Araştırma Makalesi / Research Article

Saldırı Tespiti için Makine Öğrenme Yöntemlerinin Karşılaştırmalı Analizi

Mehmet BURUKANLI[1], Musa ÇIBUK[2*], Ümit BUDAK[3]

[1]*Bitlis Eren Üniversitesi, Enformatik Bölümü, Bitlis*
[2]*Bitlis Eren Üniversitesi, Bilgisayar Mühendisliği Bölümü, Bitlis*
[3]*Bitlis Eren Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Bitlis*
(ORCID: 0000-0003-4459-0455) (ORCID: 0000-0001-9028-2221) (ORCID: 0000-0003-4082-383X)

Öz

İnternet ve ağ teknolojilerindeki hızlı gelişmeler, siber saldırılar ve izinsiz erişimlerin başta olduğu birçok dezavantajı beraberinde getirmektedir. Bu girişimlerin önceden tespiti, olası saldırıların gerçekleşmeden önlenebilmesini sağlamaktadır. Bu çalışmada makine öğrenme yaklaşımlarının saldırı tespiti üzerindeki performansları araştırılmıştır. Tüm deneyler, açık erişime sunulmuş ve yaygın olarak kullanılan KDD'99 veri kümesi altındaki KDD10CORRECTED ve KDDTEST setleri üzerinde gerçekleştirilmiştir. Sınıflandırıcı olarak, KA, TÖ ve DVM tercih edilmiştir. Veri setleri hem doğrudan sınıflandırıcıların girişi olarak hem de boyut indirgeme tekniği olan TBA uygulanarak sınıflandırılmıştır. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmıştır. En iyi başarım oranları KDD10CORRECTED veri setinde üzerinde %99,99 ile Torbalama sınıflandırıcısı, KDDTEST veri setinde üzerinde %97,90 ile Torbalama sınıflandırıcısı, KDD10CORRECTED+KDDTEST veri setinde üzerinde %100 ile Torbalama sınıflandırıcısı elde etmiştir. Elde edilen sonuçlar kıyaslanarak rapor edilmiştir. Sonuçlar gelecekteki çalışmalar için cesaret vericidir.

Anahtar kelimeler: Saldırı tespiti, topluluk öğrenme, destek vektör makinesi, karar ağacı, temel bileşen analizi.

Comparative Analysis of Machine Learning Methods for Intrusion Detection

Abstract

Rapid developments of internet and network technologies have brought about many disadvantages, including cyber-attacks and intrusions. The detection of these initiatives beforehand provides the prevention of probable attacks. In this study, the performance of machine learning approaches on intrusion detection has investigated. All experiments have conducted on KDD10CORRECTED and KDDTEST sub-sets of the publicly available KDD'99 dataset. As the classifier, Decision Tree, Ensemble Learning and Support Vector Machine have preferred. Data sets are classified as both directly input of the classifiers and by using Principal Component Analysis, which is a size reduction technique. 5-fold cross-validation technique has used in the classification stage. The best performance rates have been achieved above the KDD10CORRECTED data set with the Bagging classifier with 99.99%, above the KDDTEST data set with the Bagging classifier with 97.90%, and above the KDD10CORRECTED + KDDTEST data set with the Bagging classifier with 100%. The results have reported by comparing. The results are encouraging for future studies.

Keywords: Intrusion detection, ensemble learning, support vector machine, decision tree, principal component analysis.

1. Giriş

Son yıllarda internetin yaygın kullanımı, güvenlik tehlikelerini de beraberinde getirmektedir. 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. Örneğin; çalışanların hataları, bilgisayar virüsleri ve

*Sorumlu yazar: mcibuk@beu.edu.tr
Geliş Tarihi: 24.02.2021, Kabul Tarihi: 29.04.2021

hizmet engelleme (denial of service - DoS) saldırıları bunlardan bir kaçıdır. Yapılan saldırılar sonucu önemli bilgi kayıpları olmakta ve gizli kalması gereken bilgiler ifşa edilebilmektedir. İnternetteki güvenlik açıkları, insanları internete karşı güvensiz hale getirmekte ve web tabanlı şirketlere ve kamu hizmetlerine büyük zarar verebilmektedir. Bu yüzden şirketler ve kamu hizmetleri yürüten kurumlar, güvenlik tedbirlerini arttırmakta ve olası tehditlerin üstesinden gelmek amacıyla daha büyük yatırımlar yapmak zorunda kalmaktadırlar. Bundan dolayı, bilgisayar sistemlerinin güvenliğini sağlayan araçlar gittikçe önem kazanmakta, 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 [1]. STS kullanılarak ağ üzerinden yapılan saldırılar tespit edilebilmekte ve ilgili mekanizmalar harekete geçirilerek engellenebilmektedir. STS uygulamalarında makine öğrenmesi, veri madenciliği vb. farklı yöntemler kullanılsa da yapay zekâ tekniklerine dayalı yöntemler de sıkça kullanılmaya başlanmıştır. Literatürde saldırı tespiti için yapılan bazı çalışmaları şöyle sıralayabiliriz.

Burukanlı ve ark. [2] yaptıkları çalışmada makine öğrenme algoritmaları kullanarak saldırı tespiti gerçekleştirmişlerdir. KDD'99 veri setinin %100'lük kısmı üzerinde 5-kat çapraz doğrulama yapmışlardır. En iyi başarımı %100 oranıyla Topluluk Öğrenme (TÖ) sınıflandırıcılarından olan torbalı ağaçlar (bagged trees) algoritması elde etmiştir. Sağıroğlu ve ark [1] yaptıkları çalışmada, Çok Katmalı Ağlar (ÇKA) tabanlı etkili bir STS geliştirmişlerdir. 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. Sonawane ve ark. [3] yaptıkları çalışmada, Yapay Sinir Ağı (YSA) ve Temel Bileşen Analizi (TBA) tabanlı ağ modeli kullanılarak saldırı tespiti gerçekleştirmişlerdir. Ayrıca, her iki algoritmanın kıyaslanması yapılmış olup, YSA'nın TBA'ya göre daha iyi sonuç verdiği gözlemlenmiştir. Aburomman ve ark. [4] yaptıkları çalışmada, TÖ sınıflandırıcısı kullanılarak saldırı tespiti uygulamalarına yeni bir boyut kazandırmışlardır. Çalışmalarında, en iyi başarım oranı %85,17 olarak gerçekleşmiştir. Golovko ve ark. [5] çalışmalarında, TBA, Devirdaim Sinir Ağları ve ÇKA ağ modellerini kullanarak saldırı tespiti yapmışlardır. Model 1, Model 2 ve Model 3 olarak 3 tane model önermişlerdir. Çalışmalarında, en iyi performansı Model 3'ün DoS veri setinde %99,9 olarak gerçekleştiğini gözlemlemişlerdir. Wang ve ark. [6] yaptıkları çalışmada, TBA tabanlı model kullanarak saldırı tespiti gerçekleştirmişlerdir. Çalışmada sonucunda, başarım oranı %98,8 olarak gerçekleşmiştir. Sonawane ve ark. [7] yaptıkları çalışmada, TBA, Çekirdek TBA+Bayes tabanlı sinir ağ modelleri kullanarak saldırı tespiti gerçekleştirmişlerdir. TBA sinir ağ modeli ile Çekirdek TBA+Bayes Sinir Ağ Modelleri birbirleriyle kıyaslamışlardır. En iyi performansı %92,70 ile Çekirdek TBA+Bayes Sinir Ağ Modeli elde etmiştir. Mukkamala ve ark. [8] yaptıkları çalışmada, sinir ağlar ve destek vektör makinelerini kullanarak saldırı tespitini gerçekleştirmişlerdir. Her iki öğrenme algoritması %99'un üzerinde başarım elde etmiştir.

Bu çalışmada makine öğrenme yaklaşımlarının saldırı tespiti üzerindeki performansları araştırılmış olup. Tüm deneyler, açık erişime sunulmuş ve yaygın olarak kullanılan KDD'99 veri kümesi üzerinde gerçekleştirilmiştir. Sınıflandırıcı olarak, KA, TÖ ve DVM tercih edilmiştir. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmış olup, TÖ sınıflandırıcıların daha iyi başarım elde ettiği gözlemlenmiştir.

2. Materyal ve Metot

2.1. KDD'99 veri seti

DARPA veri seti, ilk olarak Amerika Birleşik Devletlerinin Hava Kuvvetleri Ağı örnek alınarak tasarlanmış bir benzetim veri seti kümesidir [9]. DARPA tarafından desteklenen ilk çalışma Massachusetts Teknoloji Üniversitesi (MIT) tarafından 1998 yılında gerçekleştirilmiştir.

DARPA veri seti bir takım ön işlemlerden geçirilerek KDD'99 veri seti elde edilmiştir. KDD'99, 9 temel özellik ve 32 adet türetilmiş özellik olmak üzere toplamda 41 adet özellikten oluşan bir veri setidir [10]. KDD'99 veri setinin kullanılmasının amacı; saldırı tespiti için eğitim ve test işlemleri bakımından kolaylık sağlamasıdır. Bu çalışmada kullanılan KDD10CORRECTED veri seti 494021 örnekten, KDDTEST veri seti 311029 örnekten ve bu iki veri setinin birleşiminden elde edilen KDD10CORRECTED+KDDTEST veri seti 805050 örnekten oluşmaktadır [10].

KDD10CORRECTED, KDDTEST ve KDD10CORRECTED+KDDTEST veri setlerinde bulunan saldırı tiplerinin miktarları ve yüzdelik oranları Tablo 1'de verilmiştir.

Tablo 1. KDD10CORRECTED, KDDTEST ve KDD10CORRECTED+KDDTEST veri setlerinde bulunan saldırı tiplerinin miktarları ve yüzdelik oranları [10, 11]

	KDD10CORRECTED		KDDTEST		KDD10CORRECTED+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	7741	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**

Tablo 2'de KDD'99 veri setinin 41 özelliği ve bu özelliklerin tipleri ve Tablo 3'te ise KDD'99 veri setinin birkaç örneği gösterilmiştir.

Tablo 2. KDD'99 veri setinin 41 özelliği ve bu özelliklerin tipleri [10, 11]

Özellik Adı	Özellik Tipi
duration	Sürekli
protocol_type	Sembolik
service	Sembolik
flag	Sembolik
src_bytes	Sürekli
dst_bytes	Sürekli
land	Sembolik
wrong_fragment	Sürekli
urgent	Sürekli
hot	Sürekli
num_failed_logins	Sürekli
logged_in	Sembolik
num_compromised	Sürekli
root_shell	Sürekli
su_attempted	Sürekli
num_root	Sürekli
num_file_creations	Sürekli
num_shells	Sürekli
num_access_files	Sürekli
num_outbound_cmds	Sürekli
is_host_login	Sembolik
is_guest_login	Sembolik
count	Sürekli
srv_count	Sürekli
serror_rate	Sürekli
srv_serror_rate	Sürekli
rerror_rate	Sürekli
srv_rerror_rate	Sürekli
same_srv_rate	Sürekli
diff_srv_rate	Sürekli
srv_diff_host_rate	Sürekli
dst_host_count	Sürekli
dst_host_srv_count	Sürekli
dst_host_same_srv_rate	Sürekli
dst_host_diff_srv_rate	Sürekli
dst_host_same_src_port_rate	Sürekli
dst_host_srv_diff_host_rate	Sürekli
dst_host_serror_rate	Sürekli
dst_host_srv_serror_rate	Sürekli
dst_host_rerror_rate	Sürekli
dst_host_srv_rerror_rate	Sürekli

Tablo 3. KDD'99 veri setinin birkaç saldırı tipi örneği [10]

Saldırı Tipi	Örnek
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,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,201,1,1.00,1.00,0.00,0.00,0.00,0.06,0.00,255,1,0.00,0.08,0.00,0.00,1.00,1.00,0.00,0.00,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

2.2. Temel Bileşen Analizi

Temel Bileşen Analizi (TBA), bilgisayar bilimlerinde boyut azaltmak için kullanılan bir yöntemdir [12]. TBA, sinyal işleme, görüntü işleme ve yapay zekâ tekniklerinde boyut azaltmak için sıkça kullanılan tekniklerden bir tanesidir [11]. 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 [11, 13].

Bir giriş dizi vektörü $X_1, X_2, X_3, \dots, X_n \in R^n (n < m)$ ve $\sum_{t=1}^{m} X_t = 0$ ise [11, 14], vektörlerin kovaryans matrisi denklem (1)'deki gibi hesaplanır:

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

$$\lambda_t U_t = C U_t \tag{2}$$

Burada; λ_t, U_t, k ve θ, sırasıyla C kovaryasının öz değeri, öz vektörlerin özdeşi, öz vektörlerin en büyüğü ve eşik değeri en büyük k öz vektörünün yaklaşık duyarlılığını ifade eder [11, 14].

$$\sum_{t=1}^{k} \lambda_t \setminus \sum_{t=1}^{m} \lambda_t \geq \theta \tag{3}$$

θ ,eşik değerine bakılarak;

$$U = [\, U_1, U_2, U_3, \dots, U_k] \tag{4}$$

$$\Lambda = [\, \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_k] \tag{5}$$

$$S = U^T X_t \tag{6}$$

Elde edilen yeni boyut vektörü denklem (6)'daki gibi hesaplanır [11, 14].

2.3. Karar Ağacı

Karar Ağaçları (KA), sınıflandırma ve tahmin için sıklıkla başvurulan bir öğrenme yaklaşımıdır. Düşük maliyetli olması, anlaşılması ve yorumlanması kolay olması bakımından sıklıkla tercih edilmektedir.

KA'larında belirsizlik ve kararsızlık önemli bir sorundur bunun üstesinden gelmek için en yaygın olarak kullanılan entropi ölçümü kullanılmaktadır. Entropi ölçümü ne kadar fazla ise çıkan sonuçlar da o oranda belirsiz ve kararsız olmaktadır. Bu yüzden, KA'larında entropi ölçüsü en az olan alanlar tercih edilmektedir [2, 11, 15, 16].

Entropi değeri denklem (7)'deki gibi hesaplanır [11, 15].

$$Entropi(S) = \sum_{i}^{C} -p_i \log p_i \tag{7}$$

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 (8)'deki gibi hesaplanır [11].

$$Kazanç(S, A) = Entropi(S) - \sum_{V \in (A)} \frac{|S_V|}{S}\, Entropi(S_V) \tag{8}$$

Burada; A: Özellik, V: A özelliğinin bir olası değeri, S_V: V değeri için örneklerin sayısı, S: tüm veri örneklerinin sayısıdır [11].

2.4. Topluluk Öğrenme

Topluluk Öğrenme (TÖ), birçok öğrenme algoritmasının bir arada kullanılmasıdır [2, 11]. Diğer bir deyişle; TÖ, birçok zayıf öğrenme algoritmasının birleştirilerek daha güçlü ve daha iyi sonuçlar elde eden bir öğrenme algoritmasının oluşturulması yöntemidir [2, 11]. Tek bir algoritmaya göre daha iyi sonuç vermektedir. TÖ, en iyi sınıflandırma yöntemlerinden biridir [17]. Ayrıca genelleme yetenekleri oldukça güçlüdür. TÖ, sınıflandırma ve regresyon için sıkça kullanılan öğrenme yöntemidir. En yaygın kullanılan TÖ teknikleri Bagging ve Boosting'tır [2, 11].

2.5. Destek Vektör Makinesi

Destek Vektör Makinesi (DVM), sınıflandırma ve regresyon problemlerinin çözümü için sıkça kullanılan bir denetimli öğrenme yöntemidir [2]. DVM, oldukça yüksek genelleme yapabilme yeteneğine sahiptir.

DVM'nin en önemli avantajı yüksek oranda başarılı sonuçlar elde etmeleridir. En önemli dezavantajı ise çok geç sonuç vermeleridir. DVM, margini maksimum yapan bir en uygun ayırıcı düzlemi oluşturmaya çalışmaktadır. Örneğin, bir örnek uzayında eğitim örnekleri denklem (9)'daki formülle birbirinde ayırılabilmektedir [11].

$$f(x) = w.x + b = 0 \tag{9}$$

Doğrusal olmayan durumlarda çekirdek adı verilen fonksiyonlar kullanılmaktadır. Sınıflandırma yapılırken yüksek boyutlu uzaya taşınan vektörler doğrusal olarak ayrılmaktadır. Ayrılan düzlemler içerisinde sınıflara uzaklığı en fazla olan doğrusal ayrıştırıcı olarak belirlenmektedir. Yüzeye en yakın vektörler belirlenerek en yakın uzaklık tespit edilir.

$K = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ 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 [11].
Burada;

$$f(x) = (w_0.x) + b_0 \tag{10}$$

$$w_0 = \sum_{j=1}^{N} y_j \alpha_j^0 x_j \tag{11}$$

$w_0 = (w_0^1, w_0^2, \dots, w_0^n)$ ve $x = (x^1, x^2, \dots, x^n)$ olarak verilmiş olsun. Bu durumda bu iki vektörün iç çarpımı denklem (12)'de gösterilmiştir.

$$(w_0.x) = \sum_{i=1}^{n} w_0^i . x^i \tag{12}$$

$$b_0 = y_i - \left(x_i . \sum_{j=1}^{N} y_j \alpha_j^0 x_j\right) \tag{13}$$

Denklem (11)'deki w_0, denklem (10)'da yerine yazılırsa denklem (14) elde edilir.

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i^0 (x_i . x) + b_0 \quad (14)$$

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

Denklem (15)'teki optimal hiper düzlem karar fonksiyonu kullanarak iki sınıfın birbirinden doğrusal olarak ayrılıp ayrılmadığı kontrol edilebilir[8, 11, 18–21].

$$f(x) = sgn(w . x_i + b) = sgn(\sum_{i=1}^{N} \alpha_i y_i (x_i . x) + b) \quad (15)$$

3. Bulgular ve Tartışma

Bu çalışmada on iki farklı sınıflandırma algoritması kullanılarak KDD10CORRECTED, KDDTEST ve KDD10CORRECTED+KDDTEST veri setleri üzerinde, her bir sınıflandırıcının başarım değerleri elde edilmiştir. KDD10CORRECTED ve KDDTEST veri setleri, 4 çekirdekli Intel Core i5-4590S işlemci, 4 GB Ram ve Intel HD Graphics 4600 bilgisayarda eğitilmiştir.

KDD10CORRECTED+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. Tüm deneyler MATLAB ortamında sınıflandırma öğrenme aracı kullanılarak gerçekleştirilmiştir. Her bir sınıflandırma yönteminin varsayılan özellikleri kullanılmıştır. Tablo 4'te TÖ sınıflandırıcıların temel özellikleri, Tablo 5'te DVM sınıflandırıcıların temel özellikleri ve Tablo 6'da ise KA sınıflandırıcıların temel özellikleri gösterilmiştir.

Tablo 4. TÖ sınıflandırıcıların temel özellikleri

Ana Teknik	Kullanılan Teknik	TBA	Öğrenici Tipi	Maksimum Bölme Sayısı	Öğrenici Sayısı	Öğrenme Oranı
TÖ	Uyarlamalı Güçlendirme	Pasif/ Aktif	Karar Ağacı	20	30	0.1
TÖ	Torbalama	Pasif/ Aktif	Karar Ağacı	-	30	-
TÖ	Rastgele Alt Örnekleme Artırma	Pasif/ Aktif	Karar Ağacı	20	30	0.1

Tablo 5. DVM sınıflandırıcıların temel özellikleri

Ana Teknik	Kullanılan Teknik	TBA	Çekirdek Fonksiyonu	Çekirdek Ölçeği	Kutu Kısıtlama Seviyesi	Çoklu Sınıf Metodu	Verileri Standartlaştırma
DVM	Doğrusal DVM	Pasif/ Aktif	Doğrusal	Otomatik	1	Bire Karşı Bir	Evet
DVM	Karesel DVM	Pasif/ Aktif	Karesel	Otomatik	1	Bire Karşı Bir	Evet
DVM	Kübik DVM	Pasif/ Aktif	Kübik	Otomatik	1	Bire Karşı Bir	Evet
DVM	Hassas Gauss DVM	Pasif/ Aktif	Gauss	1.6	1	Bire Karşı Bir	Evet
DVM	Ortalama Gauss DVM	Pasif/ Aktif	Gauss	6.4	1	Bire Karşı Bir	Evet
DVM	Kaba Gauss DVM	Pasif/ Aktif	Gauss	26	1	Bire Karşı Bir	Evet

Tablo 6. KA sınıflandırıcıların temel özellikleri

Ana Teknik	Kullanılan Teknik	TBA	Maksimum Bölme Sayısı	Bölme Kriteri	Vekil Karar Bölmeleri
KA	Hassas Ağaç	Pasif/Aktif	100	Gini'nin Çeşitlilik İndeksi	Pasif
KA	Ortalama Ağaç	Pasif/Aktif	20	Gini'nin Çeşitlilik İndeksi	Pasif
KA	Kaba Ağaç	Pasif/Aktif	4	Gini'nin Çeşitlilik İndeksi	Pasif

Bu çalışmada kullanılan on iki adet sınıflandırıcının değerlendirme metrikleri için Tablo 7'de hata matrisi kullanılarak elde edilmiştir.

Tablo 7. Hata matrisi [11]

Hata Matrisi		Tahmin Edilen Sınıf	
		Pozitif (Saldırı)	Negatif (Normal)
Gerçek Sınıf	Pozitif (Saldırı)	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif (Normal)	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Doğru Pozitif (DP): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde de pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.
Yanlış Negatif (YN): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde de negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir.
Yanlış Pozitif (YP): Gerçekte negatif (normal) olan ve tahmin edildiğinde de pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.
Doğru Negatif (DN): Gerçekte negatif (normal) olan ve tahmin edildiğinde de negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir. Denklem (16)'da bu çalışmada kullanılan başarım ölçütü verilmiştir [2, 11, 22].

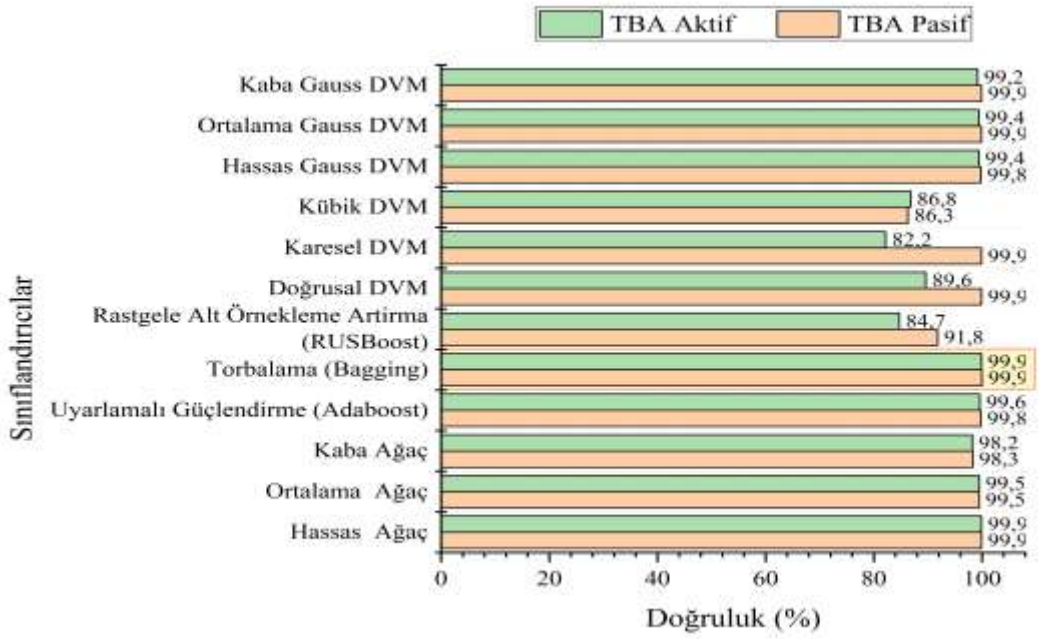
$$Doğruluk\ (Accuracy) = \frac{DP + DN}{DP + YN + YP + DN} \tag{16}$$

Tablo 8'de KDD10CORRECTED veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması, Tablo 9'da KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması ve Tablo 10'da ise KDD10CORRECTED+KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması gösterilmiştir.

Tablo 8. KDD10CORRECTED veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

	TBA Pasif		TBA Aktif	
Sınıflandırıcılar	**Doğruluk**	**Eğitim Süresi (sn)**	**Doğruluk**	**Eğitim Süresi (sn)**
Hassas Ağaç	%99,90	831,58	%99,90	762,08
Ortalama Ağaç	%99,50	1028,9	%99,50	744,09
Kaba Ağaç	%98,30	894,34	%98,20	687,03
Uyarlamalı Güçlendirme (Adaboost)	%99,80	636,4	%99,60	598,14
Torbalama (Bagging)	**%99,99**	**446,68**	**%99,90**	**368,73**
Rastgele Alt Örnekleme Artırma (RUSBoost)	%91,80	399,99	%84,70	359,64
Doğrusal DVM	%99,90	2769	%89,60	12228
Karesel DVM	%99,90	6919,4	%82,20	65404
Kübik DVM	%86,30	62429	%86,80	130300
Hassas Gauss DVM	%99,80	15394	%99,40	1783,3
Ortalama Gauss DVM	%99,90	4177,8	%99,40	1807
Kaba Gauss DVM	%99,90	2917,6	%99,20	2628,5

Yapılan çalışma sonucunda Tablo 8'de göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %99,99 ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %86,30 ile Kübik DVM sınıflandırıcısı elde etmiştir. Öte yandan, TBA aktif durumda iken en iyi performansları TÖ sınıflandırıcısından olan %99,90 ile Torbalama sınıflandırıcısı ile KA sınıflandırıcısından olan %99,90 ile Hassas Ağaç elde etmiştir. Torbalama sınıflandırıcısı ile Hassas Ağaç sınıflandırıcısı aynı başarım oranına sahip olmasına rağmen, eğitim ve test süresi bakımından en iyi sonucu yaklaşık 369 sn ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %82,2 ile Karesel DVM sınıflandırıcısı elde etmiştir. KDD10CORRECTED veri seti üzerinde 12 adet sınıflandırıcının performanslarının karşılaştırılması konunun daha iyi anlaşılması için Şekil 1'de grafiksel olarak da gösterilmiştir.



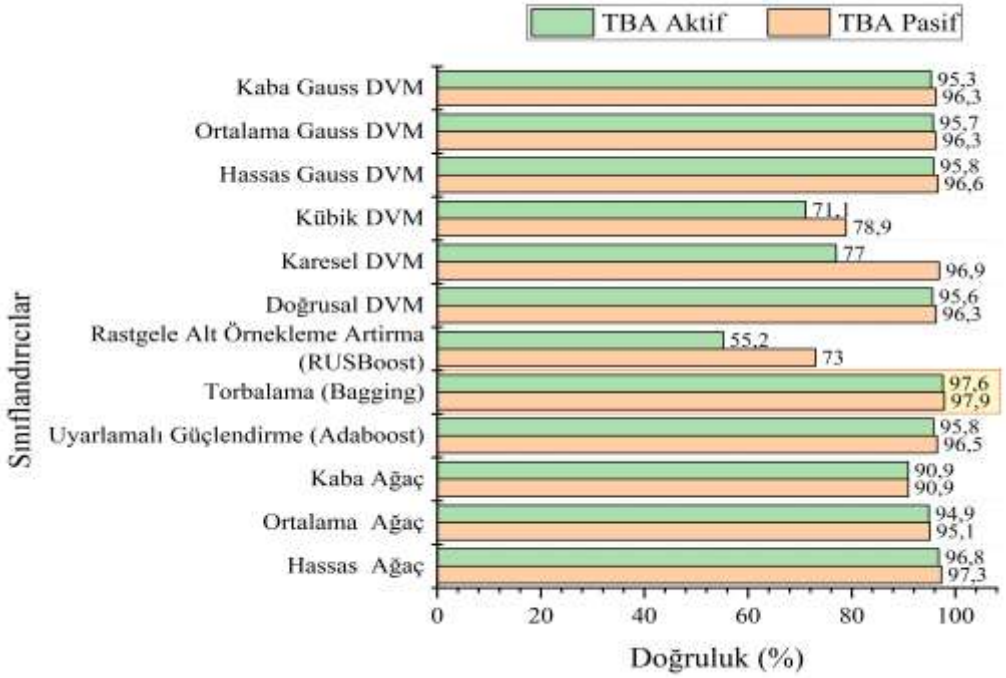
Şekil 1. KDD10CORRECTED veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

Tablo 9. KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

	TBA Pasif		**TBA Aktif**	
Sınıflandırıcılar	**Doğruluk**	**Eğitim Süresi (sn)**	**Doğruluk**	**Eğitim Süresi (sn)**
Hassas Ağaç	%97,30	22,822	%96,80	30,893
Ortalama Ağaç	%95,10	13,347	%94,90	22,018
Kaba Ağaç	%90,90	10,544	%90,90	22,015
Uyarlamalı Güçlendirme (Adaboost)	%96,50	599,6	%95,80	480,98
Torbalama (Bagging)	**%97,90**	**296,73**	**%97,60**	**236,55**
Rastgele Alt Örnekleme Artırma (RUSBoost)	%73,00	376,24	%55,20	367,3
Doğrusal DVM	%96,30	9507,7	%95,60	14826
Karesel DVM	%96,90	16756	%77,00	25586
Kübik DVM	%78,90	125060	%71,10	74122
Hassas Gauss DVM	%96,60	15306	%95,80	5308,9
Ortalama Gauss DVM	%96,30	13022	%95,70	5519,9
Kaba Gauss DVM	%96,30	10493	%95,30	8058

Yapılan çalışma sonucunda Tablo 9'da göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %97,9 ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise TÖ sınıflandırıcısından olan %73 ile Rastgele Alt Örnekleme Artırma

sınıflandırıcısı elde etmiştir. Öte yandan, TBA aktif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %97,60 ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %71,10 ile Kübik DVM sınıflandırıcısı elde etmiştir. KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performanslarının karşılaştırılması konunun daha iyi anlaşılması için Şekil 2'de grafiksel olarak da gösterilmiştir.

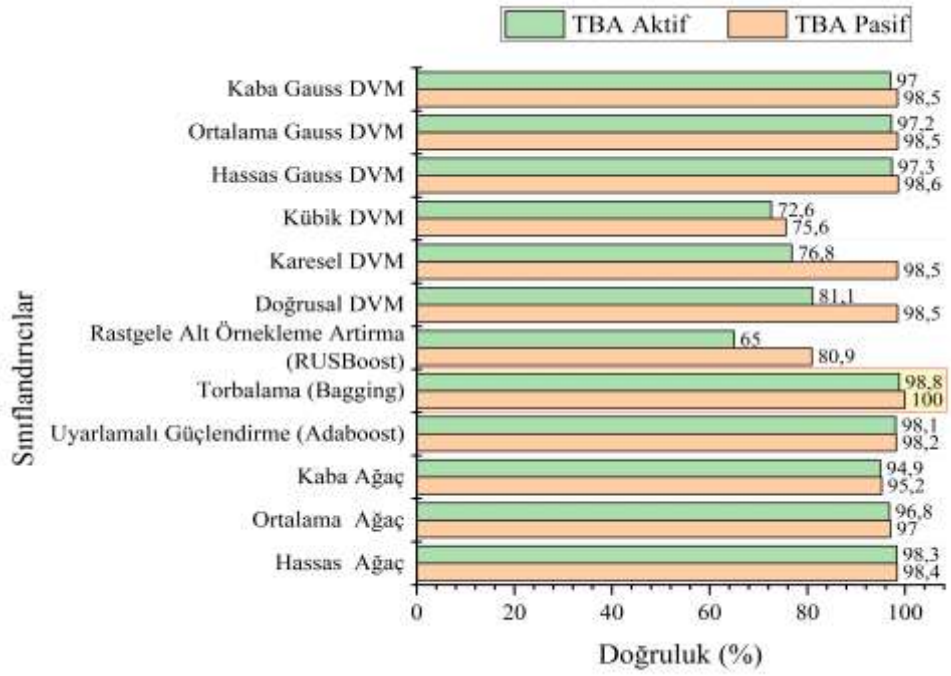


Şekil 2. KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

Tablo 10. KDD10CORRECTED+KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

	TBA Pasif		**TBA Aktif**	
Sınıflandırıcılar	**Doğruluk**	**Eğitim Süresi (sn)**	**Doğruluk**	**Eğitim Süresi (sn)**
Hassas Ağaç	%98,40	62,736	%98,30	100,72
Ortalama Ağaç	%97,00	43,566	%96,80	61,132
Kaba Ağaç	%95,20	37,23	%94,90	75,452
Uyarlamalı Güçlendirme (Adaboost)	%98,20	2080,5	%98,10	1612
Torbalama (Bagging)	**%100,00**	**1191,5**	**%98,80**	**951,76**
Rastgele Alt Örnekleme Artırma (RUSBoost)	%80,90	1298,6	%65,00	1211,1
Doğrusal DVM	%98,50	11720	%81,10	57348
Karesel DVM	%98,50	89777	%76,80	273860
Kübik DVM	%75,60	668140	%72,60	486570
Hassas Gauss DVM	%98,60	75506	%97,30	8522,3
Ortalama Gauss DVM	%98,50	20729	%97,20	8861
Kaba Gauss DVM	%98,50	13795	%97,00	13070

Yapılan çalışma sonucunda Tablo 10'da göründüğü gibi, 12 sınıflandırıcı arasında TBA pasif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %100 ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise DVM sınıflandırıcısından olan %75,60 ile Kübik DVM sınıflandırıcısı elde etmiştir. Öte yandan, TBA aktif durumda iken en iyi performansı TÖ sınıflandırıcısından olan %98,80 ile Torbalama sınıflandırıcısı elde etmiştir. En kötü performansı ise TÖ sınıflandırıcısından olan %65,00 ile Rastgele Alt Örnekleme Artırma sınıflandırıcısı elde etmiştir. KDD10CORRECTED+KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performanslarının karşılaştırılması konunun daha iyi anlaşılması için Şekil 3'de grafiksel olarak da gösterilmiştir.



Şekil 3. KDD10CORRECTED+KDDTEST veri seti üzerinde 12 adet sınıflandırıcının performans karşılaştırılması

4. Sonuç ve Öneriler

Bu çalışmada, günümüzde en çok kullanılan makine öğrenme sınıflandırıcıları kullanılarak saldırı tespiti gerçekleştirilmiştir. Makine öğrenme sınıflandırıcıları olarak; KA, TÖ ve DVM sınıflandırıcıları tercih edilmiştir. Bu üç sınıflandırma tekniği kullanılarak KDD'99 veri kümesi altındaki bulunan; KDD10CORRECTED, KDDTEST ve KDD10CORRECTED+KDDTEST veri seti üzerinde sınıflandırma işlemi gerçekleştirilmiştir. Sınıflandırma aşamasında 5-kat çapraz doğrulama tekniği kullanılmış olup, en iyi başarım oranlarına Tablo 8, Tablo 9 ve Tablo 10 incelendiğinde KDD10CORRECTED veri setinde üzerinde %99,99, KDDTEST veri setinde üzerinde %97,90 ve KDD10CORRECTED+KDDTEST veri setinde üzerinde %100 ile Torbalama sınıflandırıcısı sahip olmuştur. Bunun nedeni olarak TÖ sınıflandırıcıların birden fazla iyi başarım gösteren algoritmaları birleştirilerek başarım oranını mümkün olduğunca yükseltilmesi olarak düşünülebilir. Sonuç olarak bu tarz çalışmalar için TÖ sınıflandırıcıları, KA ve DVM'ye göre daha iyi başarım elde ettiği görülmüştür. Bu veri setleri için TÖ sınıflandırıcılarından olan Torbalama sınıflandırıcısının tercih edilmesi daha yerinde olacaktır.

Yazarların Katkısı

Tüm yazarlar eşit oranda katkı sağlamıştır.

Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

Kaynaklar

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

[2] Burukanlı M., Budak Ü., Çıbuk M. 2019. Saldırı Tespit Sistemlerinde Makine Öğrenme Metotlarının Kullanımı. In: Uluslararası Bilim ve Mühendislik Sempozyumu, 20-22 Haziran 2019, Siirt, Türkiye, 1052-1057.
[3] Sonawane H.A., Pattewar T.M. 2015. A Comparative Performance Evaluation of Intrusion Detection Based on Neural Network and PCA. In: 2015 Int. Conf. Commun. Signal Process. ICCSP 2015, 841-845.
[4] Aburomman A.A. Reaz M.B.I. 2017. A Survey of Intrusion Detection Systems Based on Ensemble and Hybrid Classifiers. Comput Secur, 65: 135-152.
[5] Golovko V.A., Vaitsekhovich L.U., Kochurko P.A., Rubanau U.S. 2007. Dimensionality Reduction and Attack Recognition Using Neural Network Approaches. In: 2007 Int. Jt. Conf. Neural Networks. IEEE, 12-17 Aug. 2007, Orlando, FL, USA, 2734-2739.
[6] Wang W., Battiti R. 2006. Identifying Intrusions in Computer Networks with Principal Component Analysis. In: First Int. Conf. Availability, Reliab. Secur. IEEE, 20-22 April 2006, Vienna, Austria, Austria, 270-279.
[7] Pattewar T.M., Sonawane H.A. 2015. Neural Network Based Intrusion Detection Using Bayesian with PCA and KPCA Feature Extraction. In: 2015 IEEE Int. Conf. Comput. Graph. Vis. Inf. Secur. IEEE, 2-3 Nov. 2015, Bhubaneswar, India, 83-88.
[8] Mukkamala S., Janoski G., Sung A. 2002. Intrusion Detection Using Neural Networks and Support Vector Machines. In: Proc. 2002 Int. Jt. Conf. Neural Networks. IJCNN'02 (Cat. No.02CH37290). IEEE, 12-17 May 2002, Honolulu, HI, USA, USA, 1702-1707.
[9] Özgür A., Erdem H. 2012. Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması. Bilişim Teknol Derg., 5 (2): 41-48.
[10] 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: 26.01.2021).
[11] Burukanlı M. 2020. Copula Fonksiyonlarını Kullanarak Bilgisayar Ağlarında Saldırı Tespiti. Yüksek Lisans Tezi, Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü, Bitlis.
[12] Liu W.M., Chang C.I. 2007. Variants of Principal Components Analysis. In: Int. Geosci. Remote Sens. Symp., 1083-1086.
[13] Abdi H., Williams L.J. 2010. Principal Component Analysis. Wiley Interdiscip Rev Comput Stat, 2 (4): 433-459.
[14] Jinhu L., Xuemei L., Lixing D., Liangzhong J. 2010. Applying Principal Component Analysis and Weighted Support Vector Machine in Building Cooling Load Forecasting. In: 2010 Int. Conf. Comput. Commun. Technol. Agric. Eng. 12-13 June 2010, Chengdu, China, 434-437.
[15] Çalış A., Kayapınar S., Çetinyokuş T. 2014. Veri Madenciliğinde Karar Ağaci Algoritmalari ile Bilgisayar ve İnternet Güvenliği Üzerine Bir Uygulama. Endüstri Mühendisliği Derg., 25 (3-4): 2-19.
[16] Safavian S.R., Landgrebe D. 1991. A Survey of Decision Tree Classifier Methodology. IEEE Trans Syst Man Cybern, 21 (3): 660-674.
[17] Yu Y., Zhong Liang F., Xiang Hui Z., Wen Fang C. 2009. Combining Classifier Based on Decision Tree. In: 2009 WASE Int. Conf. Inf. Eng. IEEE, 10-11 July 2009, Taiyuan, Chanxi, China, 37-40.
[18] Masud M., Khan L., Thuraisingham B. 2011. Data Mining Tools for Malware Detection. CRC Press. Taylor & Francis, London.
[19] Lu S-X., Meng J., Cao G-E. 2010. Support Vector Machine Based on A New Reduced Samples Method. In: 2010 Int. Conf. Mach. Learn. Cybern. IEEE, 11-14 July 2010, Qingdao, China, 1510-1514.
[20] Kim H.C., Pang S., Je H.M., Kim D., Bang S.Y. 2003. Constructing Support Vector Machine Ensemble. Pattern Recognit, 36 (12): 2757-2767.
[21] Cıbuk M., Budak U., Guo Y., Cevdet Ince M., Sengur A., 2019. Efficient Deep Features Selections and Classification for Flower Species Recognition. Meas J Int Meas Confed, 137: 7-13.
[22] Budak Ü., Cömert Z., Çıbuk M., Şengür A. 2020. DCCMED-Net: Densely Connected and Concatenated Multi Encoder-Decoder CNNs for Retinal Vessel Extraction from Fundus Images. Med Hypotheses, 134: 1-9.