardenes R, Roman LS, Frangi AF, 2013. Anatomical Labeling of the Circle of Willis Using Maximum A Posteriori Probability Estimation. IEEE Trans Med Imaging, 32 (9): 1587–1599.

[136] Peng F, Schuurmans D, Wang S, 2004. Augmenting Naive Bayes Classifiers with Statistical Language Models. Inf Retr Boston, 7(3–4):317–345.

[137] Mahfoud M, 2012. Bivariate Archimedean Copulas : An Application to Two Stock Market Indices. Master's Thesis, Business Mathematics & Informatics at the Vrije Universiteit (VU), Amsterdam.

[138] Charpentier A, 2003. Tail Distribution and Dependence Measures. Proc. 34th ASTIN Conf. 27 March 2003, s: 1–25.

[139] Ida A, Ishimura N, Nakamura M, 2014. Note on the Measures of Dependence in Terms of Copulas. Procedia Econ Financ, 14 : 273–279.

[140] Liebscher E, 2014. Copula-Based Dependence Measures. Depend Model, 2 (1): 49–64.

[141] Bekrizadeh H, Jamshidi B, 2017. A New Class of Bivariate Copulas: Dependence Measures

and Properties. Metron, 75 (1): 31–50.

[142] Hamzaçebi C, Kutay F, 2004. Yapay Sinir Ağları ile Türkiye Elektrik Enerjisi Tüketiminin 2010 Yılına KAdar Tahmini. Gazi Üniv Müh Mim Fak Der, 19 (3): 227–233.

[143] Budak H, Erpolat S, 2012. Kredi Riski Tahmininde Yapay Sinir Ağları ve Lojistik Regresyon Analizi Karşılaştırılması. Online Acad J Inf Technol, 3 (8): 19–42.

[144] Öztemel E, 2012. Yapay Sinir Ağları. Papatya Yayıncılık Eğitim. İstanbul.

[145] Anonim, 2020. Yapay Sinir Ağları. https://tr.wikipedia.org/wiki/Yapay_sinir_ağları (Erişim Tarihi: 03/03/2020).

[146] Yurtoğlu H, 2005. Yapay Sinir Ağları Metodolojisi ile Öngürü Modellemesi:Bazı Makroekonomik Değişkenler için Türkiye Örneği. Uzmanlık Tezi, Ekonomik Modeller ve Strateji Araştırmalar Genel müdürlüğü, Ankara.

[147] Bayır F, 2006. Yapay Sinir Ağları ile Tahmin Modellemesi Üzerine Bir Uygulama. Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.

[148] İlkuçar M, Işık AH, Çifci A, 2014. Harmoni Arama ve Geri Yayılım Tabanlı Yapay Sinir Ağı ile Göğüs Kanseri Verilerinin Sınıflandırılması. 2014 IEEE 22nd Signal Process. Commun. Appl. Conf. (SIU), 23-25 April 2014,Trabzon, Turkey, s: 762–765.

[149] Anderson D, Mcneill G, 1992. Artificial Neural Networks Technology. A DACS State-of-the-Art Report, New York, s: 1-83.

[150] Koç ML, Balas C., Aerslan A, 2004. Taş Dolgu Dalgakıranların Yapay Sinir Ağları ile Ön Tasarımı. İMO Tek Dergi, 15 (74): 3351–3375.

[151] Ahad N, Qadir J, Ahsan N, 2016. Neural Networks in Wireless Networks: Techniques, Applications and Guidelines. J Netw Comput Appl, 68 : 1–27.

[152] Forecasting OR, Properties L, Artificial U, Networks N, Study AE, Properties L, 2005. Yapay Sinir Ağları Kullanılarak Konaklama İşletmelerinde Doluluk Oranı Tahmini: Türkiye'deki Konaklama İşletmeleri Üzerine Bir Deneme. Anatolia Tur Araştırmaları Derg, 16 (1): 24–30.

[153] Es HA, Kalender FY, Hamzaçebi C, 2014. Yapay Sinir Ağları ile Türkiye Net Enerji Talep Tahmini. Gazi Üniversitesi Mühendislik Mimar Fakültesi Derg, 29 (3): 495–504.

[154] Komyakov AA, Nikiforov MM, Erbes V V., Cheremisin VT, Ivanchenko VI, 2016. Construction of Electricity Consumption Mathematical Models on Railway Transport Used Artificial Neural Network and Fuzzy Neural Network. 2016 IEEE 16th Int. Conf. Environ. Electr. Eng, 7-10 June 2016, Florence, Italy, s: 1–4.

[155] Karaatlı M, Güngör İ, Demir Y, Kalaycı Ş, 2005. Hisse Senedinin Fiyat Hareketlerinin Yapay Sinir Ağları Yöntemi ile Tahmin Edilmesi. Yönetim ve Ekon Araştırmaları Derg, 3

(3): 38–48.

[156] Fırat M, Güngör M, 2004. Askı Madde Konsantrasyonu ve Miktarının Yapay Sinir Ağları ile Belirlenmesi. Tek Dergi, 15 (73): 3267–3282.

[157] Civalek Ö, Ülker M, 2004. Dikdörtgen Plakların Doğrusal Olmayan Analizinde Yapay Sinir Ağı Yaklaşımı. İMO Tek Dergi, 15 (72): 3171–3190.

[158] Anonim, 2020. Brain. http://en.wikipedia.org/wiki/Brain (Erişim Tarihi: 31/03/2020).

[159] Diler Aİ, 2003. İMKB Ulusal-100 Endeksinin Yönünün Yapay Sinir Ağları ile Tahmin Edilmesi. İMKB Derg, 7 (25–26): 66–81.

[160] Ersoy E, Karal Ö, 2012. Yapay Sinir Ağları Ve İnsan Beyni. İnsan ve Toplum Bilim Araştırmaları Derg, 1 (2): 188–205.

[161] Anonim, 2020. Human Brain. https://en.wikipedia.org/wiki/Human_brain (Erişim Tarihi: 26/02/2020).

[162] Agatonovic Kustrin S, Beresford R, 2000. Basic Concepts of Artificial Neural Network (ANN) Modeling and its Applicatiob in Pharmaceutical Research. J Pharm Biomed Anal, 22 (5): 717–727.

[163] Anonim, 2020. Artificial Neural Network. https://en.wikipedia.org/wiki/Artificial_neural_network (Erişim Tarihi: 25/02/2020).

[164] Oğuz M, 2001. Yalıtkan Maddelerde Elektriksel Delinme Dayanımının Yapay Sinir Ağları İle Belielenmesi. Yüksek Lisans Tezi, İstanbul Teknik Ünivetrsitesi Fen Bilimleri Enstitüsü ,İstanbul.

[165] Zhang Y, Ding X, Liu Y, Griffin PJ, 1996. An Artificial Neural Network Approach to Transformer Fault. IEEE Power Eng Rev, 16 (10): 54–55.

[166] Ataseven B, 2013. Yapay Sinir Ağlari ile Öngörü Modellemesi. DergiPark, 10 (39): 101–115.

[167] Tok K, 2017. Hastalık Teşhisi için Bir Yapay Sinir Ağları Yazılımının Tasarlanması ve Gerçekleştirimesi. Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

[168] Karampelas P, Vita V, Pavlatos C, Mladenov V, Ekonomou L, 2010. Design of Artificial Neural Network Models for the Prediction of the Hellenic Energy Consumption. 10th Symposium on Neural Network Applications in Electrical Engineering, 23-25 Sept. 2010, Belgrade, Serbia, s: 41–44.

[169] Yavuz S, Deveci M, 2012. İstatiksel Normalïzasyon Tekniklerïnin Yapay Sïnir Ağın Performansina Etkisi. Erciyes Üniversitesi İktisadi ve İdari Bilim Fakültesi Derg, 40 : 167–187.

[170] Anonim, 2020. Yapay Sinir Ağları. http://www.derinogrenme.com/2017/03/04/yapay-

sinir-aglari/ (Erişim Tarihi: 26/02/2020).

[171] Yalçın N, 2012. Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları İle Epilepsi Hastalığının Teşhisi. Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

[172] Sönmez F, 2015. Mevduat Bankalarının Karlılığının Yapay Sinir Ağları ile Tahmini : Bir Yazılım Modeli Tasarımı. BDDK Bankacılık ve Finans Piyas, 9 (1): 9–46.

[173] Kaynar O, Taştan S, 2009. Zaman Seris Analizinde MLP Yapay Sinir Ağları ve Arıma Modelinin Karşılaştırılması. Erciyes Üniversitesi İktisadi ve İdari Bilim Fakültesi Dergisi, 33 : 161–172.

[174] Takma Ç, Atıl H, Aksakal V, 2012. Çoklu Doğrusal Regresyon ve Yapay Sinir Ağı Modellerinin Laktasyon Süt Verimlerine Uyum Yeteneklerinin Karşılaştırılması. Kafkas Univ Vet Fak Derg, 18 (6): 941–944.

[175] Güngör İ, Çuhadar M, 2005. Antalya İline Yönelik Alman Turist Talebinin Yapay Sinir Ağlari Yöntemiyle Tahmini. Ticaret ve Tur Eğitim Fakültesi Derg, (1): 84–98.

[176] Samanta B, Al-Balushi KR, 2003. Artificial Neural Network Based Fault Diagnostics of Rolling Element Bearings Using Time-Domain Features. Mech Syst Signal Process, 17 (2): 317–328.

[177] Aktaş R, Doğanay MM, Yıldız B, 2003. Mali Başansızlığın Öngörülmesi : İstatistiksel Yöntemler ve Yapay Sinir Ağı Karşılaştırması. Ankara Üniversitesi SBF Derg, 58 (04): 1–24.

[178] Birgül K, Bertan B, 2009. Yapa Sinir ağlari ile Borsa Endeksi Tahmi̇ni. Yönetim Derg, 20 (63): 25–40.

[179] Sarle WS, 1994. Neural Networks and Statistical Learning. 1994 Proc. Ninet. Annu. SAS Users Gr. Int. Conf. Springer London, April 1994,London, s: 1–13.

[180] Kabalci E, 2020. Yapay Sinir Ağları. https://ekblc.files.wordpress.com/2014/02/ysa.pdf (Erişim Tarihi: 26/02/2020).

[181] Ozyilmaz L, Yildirim T, 2003. Artificial Neural Networks for Diagnosis of Hepatitis Disease. Proc. Int. Jt. Conf. Neural Networks, 20-24 July 2003, Portland, OR, USA, s: 586–589.

[182] Gardner M., Dorling S., 1998. Artificial Neural Networks (The Multilayer Perceptron)-A Riview of Applications in The Atmospheric Sciences. Atmos Environ, 32 (14–15): 2627–2636.

[183] Çayıroğlu İ, 2020. İleri Algoritma Analizi-5: Yapay Si̇nir Ağları. http://papatyabilim.com.tr/PDF/yapay_sinir_aglari.pdf (Erişim Tarihi: 10/10/2020).

[184] Mikaeil AM, Hu W, Hussain SB, 2018. A Low-Latency Traffic Estimation Based TDM-

PON Mobile Front-Haul for Small Cell Cloud-RAN Employing Feed-Forward Artificial Neural Network. Int. Conf. Transparent Opt. Networks, 1-5 July 2018, Bucharest, Romania, s: 1–4.

[185] Schmitz GPJ, Aldrich C, Gouws FS, 1999. ANN-DT: An Algorithm for Extraction of Decision Trees from Artificial Neural Networks. IEEE Trans Neural Networks, 10 (6): 1392–1401.

[186] Hameed AAH, 2017. Robust Adaptive Learning Approach of Artificial Neural Networks. Doktora Tezi,Selçuk Üniversitesi Fen Bilimleri Enstitüsü ,Konya.

[187] Lv X, Jia Y, Liu M, 2012. Dynamic System Simulation Model and Algorithm Based on Artificial Neural Networks. Proc. - 2012 Int. Symp. Instrum. Meas. Sens. Netw. Autom. (IMSNA), 25-28 Aug. 2012, Sanya, China, s: 464–467.

[188] Aşkın D, İskender İ, Mamızadeh A, 2011. Farklı Yapay Sinir Ağları Yöntemleri Kullanılark Kuru Tip Transformatör Sargısının Termal Analizi. Gazi Üniversitesi Mühendislik Mimar Fakültesi Derg, 26 (4): 905–913.

[189] Shimakura Y, Fujisawa Y, Maeda Y, Makino R, Kishi Y, Ono M, Fann JY, Fukusima N, 1993. Short-Term Load Forecasting Using an Artificial Neural Network. Proc 2nd Int Forum Appl Neural Networks to Power Syst ANNPS 1993, 7 (1): 233–238.

[190] Anonim, 2020. Backpropagation. http://en.wikipedia.org/wiki/Backpropagation (Erişim Tarihi: 26/03/2020).

[191] Huang H, Xia XL, 2017. Wine Quality Evaluation Model Based on Artificial Bee Colony and BP Neural Network. Proc. - 2017 Int. Conf. Netw. Inf. Syst. Comput. (ICNISC), 14-16 April 2017,Shanghai, China, s: 83–87.

[192] Hamzaçebi C, Kutay F, 2004. Yapay Sinir Ağları ile Türkiye Elektrik Enerjisi Tüketiminin 2010 Yılına Kadar Tahmini. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Derg, 19 (3): 227–233.

[193] Alp M, Cığızoğlu HK, 2004. Farklı Yapay Sinir Ağı Metodları ile Yağış-Akış İlişkisinin Modellenmesi. itüdergisi/d, 3 (1): 80–88.

[194] Anonim, 2020. Geri Yayılım (Backpropagation). https://devhunteryz.wordpress.com/2018/06/20/geri-yayilimbackpropagation/ (Erişim Tarihi: 26/03/2020).

[195] Karaatlı M, Helvacıoğlu ÖC, Ömürbek N, Tokgöz G, 2012. Yapay Sinir Ağları Yöntemi İle Otomobil Satış Tahmini. Int J Manag Econ Bus, 8 (17): 87–100.

[196] Akcan A, Kartal C, 2011. İMKB Sigorta Endeksini Olusturan Sirketlerin Hisse Senedi Fiyatlarının Yapay Sinir Ağları İle Tahmini. Muhasebe ve Finans Derg, (51) : 27–40.

[197] Tektaş A, Karataş A, 2004. Yapay Sinir Ağları ve Finans Alanaına Uygulaması: Hisse Senedi Fiyat Tahminlemsi. Atatürk Üniversitesi İktisadi ve İdari Bilim Derg, 18 (3–4): 338–349.

[198] Van Gestel T, Suykens JAK, Baesens B, Viaene S, Vanthienen J, Dedene G, De Moor B, Vandewalle J, 2004. Benchmarking Least Squares Support Vector Machine Classifiers. Mach Learn, 54 (1): 5–32.

[199] Yan G, Ma G, Zhu L, Shi Z, 2006. Combining Multiple Support Vector Machines using Fuzzy Integral for Classification. 2006 Int. Conf. Mach. Learn. Cybern, 13-16 Aug. 2006, Dalian, China, China, s: 3438–3441.

[200] Zhibin Liu, Li Bai, 2008. Evaluating the Supplier Cooperative Design Ability Using A Novel Support Vector Machine Algorithm. 2008 12th Int. Conf. Comput. Support. Coop. Work Des, 16-18 April 2008, Xi'an, China, s: 986–989.

[201] Cheng G, Tong X, 2018. Fuzzy Clustering Multiple Kernel Support Vector Machine. 2018 Int. Conf. Wavelet Anal. Pattern Recognit, 15-18 July 2018, Chengdu, China, s: 7–12.

[202] Zhou MM, Li L, Lu YL, 2009. Fuzzy Support Vector Machine Based on Density with Dual Membership. 2009 Int. Conf. Mach. Learn. Cybern, 12-15 July 2009, Hebei, China, s: 674–678.

[203] Xiong SW, Liu HB, Niu XX, 2005. Fuzzy Support Vector Machines Based on FCM Clustering. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 2608–2613.

[204] Wang X, Lu S, 2006. Improved Fuzzy Multicategory Support Vector Machines Classifier. 2006 Int. Conf. Mach. Learn. Cybern, 13-16 Aug. 2006, Dalian, China, China, s: 3585–3589.

[205] He Q, Song X, Yang G, 2006. Linear Programming Approach for the Inverse Problem of Support Vector Machines. 2006 Int. Conf. Mach. Learn. Cybern, 13-16 Aug. 2006, Dalian, China, China, s: 3519–3522.

[206] Suriya Prakash J, Annamalai Vignesh K, Ashok C, Adithyan R, 2012. Multi Class Support Vector Machines Classifier for Machine Vision Application. 2012 Int. Conf. Mach. Vis. Image Process, 14-15 Dec. 2012, Taipei, Taiwan, s: 197–199.

[207] Fung GM, Mangasarian OL, 2005. Multicategory Proximal Support Vector Machine Classifier. Mach Learn, 59 (1–2): 77–97.

[208] Wu Y, Guo L, Li Y, Shen X, Yan W, 2006. Multi-Layer Support Vector Machine and Its Application. 2006 Int. Conf. Mach. Learn. Cybern, 13-16 Aug. 2006, Dalian, China, China, s: 3627–3631.

[209] Liu B, Hao Z-F, Yang X-W, 2005. Nesting Support Vector Machine for Muti-Classification. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4220–4225.

[210] Ertekin Ş, Bottou L, Giles CL, 2011. Nonconvex Online Support Vector Machines. IEEE Trans Pattern Anal Mach Intell, 33 (2): 368–381.

[211] Xue X, He G, Zhao C, 2008. Proving to the Coincidence of the Solutions of the Modified Problem with the Original Problem of Multi-Class Support Vector Machine. 2008 Fifth Int. Conf. Fuzzy Syst. Knowl. Discov, 18-20 Oct. 2008, Shandong, China, s: 43–47.

[212] Niu DX, Wanq Q, Li JC, 2005. Short Term Load Forecasting Model Using Support Vector Machine Based on Artificial Neural Network. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4260–4265.

[213] Tong S, Chang E, 2001. Support Vector Machine Active Learning for Image Retrieval. Proc. ninth ACM Int. Conf. Multimed, September 2001, Ottawa, Canada, s: 107–118.

[214] Ji A, Pang J, Qiu H, 2006. Support Vector Machine for Classification Based on Fuzzy Training Data. Proc. Fifth Int. Conf. Mach. Learn. Cybern, 13-16 August 2006, Dalian, s: 1609–1614.

[215] Meyer D, Leisch F, Hornik K, 2003. The Support Vector Machine under Test. Neurocomputing, 55 (1–2): 169–186.

[216] Huang W, Nakamori Y, Wang S-Y, 2005. Forecasting Stock Market Movement Direction with Support Vector Machine. Comput Oper Res, 32 (10): 2513–2522.

[217] Kim HC, Pang S, Je HM, Kim D, Bang SY, 2003. Constructing Support Vector Machine Ensemble. Pattern Recognit, 36 (12): 2757–2767.

[218] Kanchan BD, Kishor MM, 2016. Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis. 2016 Int. Conf. Glob. Trends Signal Process. Inf. Comput. Commun, 22-24 Dec. 2016, Jalgaon, India, s: 5–10.

[219] Liu H, Xiong S, Chen Q, 2008. Fuzzy Support Vector Machines Based on Convex Hulls. 2008 IEEE Int. Symp. Knowl. Acquis. Model. Work, 21-22 Dec. 2008, Wuhan, China, s: 920–923.

[220] Liu H, Xiong S, 2007. Fuzzy Support Vector Machines Based on Density Clustering. 2007 IEEE Int. Conf. Control Autom, 30 May-1 June 2007, Guangzhou, China, s: 784–787.

[221] Braun AC, Weidner U, Hinz S, 2011. Support Vector Machines, Import Vector Machines and Relevance Vector Machines for Hyperspectral Classification-A Comparison. 2011 3rd Work. Hyperspectral Image Signal Process. Evol. Remote Sens, 6-9 June 2011, Lisbon, Portugal, s: 1–4.

[222] He Q, Chen J, 2005. The Inverse Problem of Support Vector Machines and Its Solution. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4322–4327.

[223] Zhang L, Zhou W, Jiao L, 2004. Wavelet Support Vector Machine. IEEE Trans Syst Man, Cybern Part B Cybern, 34 (1): 34–39.

[224] Wang LS, Xu YT, Zhao LS, 2005. A Kind of Hybrid Classification Algorithm Based on Rough Set and Support Vector Machine. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 1676–1679.

[225] Liu HJ, Wang YN, Lu XF, 2005. A Method to Choose Kernel Function and its parameters for Support Vector Machines. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4277–4280.

[226] Jia YS, Jia CY, Qi HW, 2005. A New Nu-Support Vector Machine for Training Sets with Duplicate Samples. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4370–4373.

[227] Balasundaram S, Kapil N, 2010. Application of Lagrangian Twin Support Vector Machines for Classification. 2010 Second Int. Conf. Mach. Learn. Comput, 9-11 Feb. 2010, Bangalore, India, s: 193–197.

[228] Aydin I, Karakose M, Akin E, 2007. Artificial Immune Based Support Vector Machine Algorithm for Fault Diagnosis of Induction Motors. 2007 Int. Aegean Conf. Electr. Mach. Power Electron, 10-12 Sept. 2007, Bodrum, Turkey, s: 217–221.

[229] Chen G, Dudek G, 2005. Auto-Correlation Wavelet Support Vector Machine and Its Applications to Regression. 2nd Can. Conf. Comput. Robot Vis, 9-11 May 2005, Victoria, BC, Canada, Canada, s: 246–252.

[230] Mehedihasan MA, Nasser M, Pal B, 2013. On the KDD'99 Dataset: Support Vector Machine Based Intrusion Detection System (IDS) with Different Kernels. Int J Electron Commun Comput Eng, 4 (4): 1164–1170.

[231] Jha J, Ragha L, 2013. Intrusion Detection System Using Support Vector Machine. Int. Conf. Work. Adv. Comput (ICWAC). June 2013, New York, USA, s: 25–30.

[232] Widodo A, Yang B-S, 2007. Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis. Mech Syst Signal Process, 21 (6): 2560–2574.

[233] Lu SX, Meng J, Cao GE, 2010. Support Vector Machine Based on A New Reduced Samples Method. 2010 Int. Conf. Mach. Learn. Cybern, 11-14 July 2010, Qingdao, China, s: 1510–1514.

[234] Amari S, Wu S, 1999. Improving Support Vector Machine Classifier by Modifying Kernel

Functions. Neural Networks, 12 (6): 783–789.

[235] Ji GR, Han P, Zhani YJ, 2007. Wind Speed Forecasting Based on Support Vector Machine with Forecasting Error Estimation. Int. Conf. Mach. Learn. Cybern, 19-22 Aug. 2007, Hong Kong, China, s:2735–2739.

[236] Tang H, Qu LS, 2008. Fuzzy Support Vector Machine with A New Fuzzy Membership Function For Pattern Classification. 2008 Int. Conf. Mach. Learn. Cybern, 12-15 July 2008, Kunming, China, s: 768–773.

[237] Zhao QH, Ha MH, Peng GB, Zhang XK, 2009. Support Vector Machine Based on Half-Suppressed Fuzzy C-Means Clustering. 2009 Int. Conf. Mach. Learn. Cybern, 12-15 July 2009, Hebei, China, s: 1236–1240.

[238] Subaira AS, Anitha P, 2014. Efficient Classification Mechanism for Network Intrusion Detection System Based on Data Mining Techniques:A Survey. 2014 IEEE 8th Int. Conf. Intell. Syst. Control, 10-11 Jan. 2014, Coimbatore, India, s: 274–280.

[239] Dreiseitl S, Ohno-Machado L, 2002. Logistic Regression and Artificial Neural Network Classification Models:A Methodology Reivew. J Biomed Inform, 35 (5–6): 352–359.

[240] Zhou JG, Wang K, Wu J, Yan PL, Wu Ming, 2005. A Method of Chinese Text Categorization Based on Proximal Support Vector Machine. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 1615–1619.

[241] Karacalarli U, 2018. Destek Vektör Makinesi (DVM) Sınıflandırma Metodu Kullanan Saldırı Tespit Sistemlerinin Performansının Özellik Seçimi ile Artırılması. Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

[242] Suykens JAK, Vandewalle J, 1999. Least Squares Support Vector Machine Classifier. Neural Process Lett, 9 (3): 293–300.

[243] Elaidi H, Elhaddar Y, Benabbou Z, Abbar H, 2018. An Idea of A Clustering Algorithm Using Support Vector Machines Based on Binary Decision Tree. 2018 Int. Conf. Intell. Syst. Comput. Vis, 2-4 April 2018, Fez, Morocco, s: 1–5.

[244] Chen P, Wen T, 2006. Margin Maximization Model of Text Classification Based on Support Vector Machines. 2006 Int. Conf. Mach. Learn. Cybern, 13-16 Aug. 2006, Dalian, China, China, s: 3514–3518.

[245] Sun W, Ma G, 2009. Condition Assessment of Power Supply Equipment Based on Kernel Principal Component Analysis and Multi-Class Support Vector Machine. 2009 Fifth Int. Conf. Nat. Comput, 14-16 Aug. 2009, Tianjin, China, s: 485–488.

[246] Lei LY, Sun ZH, 2005. Soft Sensor Based on Generalized Support Vector Machines for Microbiological Fermentation. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005,

Guangzhou, China, China, s: 4305–4309.

[247] Lee YJ, Mangasarian OL, 2001. SSVM: A Smooth Support Vector Machine for Classification. Comput Optim Appl, 20 (1): 5–22.

[248] Furey TS, Cristianini N, Duffy N, David W, 2000. Microarray Expression Data. Bioinformatics, 16 (10): 906–914.

[249] Lu SX, Liu XH, Zhai JH, 2007. A New Fuzzy Multicategory Support Vector Machines Classifier. 2007 Int. Conf. Mach. Learn. Cybern, 19-22 Aug. 2007, Hong Kong, China, s: 2859–2862.

[250] Huang YM, Du SX, 2005. Weighted Support Vector Machine for Classification with Uneven Training Class Sizes. 2005 Int. Conf. Mach. Learn. Cybern, 18-21 Aug. 2005, Guangzhou, China, China, s: 4365–4369.

[251] Shi L, Ma X, Xi L, Hu X, 2010. Financial Data Mining based on Support Vector Machines and Ensemble Learning. 2010 Int. Conf. Intell. Comput. Technol. Autom (ICICTA), 11-12 May 2010, Changsha, China, s: 313–314.

[252] Liu H, Chen SM, 2018. Multi-Level Fusion of Classifiers Through Fuzzy Ensemble Learning. 2018 11th Int. Symp. Comput. Intell. Des, 8-9 Dec. 2018, Hangzhou, China, China, s: 19–22.

[253] Sewell M, 2011. Ensemble Learning. Res Note, 11 (02): 1–12.

[254] Brown G, 2010. Ensemble Learning. Encycl. Mach. Learn. Springer Press, United Kingdom, s: 1–24.

[255] Koçyi G, Yaslan Y, 2016. Görüntü Sınıflandırması için Sözlük Topluluğu Tabanlı Çoklu Örnekli Aktif Öğrenme Metodu. 2016 24th Signal Process. Commun. Appl. Conf, 16-19 May 2016, Zonguldak, Turkey, s: 1221–1224.

[256] Lee S, Amgad M, Masoud M, 2019. An Ensemble-based Active Learning for Breast Cancer Classification. 2019 IEEE Int. Conf. Bioinforma. Biomed, 18-21 Nov. 2019, San Diego, CA, USA, USA, s: 2549–2553.

[257] Na YH, Jo H, Song JB, 2017. Learning to Grasp Objects Based on Ensemble Learning Combining Simulation Data and Real Data. 2017 17th Int. Conf. Control. Autom. Syst (ICCAS), Oct. 18-21, 2017, Ramada Plaza, Jeju, Korea, s: 1030–1034.

[258] Hu X, Zhang R, 2013. Clustering-Based Subset Ensemble Learning Method for Imbalanced Data. 2013 Int. Conf. Mach. Learn. Cybern, 14-17 July 2013, Tianjin, China, s: 35–39.

[259] Tuysuzoglu G, Moarref N, Yaslan Y, 2016. Ensemble Based Classifiers Using Dictionary Learning. 2016 Int. Conf. Syst. Signals Image Process, 23-25 May 2016, Bratislava, Slovakia, s: 1–4.

[260] Deng L, Platt JC, 2014. Ensemble Deep Learning for Speech Recognition. Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, 14-18 September 2014, Singapore, s: 1915–1919.

[261] Su L, Liao H, Yu Z, Zhao Q, 2009. Ensemble Learning for Question Classification. Proc. - 2009 IEEE Int. Conf. Intell. Comput. Intell. Syst (ICIS), 20-22 Nov. 2009,Shanghai, China, s: 501–505.

[262] Zhou ZH, 1990. Ensemble Learning. Encycl biometrics, 1 : 270–273.

[263] Webb GI, Zheng Z, 2004. Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. IEEE Trans Knowl Data Eng, 16 (8): 980–991.

[264] Huang F, Xie G, Xiao R, 2009. Research on Ensemble Learning. 2009 Int. Conf. Artif. Intell. Comput. Intell (AICI), 7-8 Nov. 2009, Shanghai, China, s: 249–252.

[265] Zhou G, Guo F, 2019. Research on Sampling Diversity Method in Ensemble Learning Base on Margin. 2019 Int. Conf. Mach. Learn. Big Data Bus. Intell, 8-10 Nov. 2019,Taiyuan, China, China, s: 316–319.

[266] Golovko V, Kachurka P, Vaitsekhovich L, 2007. Neural Network Ensembles for Intrusion Detection. 2007 4th IEEE Work. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl, 6-8 Sept. 2007,Dortmund, Germany, s: 578–583.

[267] Samat A, Du P, Liu S, Li J, Cheng L, 2014. E2LMs: Ensemble Extreme Learning Machines for Hyperspectral Image Classification. IEEE J Sel Top Appl Earth Obs Remote Sens, 7 (4): 1060–1069.

[268] Gurram P, Kwon H, 2010. A Full Diagonal Bandwidth Gaussian Kernel SVM Based Ensemble Learning for Hyperspectral Chemical Plume Detection. 2010 IEEE Int. Geosci. Remote Sens. Symp, 25-30 July 2010, Honolulu, HI, USA, s: 2804–2807.

[269] Özgür A, Erdem H, 2012. Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması Comparison of Out-of-Box Machine Learning Algorithms used in Intrusion Detection Systems. Bilişim Teknol Derg, 5 (2): 41–48.

[270] Zhou ZH, Jiang Y, Yang YB, Chen SF, 2002. Lung Cancer Cell Identification Based on Artificial Neural Network Ensembles. Artif Intell Med, 24 (1): 25–36.

[271] Lu YC, Lu CJ, Chang CC, Lin YW, 2017. A Hybrid of Data Mining and Ensemble Learning Forecasting for Recurrent Ovarian Cancer. 2017 Int. Conf. Intell. Informatics Biomed. Sci, 24-26 Nov. 2017, Okinawa, Japan, s: 216–216.

[272] Kim HC, Ghahramani Z, 2012. Bayesian Classifier Combination. 2012 Proc. 15th Int. Conference Artif. Intell. Stat, 21-23 April 2012, La Palma, Canary Islands, Spain, s: 619–627.

[273] Yu Y, Zhong Liang F, Xiang Hui Z, Wen Fang C, 2009. Combining Classifier Based on

Decision Tree. 2009 WASE Int. Conf. Inf. Eng, 10-11 July 2009, Taiyuan, Chanxi, China, s: 37–40.

[274] Yu Yan J, 2010. Selective Ensemble Learning Algorithm. 2010 Int. Conf. Electr. Control Eng, 25-27 June 2010, Wuhan, China, s: 1859–1862.

[275] Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M, 2017. Ensemble learning for data stream analysis: A survey. Inf Fusion, 37 : 132–156.

[276] Bulut F, 2017. Örnek Tabanli Siniflandirici Topluluklariyla Yeni Bir Klinik Karar Destek Sistemi. J Fac Eng Archit Gazi Univ, 32 (1): 65–76.

[277] Maclin R, Opitz D, 1997. An Empirical Evaluation of Bagging and Boosting. Fourteenth Natl. Conf. Artificial Intell, 1997, Providence, Rhode Island, s: 546–551.

[278] Opitz DW, MacLin RF, 1997. An Empirical Evaluation of Bagging and Boosting for Artificial Neural Networks. Proc. Int. Conf. Neural Networks, 12-12 June 1997, Houston, TX, USA, USA, s: 1401–1405.

[279] Liang K, Zhou Z, 2012. Using an Ensemble Classifier on Learning Evaluation for E-Learning System. 2012 Int. Conf. Comput. Sci. Serv. Syst, 11-13 Aug. 2012,Nanjing, China, s: 538–541.

[280] Rokach L, 2010. Ensemble-Based Classifiers. Artif Intell Rev, 33 (1–2): 1–39.

[281] Sahinturk H, Ankara N, 2019. Performance Evaluation of Ensemble Learning Algorithms on Unbalanced Credit Scoring Data Sets. Pressacademia, 9 (9): 180–185.

[282] Freund Y, Schapire RE, 1999. A Short Introduction to Boosting. Journal-Japanese Soc Artif Intell, 14 (5): 771–780.

[283] Chen J, 2012. Scalable Ensemble Learning by Adaptive Sampling. 2012 11th Int. Conf. Mach. Learn. Appl, 12-15 Dec. 2012, Boca Raton, FL, USA, s: 622–625.

[284] Xie H, Shang F, 2014. The Study of Methods for Post-pruning Decision Trees Based on Comprehensive Evaluation Standard. 2014 11th Int. Conf. Fuzzy Syst. Knowl. Discov (FSKD), 19-21 Aug. 2014, Xiamen, China, s: 903–908.

[285] Abdelhalim A, Traore I, 2009. A New Method for Learning Decision Trees from Rules. 8th Int. Conf. Mach. Learn. Appl. (ICMLA), 13-15 Dec. 2009, Miami Beach, FL, USA, s: 693–698.

[286] Patil S, Kulkarni U, 2019. Accuracy Prediction for Distributed Decision Tree Using Machine Learning Approach. Proc. Int. Conf. Trends Electron. Informatics (ICOEI), 23-25 April 2019, Tirunelveli, India, India, s: 1365–1371.

[287] Sun J, Wang XZ, 2005. An Initial Comparison on Noise Resisting between Crisp and Decision Trees. 2005 Int. Conf. Mach. Learn. Cybern (ICMLC), 18-21 Aug. 2005,

Guangzhou, China, China, s: 2545–2550.

[288] Liu RZ, Fang B, Luo HW, 2016. Automatic Decision Support by Rule Exhaustion Decision Tree Algorithm. 2016 Int. Conf. Wavelet Anal. Pattern Recognit, 10-13 July 2016, Jeju, South Korea, s: 25–30.

[289] Li Y, Dong M, Kothari R, 2005. Classifiability-Based Omnivariate Decision Trees. IEEE Trans Neural Networks, 16 (6): 1547–1560.

[290] Gehrke J, Ganti V, Ramakrishnan R, Loh W-Y, 1999. BOAT--Optimistic Decision Tree Construction. Proc. 1999 ACM SIGMOD Int. Conf. Manag. data - SIGMOD '99. ACM Press, June 1999, New York, New York, USA, s: 169–180.

[291] Gavankar SS, Sawarkar SD, 2017. Eager Decision Tree. 2017 2nd Int. Conf. Converg. Technol (I2CT), 7-9 April 2017, Mumbai, India, s: 834–840.

[292] Li F, Yang B, Li YY, 2009. Research on Evidence Theory Decision Tree Adaptive Website. 2009 Chinese Control Decis. Conf (CCDC), 17-19 June 2009, Guilin, China, s: 2237–2240.

[293] Utgoff PE, Berkman NC, Clouse JA, 1997. Decision Tree Induction Based on Efficient Tree Restructuring. Mach Learn, 29 (1): 5–44.

[294] Kohavi R, 1996. Scaling Up the Accuracy of Naive-Bayes Classifers: A Decision Tree Hybrid. 202-207, in: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (eds: Simoudis E, Han J, Fayyad U). AAAI Press, Portland.

[295] Du W, Zhan Z, 2002. Building Decision Tree Classifier on Private Data. 1-8, in: CRPIT '14: Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14 (eds: Clifton C). Australian Computer Society, Inc Press, Australia.

[296] Friedl M., Brodley C., 1997. Decision Tree Classification of Land Cover from Remotely Sensed Data. Remote Sens Environ, 61 (3): 399–409.

[297] Du H, Ma C, 2009. Study on Constructing Generalized Decision Tree by Using DNA Coding Genetic Algorithm. 2009 Int. Conf. Web Inf. Syst. Mining (WISM), 7-8 Nov. 2009, Shanghai, China, s: 163–167.

[298] Lee JH, Lee JH, Sohn SG, Ryu JH, Chung TM, 2008. Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System. 2008 10th Int. Conf. Adv. Commun. Technol, 17-20 Feb. 2008, Gangwon-Do, South Korea, s: 1170–1175.

[299] Liu R, Qian XL, Mao S, Zhu SZ, 2011. Research on Anti-Money Laundering Based on Core Decision Tree Algorithm. Proc. 2011 Chinese Control Decis. Conf (CCDC), 23-25 May 2011, Mianyang, China, s: 4322–4325.

[300] Shamim A, Hussain H, Shaikh MU, 2010. A Framework for Generation of Rules from

Decision Tree and Decision Table. 2010 Int. Conf. Inf. Emerg. Technol (ICIET), 14-16 June 2010, Karachi, Pakistan, s: 1–6.

[301] Patil D V., Bichkar RS, 2006. A Hybrid Evolutionary Approach To Construct Optimal Decision Trees With Large Data Sets. Proc. IEEE Int. Conf. Ind. Technol, 15-17 Dec. 2006, Mumbai, India, s: 429–433.

[302] Mingers J, 1989. An Empirical Comparison of Pruning Methods for Decision Tree Induction. Mach Learn, 4 (2): 227–243.

[303] Mingers J, 1989. An Empirical Comparison of Selection Measures for Decision-Tree Induction. Mach Learn, 3 (4): 319–342.

[304] Azad M, Moshkov M, 2014. Minimization of Decision Tree Depth for Multi-Label Decision Tables. Procedia Comput. Sci, 22-24 Oct. 2014, Noboribetsu, Japan, s: 368–377.

[305] Mrva J, Neupauer S, Hudec L, Sevcech J, Kapec P, 2019. Decision Support in Medical Data Using 3D Decision Tree Visualisation. 2019 E-Health Bioeng. Conf, 21-23 Nov. 2019, Iasi, Romania, Romania, s: 1–4.

[306] Safavian SR, Landgrebe D, 1991. A Survey of Decision Tree Classifier Methodology. IEEE Trans Syst Man Cybern, 21 (3): 660–674.

[307] Demirel Ş, 2019. Karar Ağacı Algoritmaları ve Çocuk İşçiliği Üzerine bir Uygulama. Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal bilimler Enstitüsü, İstanbul.

[308] Amin RK, Indwiarti, Sibaroni Y, 2015. Implementation of Decision Tree Using C4.5 Algorithm in Decision Making of Loan Application by Debtor (Case Study: Bank Pasar of Yogyakarta Special Region). 2015 3rd Int. Conf. Inf. Commun. Technol, 27-29 May 2015, Nusa Dua, Bali, s: 75–80.

[309] Fayyad UM, Irani KB, 1992. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. Mach Learn, 8 (1): 87–102.

[310] Çalış A, Kayapınar S, Çetinyokuş T, 2014. Veri Madenciliğinde Karar Ağacı Algoritmaları ile Bilgisayar ve İnternet Güvenliği Üzerine Bir Uygulama. Dergipark Endüstri Mühendisliği Derg, 25 (3–4): 2–19.

[311] Patil S, Kulkarni U, 2019. Accuracy Prediction for Distributed Decision Tree using Machine Learning Approach. 2019 3rd Int. Conf. Trends Electron. Informatics, 23-25 April 2019, Tirunelveli, India, India, s: 1365–1371.

[312] Araujo N, de Oliveira R, Ferreira E, Shinoda AA, Bhargava B, 2010. Identifying Important Characteristics in the KDD99 Intrusion Detection Dataset by Feature Selection using a Hybrid Approach. 2010 17th Int. Conf. Telecommun, 4-7 April 2010, Doha, Qatar, s: 552–558.

[313] Olusola AA, Oladele AS, Abosede DO, 2010. Analysis of KDD'99 Intrusion Detection Dataset for Selection of Relevance Features. Proc. World Congr. Eng. Comput. Sci, 20-22 October 2010, San Francisco, USA, s: 162–168.

[314] Yang HT, 2010. Research on Cost Decision of Specialized-Automobile Manufacturing Enterprise Based on the Theory of Decision Tree. Proc. - 2010 Int. Conf. Digit. Manuf. Autom (ICDMA), 18-20 Dec. 2010, Changsha, China, s: 198–203.

[315] Yazıcı B, Yaslı F, Gürleyik HY, Turgut UO, Aktas MS, Kalıpsız O, Veri Madenciliğinde Özellik Seçim Tekniklerinin Bankacılık Verisine Uygulanması Üzerine Araştırma ve Karşılaştırmalı Uygulama. 9. Ulus. Yazılım Mühendisliği Sempozyumu, 15-17 Eylül 2015, Türkiye, s: 72–83.

[316] Şengöz N, Özdemir G, 2016. Temel Bileşenler Analizi Ve K-Ortalama Kümeleme Yönteminin Birlikte Kullanımı: Bir Örnek Uygulama. Mehmet Akif Ersoy Üniversitesi Sos Bilim Enstitüsü Derg, 8 (15): 85–94.

[317] Zhang X, Zhang X, Ren X, 2011. Two Dimensional Principal Component Analysis based Independent Component Analysis for face recognition. 2011 Int. Conf. Multimed. Technol, 26-28 July 2011, Hangzhou, China, s: 934–936.

[318] Hung JW, Wang HM, Lee LS, 2000. Automatic Metric-based Speech Segmentation for Broadcast News via Principal Component Analysis. 6th Int. Conf. Spok. Lang. Process (ICSLP), 16-20 October 2000, Beijing, China, s: 121–124.

[319] Nethu B, 2012. Classification of Intrusion Detection Dataset using Machine Learning Approaches. Int J Electron Comput Sci Eng, 1 (3): 1044–1051.

[320] Ersungur ŞM, Kızıltan A, Polat Ö, 2007. Türkiye'de Bölgelerin Sosyo-Ekonomik Gelişmişlik Sıralaması:Temel Bileşen Analizi. Atatürk Üniversitesi İktisadi ve İdari Bilim Derg, 21 (2): 55–66.

[321] Yaycılı AÖ, 2006. Temel Bileşen Analizi için Robust Algoritmaları. Yüksek Lisans Tezi, Gazi Üniversitesi Fen bilimleri Enstitüsü, Ankara.

[322] Yang TN, Wang SD, 1999. Robust Algorithms for Principal Component Analysis. Pattern Recognit Lett, 20 (9): 927–933.

[323] Lin J, Zhang Q, Sheng G, Yan Y, Jiang X, 2018. Prediction System for Dynamic Transmission Line Load Capacity Based on PCA and Online Sequential Extreme Learning Machine. 2018 IEEE Int. Conf. Ind. Technol, 20-22 Feb. 2018, Lyon, France, s: 1714–1717.

[324] Zou H, Hastie T, Tibshirani R, 2006. Sparse Principal Component Analysis. J Comput Graph Stat, 15 (2): 265–286.

[325] Ringnér M, 2008. What is Principal Component Analysis? Nat Biotechnol, 26 (3): 303–

304.

[326] Ding C, He X, 2004. K-Means Clustering via Principal Component Analysis. https://dl.acm.org/doi/pdf/10.1145/1015330.1015408 (Erişim Tarihi: 18/05/2020).

[327] Tipping ME, Bishop CM, 1999. Probabilistic Principal Component Analysis. J R Stat Soc Ser B Stat Methodol, 61 (3): 611–622.

[328] Liu WM, Chang CI, 2007. Variants of Principal Components Analysis. 2007 IEEE Int. Geosci. Remote Sens. Symp, 23-28 July 2007, Barcelona, Spain, s: 1083–1086.

[329] Liu Q, Cheng Y, 2014. Bearing Fault Diagnosis Based on PCA and SVM. Vibroengineering Procedia, 4 (1): 206–210.

[330] Ran He, Bao-Gang Hu, Wei-Shi Zheng, Xiang-Wei Kong, 2011. Robust Principal Component Analysis Based on Maximum Correntropy Criterion. IEEE Trans Image Process, 20 (6): 1485–1494.

[331] Yazar I, Yavuz HS, Çay MA, 2009. Temel Bileşen Analizi Yönteminin ve Bazı Klasik ve Robust Uygulamalarının Yüz tanıma Uygulamaları. Eskişehir Osmangazi Üniversitesi Mühendislik Mimar Fakültesi Derg CiltXXII, XXII (1): 49–63.

[332] Abdi H, Williams LJ, 2010. Principal Component Analysis. Wiley Interdiscip Rev Comput Stat, 2 (4): 433–459.

[333] Jinhu L, Xuemei L, Lixing D, Liangzhong J, 2010. Applying Principal Component Analysis and Weighted Support Vector Machine in Building Cooling Load Forecasting. 2010 Int. Conf. Comput. Commun. Technol. Agric. Eng, 12-13 June 2010, Chengdu, China, s: 434–437.

[334] Kim KI, Jung K, Kim HJ, 2016. Face Recognition Using Kernel Principal Component Analysis. EEE signal Process Lett, 9 (2): 40–42.

[335] Pei Y, 2015. Linear Principal Component Discriminant Analysis. 2015 IEEE Int. Conf. Syst. Man, Cybern, 9-12 Oct. 2015, Kowloon, China, s: 2108–2113.

[336] Shlens J, 2014. A Tutorial on Principal Component Analysis. arXiv Prepr arXiv14041100, : 1–12.

[337] Wang S, Ye J, Ying D, 2013. Research of 2DPCA Principal Component Uncertainty in Face Recognition. 2013 8th Int. Conf. Comput. Sci. Educ, 26-28 April 2013, Colombo, Sri Lanka, s: 159–162.

[338] Zhou Y, Cao S, Wen D, Zhang H, Zhao L, 2011. The Study Of Face Recognition Based On Hybrid Principal Components Analysis and Independent Component Analysis. 2011 Int. Conf. Electron. Commun. Control, 9-11 Sept. 2011, Ningbo, China, s: 2964–2966.

[339] Li B, 2018. A Principal Component Analysis Approach to Noise Removal for Speech

Denoising. 2018 Int. Conf. Virtual Real. Intell. Syst, 10-11 Aug. 2018, Changsha, China, s: 429–432.

[340] Cui J, Li G, Yu M, Jiang L, Lin Z, 2019. Aero-Engine Fault Diagnosis Based on Kernel Principal Component Analysis and Wavelet Neural Network. 2019 Chinese Control Decis. Conf, 3-5 June 2019, Nanchang, China, China, s: 451–456.

[341] Muhammed HH, Ammenberg P, Bengtsson E, 2001. Using Feature-Vector Based Analysis, based on Principal Component Analysis and Independent Component Analysis, for Analysing Hyperspectral Images. Proc. 11th Int. Conf. Image Anal. Process. IEEE Comput. Soc, 26-28 Sept. 2001, Palermo, Italy, s: 309–315.

[342] Saxena H, Richariya V, 2014. Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain. Int J Comput Appl, 98 (6): 25–29.

[343] Sahu SK, Sarangi S, Jena SK, 2014. A Detail Analysis on Intrusion Detection Datasets. 2014 IEEE Int. Adv. Comput. Conf, 21-22 Feb. 2014, Gurgaon, India, s: 1348–1353.

[344] Meena G, Choudhary RR, 2017. A Review Paper on IDS Classification Using KDD 99 and NSL KDD Dataset in WEKA. 2017 Int. Conf. Comput. Commun. Electron, 1-2 July 2017, Jaipur, India, s: 553–558.

[345] Güven EN, 2007. Zeki Saldırı Tespit Sistemlerinin İncelenmesi,Tasarımı ve Gerçekleştirilmesi. Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü,Ankara.

[346] Özgür A, Erdem H, 2016. A Review of KDD99 Dataset Usage in Intrusion Detection and Machine Learning between 2010 and 2015. PeerJ Prepr, 4 : 1–21.

[347] Kandeeban S, Rajesh R, 2011. A Genetic Algorithm Based elucidation for improving Intrusion Detection through condensed feature set by KDD 99 data set. Inf Knowl Manag, 1 (1): 1–9.

[348] Anonim, 1999. Kddcup1999. http://kdd.ics.uci.edu/databases/kddcup99/task.html (Erişim Tarihi: 09/04/2020).

[349] Moustafa N, Slay J, 2017. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. 2015 4th Int. Work. Build. Anal. Datasets Gather. Exp. Returns Secur, 5-5 Nov. 2015, Kyoto, Japan, s :25–31.

[350] Deshmukh DH, Ghorpade T, Padiya P, 2014. Intrusion Detection Systemby Improved Preprocessing Methods and Naive Bayes Classifier using NSL-KDD 99 Dataset. 2014 Int. Conf. Electron. Commun. Syst, 13-14 Feb. 2014, Coimbatore, India, s: 1–7.

[351] Janarthanan T, Zargari S, 2017. Feature Selection in UNSW-NB15 and KDDCUP'99 Datasets. 2017 IEEE 26th Int. Symp. Ind. Electron, 19-21 June 2017, Edinburgh, UK, s: 1881–1886.

[352] Anonim, 1999. The UCI KDD Archive Information and Computer Science University of California, Irvine. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Erişim Tarihi: 24/02/2020).

[353] Singh P, Tiwari A, 2015. An Efficient Approach for Intrusion Detection in Reduced Features of KDD99 using ID3 and Classification with KNNGA. 2015 Second Int. Conf. Adv. Comput. Commun. Eng, 1-2 May 2015, Dehradun, India, s: 445–452.

[354] Levin I, 2000. KDD-99 Classifier Learning Contest LLSoft's Results Overview. ACM SIGKDD Explor Newsl, 1 (2): 67–75.

[355] Eldos T, Siddiqui M, Kanan A, 2012. On the KDD'99 Dataset: Statistical Analysis for Feature Selection. J Data Min Knowl Discov, 3 (3): 88–90.

[356] Ingre B, Yadav A, 2015. Performance Analysis of NSL-KDD Dataset using ANN. 2015 Int. Conf. Signal Process. Commun. Eng. Syst, 2-3 Jan. 2015, Guntur, India, s: 92–96.

[357] Kunhare N, Tiwari R, 2018. Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques. 2018 8th Int. Conf. Commun. Syst. Netw. Technol, 24-26 Nov. 2018, Bhopal, India, India, s: 127–131.

[358] Gunes Kayacik H, Nur Zincir-Heywood A, Heywood MI, 2007. A Hierarchical SOM-Based Intrusion Detection System. Eng Appl Artif Intell, 20 (4): 439–451.

[359] Kaya Ç, Yıldız O, 2014. Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz. Marmara Univ J Sci, 26 (3): 89–104.

[360] Kendall K, 1999. A Database of Computer Attacks for the Evaluation of Intrusion Systems. Master's Thesis, Bchelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institüte of Technology ,Cambridge, USA.

[361] Peng H, Long F, Ding C, 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Trans Pattern Anal Mach Intell, 27 (8): 1226–1238.

[362] Gulgezen G, Cataltepe Z, Yu L, 2009. MRMR Algoritması Kullanılarak Kararlı Öznitelik Seçimi. Signal Process Commun Appl Conf, : 596–599.

[363] Radovic M, Ghalwash M, Filipovic N, Obradovic Z, 2017. Minimum Redundancy Maximum Relevance Feature Selection Approach for Temporal Gene Expression Data. BMC Bioinformatics, 18 (1): 1–14.

[364] Shirzad MB, Keyvanpour MR, 2015. A Feature Selection Method based on Minimum Redundancy Maximum Relevance for Learning to Rank. 2015 AI Robot, 12-12 April 2015, Qazvin, Iran, s: 1–5.

[365] Tunç A, Ülger İ, 2016. Veri Madenciliği Uygulamalarında Özellik Seçimi İçin Finansal Değerlere Binning ve Five Number Summary Metot ları ile Normaliza syon İşleminin Uygulanması. XVIII. Akad. Bilişim Konf. Adnan Menderes Üniversitesi, 30 ocak-5 Şubat 2016, Aydın, s: 1–8.

[366] Sakar CO, Kursun O, Gurgen F, 2012. A Feature Selection Method based on Kernel Canonical Correlation Analysis and the Minimum Redundancy-Maximum Relevance Filter Method. Expert Syst Appl, 39 (3): 3432–3437.

[367] El Akadi A, Amine A, El Ouardighi A, Aboutajdine D, 2009. A New Gene Selection Approach Based on Minimum Redundancy-Maximum Relevance (MRMR) and Genetic Algorithm (GA). 2009 IEEE/ACS Int. Conf. Comput. Syst. Appl, 10-13 May 2009, Rabat, Morocco, s: 69–75.

[368] Kurşun O, Şakar CO, Favorov O, Aydin N, Gürgen F, 2010. Using Covariates for İmproving the Minimum Redundancy Maximum Relevance Feature Selection Method. Turkish J Electr Eng Comput Sci, 18 (6): 975–987.

[369] Çelik C, Bilge HŞ, 2015. Ağırlıklandırılmış Koşulu Karşılıklı Bilgi ile Öznitelik Seçimi. Gazi Üniversitesi Mühendislik Mimar Fakültesi Derg, 30 (4): 585–596.

[370] Vinh LT, Thang ND, Lee Y-K, 2010. An Improved Maximum Relevance and Minimum Redundancy Feature Selection Algorithm Based on Normalized Mutual Information. 2010 10th IEEE/IPSJ Int. Symp. Appl. Internet, 19-23 July 2010, Seoul, South Korea, s: 395–398.

[371] Kamandar M, Ghassemian H, 2011. Maximum Relevance, Minimum Redundancy Band Selection for Hyperspectral Images. 2011 19th Iran. Conf. Electr. Eng, 17-19 May 2011, Tehran, Iran, s: 1–5.

[372] Yang J, Shen A, Yu K, Chen Y, 2019. Predicting the Semantic Characteristics of Pulmonary Nodules using Feature Selection Based on Maximum-Relevance Minimum-Redundancy. 2019 IEEE Int. Conf. Bioinforma. Biomed, 18-21 Nov. 2019, San Diego, CA, USA, USA, s: 1318–1323.

[373] Zhao Z, Anand R, Wang M, 2019. Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform. 2019 IEEE Int. Conf. Data Sci. Adv. Anal, 5-8 October 2019, Washington DC, USA, s: 442–452.

[374] Hejazi MI, Cai X, 2009. Input Variable Selection for Water Resources Systems using a modified Minimum Redundancy Maximum Relevance (MMRMR) algorithm. Adv Water Resour, 32 (4): 582–593.

[375] Oufaida H, Nouali O, Blache P, 2014. Minimum Redundancy and Maximum Relevance for

Single and Multi-Document Arabic Text Summarization. J King Saud Univ - Comput Inf Sci, 26 (4): 450–461.

[376] Ma X, Guo J, Sun X, 2015. Sequence-Based Prediction of RNA-Binding Proteins Using Random Forest with Minimum Redundancy Maximum Relevance Feature Selection. http://downloads.hindawi.com/journals/bmri/2015/425810.pdf (Erişim Tarihi: 20/05/2020).

[377] Gao YF, Li BQ, Cai YD, Feng KY, Li ZD, Jiang Y, 2013. Prediction of Active Sites of Enzymes by Maximum Relevance Minimum Redundancy (mRMR) Feature Selection. Mol Biosyst, 9 (1): 61–69.

[378] Unler A, Murat A, Chinnam RB, 2011. MR2PSO: A Maximum Relevance Minimum Redundancy Feature Selection Method based on Swarm Intelligence for Support Vector Machine Classification. Inf Sci (Ny), 181 (20): 4625–4641.

[379] Ramirez Gallego S, Lastra I, Martinez Rego D, Bolon Canedo V, Benitez JM, Herrera F, Alonso Betanzos A, 2017. Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data. Int J Intell Syst, 32 (2): 134–152.

[380] Gülgezen G, 2009. Kararlı ve Başarımı Yüksek Öznitelik Seçimi. Yüksek Lisans Tezi, İstanbul teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[381] Doewes A, Swasono SE, Harjito B, 2017. Feature Selection on Human Activity Recognition Dataset using Minimum Redundancy Maximum Relevance. 2017 IEEE Int. Conf. Consum. Electron. - Taiwan, 12-14 June 2017, Taipei, Taiwan, s: 171–172.

[382] Li Z, Zhou X, Dai Z, Zou X, 2010. Classification of G-Protein Coupled Receptors based on Support Vector Machine with Maximum Relevance Minimum Redundancy and Genetic Algorithm. BMC Bioinformatics, 11 (1): 1-15.

[383] Rabiu H, Saripan MI, Mashohor S, Marhaban MH, 2012. 3D Facial Expression Recognition using Maximum Relevance Minimum Redundancy Geometrical Features. EURASIP J Adv Signal Process, 2012 (1): 1–8.

[384] Ding C, Peng H, 2003. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. Comput. Syst. Bioinformatics. CSB2003. Proc. 2003 IEEE Bioinforma. Conf. CSB2003, 11-14 Aug. 2003, Stanford, CA, USA, USA, s: 523–528.

[385] Wang S, Zhang Y-H, Lu J, Cui W, Hu J, Cai Y-D, 2016. Analysis and Identification of Aptamer-Compound Interactions with a Maximum Relevance Minimum Redundancy and Nearest Neighbor Algorithm. http://downloads.hindawi.com/journals/bmri/2016/8351204.pdf (Erişim Tarihi: 29/04/2020).

[386] Huang M, Sun L, Xu J, Zhang S, 2020. Multilabel Feature Selection Using Relief and Minimum Redundancy Maximum Relevance Based on Neighborhood Rough Sets. IEEE Access, 8 : 62011–62031.

[387] Ding C, Peng H, 2005. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. J Bioinform Comput Biol, 3 (2): 185–205.

[388] Acid S, de Campos LM, Fernandez M, 2011. Minimum Redundancy Maximum Relevancy versus Score-based Methods for Learning Markov Boundaries. 2011 11th Int. Conf. Intell. Syst. Des. Appl, 22-24 Nov. 2011, Cordoba, Spain, s: 619–623.

[389] Li X, Zheng Z, Wu L, Li R, Huang J, Hu X, Guo P, 2019. A Stratified Method for Large-Scale Power System Transient Stability Assessment Based on Maximum Relevance Minimum Redundancy Arithmetic. IEEE Access, 7 : 61414–61432.

[390] Mandal M, Mukhopadhyay A, 2013. An Improved Minimum Redundancy Maximum Relevance Approach for Feature Selection in Gene Expression Data. Procedia Technol, 10 : 20–27.

ÖZGEÇMİŞ

1989 yılında Malazgirt'te doğdum. İlköğretim ve ortaokulu Beşçatak İlköğretim Okulu'nda ve liseyi Malazgirt Alparslan Lisesi'nde tamamladım. 2009 yılında kazandığım Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümün'den 2013 yılında mezun oldum. 2017'de Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisansa başladım. Yabancı dilim İngilizce'dir.

Mehmet BURUKANLI