İÇİNDEKİLER

GİRİŞ

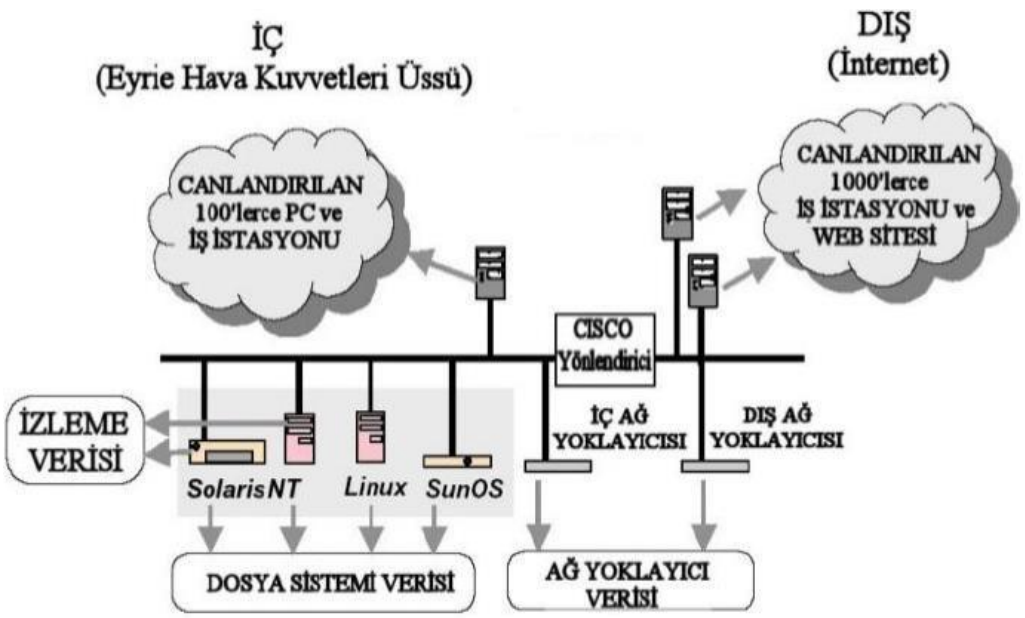
Yaşadığımız çağda şimdiye kadar hiç olmadığı kadar bilgi üretilmekte işlenmekte ve bunlara erişilmektedir. Bilginin bu denli hızlı üretilip yayılmasında hiç kuşkusuz bilgisayar teknolojileri en büyük teknik faydayı sağlamaktadır. Bilgiye erişim ve paylaşım için ise en fazla verimliliği bilgisayar ağları sağladığından, ağ işleyişinin düzgün olması hayati önem taşımaktadır. Ağ trafiğindeki anormallikler ise ağın gerektiği gibi kullanımını engelleyen unsurların başında gelmektedir. Bu anormallikler altyapı sorunlarından kaynaklanabileceği gibi ağın kötüye kullanılması veya ağa yapılan saldırılardan da kaynaklanabilmektedir [1].
Birçok kaynakta saldırı tespit sistemleri olarak da anılan anomali tespit sistemleri ağda oluşan düzensizlikleri tespit edip ilgili kişileri veya yazılımları uyarmayı sağlayan sistemlerdir.

Bu çalışmada büyük veri madenciliği tekniklerinden biri olan makine öğrenmesinin sınıflandırma algoritmalarından k-NN algoritması algoritması ile KDDCup99 veri seti üzerinde analizler yapılarak olası anomalilerin tespitinin analizi gerçekleştirilmiştir.

1. KULLANILAN ARAÇ VE YÖNTEMLER

1.1. KDDCup99 Veri Setinin Detaylı Anlatımı

Saldırı Tespit Sistemleriyle ilgili çalışmalarda en sık kullanılan veri seti DARPA 1998 ve 1999 veri setleridir. Biz de model oluşturma çalışmamızda yine bu verilerden türetilen KDD Cup'99 veri setini kullanacağız. Veri setini oluşturan kaynak aşağıdaki Şekil6'da da görüldüğü gibi saldırının hedefi olan bir iç ağ ve saldırıyı gerçekleştiren bir dış ağ olmak üzere iki farklı ağdan oluşmaktadır:



Şekil 1.1 Darpa Ağ Yapısı [3]

KDD'99 veri kümesi DARPA veri kümesinin bazı önişlemlerden geçirilmesiyle elde edilen 41 özelliğe sahip bir veri kümesidir. Bu veri kümesinin amacı diğer veri kümelerinde olduğu gibi farklı anomali tespit sistemlerinin eğitim ve test işlemlerini kolaylıkla gerçekleştirmek ve bu sistemlerin değerlendirilmesini sağlamaktır. DARPA ile her ne kadar anomali tespit sistemleri için veri kümesi probleminin çözülmesi adına önemli adımlar atılmışsa da KDD'99 ile eğitim ve test sonuçları çok daha hızlı alınabilmesi ve buna bağlı olarak anomali tespit sistemlerinin değerlendirilmesinin daha kısa sürede yapılabilmesi adına bu alanda atılmış önemli bir adımdır [2].

KDD'99 veri kümesi, 9 temel ve 32 adet türetilmiş olmak üzere toplam 41 özellikten oluşturmaktadır. Bu 41 özellik aşağıdaki gibi 3 ana gruba ayrılmıştır:

-İçerik özellikleri (content features)

-Sunucu tabanlı trafik özellikleri (host-based traffic features)

-Zamana bağlı trafik özellikleri (time-based traffic features)

Çizelge 1.1, Çizelge 1.2 ve Çizelge 1.3'de, sırasıyla bu 3 grup ve gruplara ait veri özellikleri gösterilmiştir.

Çizelge 1.1 İçerik özellikleri

Özellik Adı	Tanım	Tip
Duration	Bağlantı Uzunluğu	Sürekli
Protocol_type	Protokol Tipi	Ayrık
Service	Servis tipi	Ayrık
Src_bytes	Kaynaktan hedefe veri	Sürekli
Dst_bytes	Veri bayt sayısı	Sürekli
Flag	Bayrak	Ayrık
Land	Kaynak ve Hedef ip aynı ise 1 değil ise 0	Ayrık
Wrong_fragment	Yanlış parçalama	Sürekli
Urgent	Acil Paket sayısı	Sürekli

İçerik özellikleri, sadece TCP bağlantılarıyla ilgili olan özelliklerdir. Diğer gruplarda olduğu gibi ağdaki veriler üzerinde herhangi bir önişlem yapılmasına gerek yoktur [2]. Sunucu tabanlı trafik özellikleri, etki alanı (domain) bilgisi ile ilgili içerik özellikleridir.

Çizelge 1.2 Sunucu tabanlı trafik özellikleri

Özellik Adı	Tanım	Tip
Hot	"hot" göstergesi	Sürekli
Num_failed_logins	Hatalı giriş sayısı	Sürekli
Logged_in	Giriş başarılı ise 1 değil ise 0	Ayrık
Num_compromised	Gizliliğin ihlal edilme sayısı	Sürekli
Root_shell	"Root Shell" elde edildiyse 1 değilse 0	Ayrık
Su_attempted	"Su root" komutu girildiyse 1 değilse 0	Ayrık
Num_root	"root" erişim sayısı	Sürekli
Num_file_creations	Dosya oluşturma işlemleri sayısı	Sürekli
Num_shells	Shell promptlarının sayısı	Sürekli
Num_access_files	Kontrol dosyalarına erişim işlemleri sayısı	Sürekli
Num_outbound_cmds	ftp oturumunda giden komut sayısı	Sürekli
İs_hot_login	Giriş "hot" listesindeyse 1 değilse 0	Ayrık
İs_guest_login	Giriş "guest" ise 1 değilse 0	Ayrık

Zamana bağlı trafik özellikleri, “aynı sunucu” ve “aynı servis” özelliklerine göre kullanılan özelliklerdir. “Aynı sunucu” özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantılardır, “aynı servis” özellikleri ise son iki saniye içerisinde aynı servise yapılan bağlantılardır [2].

Çizelge 1.3 Zamana bağlı trafik özellikleri

Özellik Adı	Tanım	Tip
Count	Aynı sunucuya önceki iki bağlantıyla aynı bağlantıların sayısı	Sürekli
Serror_rate	“SYN” hata bağlantılarının yüzdesi	Sürekli
Rerror_rate	“Rej” hata bağlantılarının yüzde	Sürekli
Same_srv_rate	Aynı servise bağlantıların yüzdesi	Sürekli
Diff_srv_rate	Farklı servislere bağlantıların yüzdesi	Sürekli
Srv_count	Aynı servise önceki iki bağlantıyla aynı bağlantıların sayısı	Sürekli
Srv_serror_rate	“SYN” hata bağlantılarının yüzdesi	Sürekli
Srv_rerror_rate	“Rej” hata bağlantılarının yüzdesi	Sürekli
Srv_diff_host_rate	Farklı servislere bağlantıların yüzdesi	Sürekli

1.2. Saldırı Tipleri

Bilgisayar sistemlerine yapılan saldırılar, birçok araştırmacı tarafından farklı gruplandırmalar yapılmıştır [4,5]. Saldırganların sürekli kendilerini yenilemeleri ve bilgisayar sistemlerindeki açıkları tespit etmeleri nedeniyle saldırı tiplerindeki çeşitlilik çok fazla artmıştırç MIT Lincoln Laboratuvarlarında yapılan bu çalışmalar sonucunda bilgisayar sistemlerine yapılan saldırılar kullandıkları yöntemlere göre dört ana gruba ayrılmış ve **DoS, U2R, R2L** ve **Probing** olarak adlandırılmıştır [6]. Bu saldırıların detaylı tanımları aşağıdaki başlıklarda sunulmuştur.

1.2.1. Hizmet aksattırma saldırıları

Hizmet aksattırma saldırıları olarak da bilinen DoS saldırıları sistemin hizmetlerini engellemek amacıyla yapılırlar. Bunu yapmak için sisteme cevap verebileceğinden çok daha fazla istek gönderilerek verilen hizmet aksattırılır. DoS saldırılarına örnek olarak SYN flood, Smurf, UDPstorm, Pingflood, Neptune, Mailbomb saldırıları verilebilir [7].

1.2.2. U2R saldırıları

U2R saldırıları sayesinde kullanıcılar normal yetkilere sahip olan kendi hesaplarından oturum açtıktan sonra yönetici yetkisine ulaşarak sistem üzerinde istedikleri bilgilere erişebilirler. En çok bilinen U2R saldırı tipleri Eject, Ffbconfig, Fdformat, Loadmodule, Perl gibi saldırılardır [7].

1.2.3. R2L saldırıları

Bu saldırı tipinde saldırgan saldırdığı makineye ağ üzerinden paketler yollayarak açıkları tespit etmeye çalışır. Bu konuda birçok araç olması ve bu araçlara kolay erişilebilir olması nedeniyle, sistemde var olan açıklar kapatılmamışsa etkili ve kolay bir saldırı yöntemidir. En çok bilinen R2L saldırı tiplerine Dictionary, Guest, Imap, Named, Sendmail gibi saldırılar örnek olarak verilebilir [7].

1.2.4. Probing saldırıları

Probing saldırıları, ağı veya bilgisayarı tarayarak zayıflıkları tespit etmek ve sistem yapısıyla ilgili genel bir bilgiye ulaĢmak için yapılmaktadır. Bu tip saldırılarda önce sistem hakkında detaylı bilgi edinildikten sonra saldırının nasıl olacağı belirlenir. Bu saldırılar için kullanılan araçlar aynı zamanda güvenlik uzmanları tarafından sistemi güvenliğinin test edilmesi için de kullanılır. . En çok bilinen Probe saldırı tipleri, Ipsweep, Mscan, Nmap, Saint, Satan gibi saldırılar örnek olarak verilebilir [7].

KDD'99 veri kümesi 38 farklı atak içermektedir. KDD'99 eğitim veri kümesinde bulunan 24 atak ve bu atakların ait oldukları saldırı tipleri ve saldırıları veri kümesinde bulunduran örnek sayıları Çizelge 1.4'de verilmiştir.

Çizelge 1.4 KDD'99 veri kümesinin %10'luk kısmındaki saldırı örneklerinin sayıları

Saldırı	Örnek sayısı	Kategori
Smurf.	280790	DoS
Neptune.	107201	DoS
Back.	2203	DoS
Teardrop.	979	DoS
Pod.	264	DoS
Land.	21	Dos
Normal.	97277	Normal
Satan.	1589	Probe
Ipsweep.	1247	Probe
Portsweep.	1040	Probe
Nmap.	231	Probe
Warezclient.	1020	R2L

Guess_passwd	53	R2L
Warezmaster	20	R2L
İmap.	12	R2L
ftp_write	8	R2L
Multihop.	7	R2L
Phf.	4	R2L
Spy.	2	R2L
Buffer_overflow	30	U2R
Rootkit.	10	U2R
Loadmodule	9	U2R
Perl	3	U2R

Sadece test veri kümesinde yer alan, eğitim kümesinde yer almayan ve etiketlenmiş KDD'99 dosyasından alınan 14 farklı atağa ait saldırı tipi ve örnek sayıları Çizelge 1.5'de sunulmuştur.

Çizelge 1.5 Eğitim kümesinde yer almayan test veri kümesinde bulunan saldırılar

Saldırı	**Örnek Sayısı**	**Kategori**
Apache	794	DoS
Mailbomb	5000	DoS
Processtable	759	DoS
Udpstorm	2	DoS
Mscan	1053	Probe
Saint	736	Probe
Httptunnel	138	R2L
Named	17	R2L
Sendmail	17	R2L
Snmpgetattack	1040	R2L
Xlock	9	R2L
Xsnoop	4	R2L
Ps	16	U2R

Şekil 1.2. Verisetindeki özellikler ve tipleri

1.3.Veri Setinin Hazırlanması

Çalışmamda "kddcup.data_10_ percent_corrected" dosya ismi ile internetten indirilebilen yaklaşık 75Mb büyüklüğünde ve içinde 494021 kayıt bulunan gerçek kddCupp-99 veri seti'nin 10%'una karşılık gelen veri seti kullanıldı. Veri setinde toplam 41 adet özellik bulunmaktadır.

Verisetinin daha anlamlı hale gelmesi için özellik seçmesi yapılmasına karar verilmiştir ve özellik seçimi yaparken literatür taramasında bulduğum "Lojistik Regresyon ile Bilgisayar Ağlarında Anomali Tespiti" adlı makale göz önüne alınmıştır. Makaleye göre aşağıdaki prensiplere uyarak bu sayıyı 9'a indirilmiştir:

-Parametrelerin birbirlerinden bağımsız olanları seçilmiştir. Örneğin root_shell, su_attempted, num_root alanlarının tümü birden alınmak yerine su_attempted alanı alınmıştır.

-Parametrelerin bağımlı değişkeni etkilemeyecek olanları seçilmemiştir.

-Veriler daha kolay işlenebilmek için binary hale getirilmiştir.

Bu prensiplere uyarak özellik olarak aşağıda sıralanan özellikleri kullanılmıştır.

- **protocol_type**
- **service**
- **flag**

- **land**
- **wrong_fragment**
- **hot**
- **um_failed_logins**
- **nsu_attempted**
- **num_access_files**

	Missing Records	Percentage (%)
duration	0	0.0
protocol_type	0	0.0
service	0	0.0
flag	0	0.0
src_bytes	0	0.0
dst_bytes	0	0.0
land	0	0.0
wrong_fragment	0	0.0
urgent	0	0.0
hot	0	0.0
num_failed_logins	0	0.0
logged_in	0	0.0
num_compromised	0	0.0
root_shell	0	0.0
su_attempted	0	0.0
num_root	0	0.0
num_file_creations	0	0.0
num_shells	0	0.0
num_access_files	0	0.0
num_outbound_cmds	0	0.0
is_host_login	0	0.0
is_guest_login	0	0.0
count	0	0.0
srv_count	0	0.0
serror_rate	0	0.0
srv_serror_rate	0	0.0
rerror_rate	0	0.0

Şekil 1.3. Eksik veri kontrol sonucu

Özellik seçiminden sonra verisetinden eksik veri olup olmadığı kontrol edilmiş ve Şekil 1.3' te görüldüğü gibi eksik veri bulunmamaktadır.

Verinin önişleme aşamasında verisetinin ölçeklenmesi de önemlidir. Veri ölçekleme, mevcut konumu ve yönü bozulamadan büyütüp küçültmek, veriyi algoritmanın daha iyi anlayabileceği şekle getirmek, veriyi daha iyi ifade etmeyi sağlamaktadır. Bu çalışmada ölçekleme olarak pratikte fazlasıyla kullanılan **standardizasyon** yöntemi tercih edilmiştir. Böylece, verisetindeki her feature için ortalaması 0, varyansı 1 yapar ve tüm özelliklerin aynı büyüklüte olmasını sağlar, veriyi sıkıştırır.

	protocol_type	service	flag	land	wrong_fragment	hot	num_failed_logins
0	tcp	http	SF	0	0	0	0
1	tcp	http	SF	0	0	0	0
2	tcp	http	SF	0	0	0	0
3	tcp	http	SF	0	0	0	0
4	tcp	http	SF	0	0	0	0

Şekil 1.4. Verisetindeki kategorik verilerin bir kısmı

```
In [40]: dataset['flag'].value_counts()

Out[40]: SF      378440
         S0       87007
         REJ      26875
         RSTR       903
         RSTO       579
         SH         107
         S1          57
         S2          24
         RSTOS0      11
         S3          10
         OTH          8
         Name: flag, dtype: int64
```

Şekil 1.5. Flag özelliğine ait farklı verileri

```
In [38]: dataset['service'].value_counts()

Out[38]: ecr_i          281400
         private        110893
         http            64293
         smtp             9723
         other            7237
         domain_u         5863
         ftp_data         4721
         eco_i            1642
         ftp               798
         finger            670
         urp_i             538
         telnet            513
         ntp_u             380
         auth              328
         pop_3             202
         time              157
         csnet_ns          126
         remote_job        120
         gopher            117
         imap4             117
         domain            116
         discard           116
         systat            115
         iso_tsap          115
         shell             112
         echo              112
         rje               111
         sql_net           110
         whois             110
         printer           109
                           ...
         klogin            106
         bgp               106
         vmnet             106
         uucp_path         106
         supdup            105
         ssh               105
         nnsp              105
         login             104
         hostnames         104
         efs               103
         daytime           103
         netbios_ns        102
         link              102
         pop_2             101
         ldap              101
         exec               99
         http_443           99
         netbios_dgm        99
         name               98
         kshell             98
         ctf                97
         netstat            95
         Z39_50             92
         IRC                43
         urh_i              14
         X11                11
         tim_i               7
         tftp_u              1
         red_i               1
         pm_dump             1
         Name: service, Length: 66, dtype: int64
```

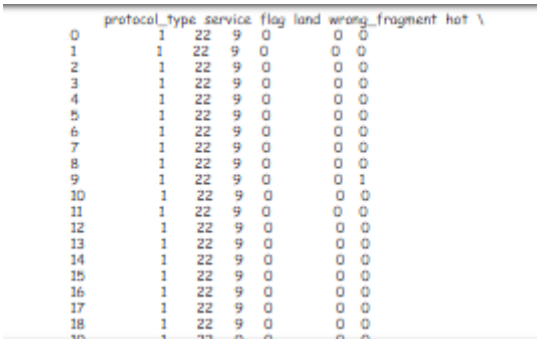
Şekil 1.6. service özelliğine ait farklı verileri

```
In [37]: dataset['protocol_type'].value_counts()

Out[37]: icmp    283602
         tcp     190065
         udp      20354
         Name: protocol_type, dtype: int64
```

Şekil 1.7. protocol_type özelliğine ait farklı verileri

Şekil 1.5. , Şekil 1.6. , Şekil 1.7.' te görüldüğü gibi flag, protocol_type, service özelliklerinin verileri sayısal değil yani kategorik veriler olarak adlandırılmaktadır. Bu durum, makine öğrenmesi algoritmaları matematiğe dayalı çalıştığından sayısal olması gerekmektedir. Bundan dolayı bu özelliklere ait veriler LabelEncoder yöntemin ile sayısal verilere çevrilmiştir.



	protocol_type	service	flag	land	wrong_fragment	hot
0	1	22	9	0	0	0
1	1	22	9	0	0	0
2	1	22	9	0	0	0
3	1	22	9	0	0	0
4	1	22	9	0	0	0
5	1	22	9	0	0	0
6	1	22	9	0	0	0
7	1	22	9	0	0	0
8	1	22	9	0	0	0
9	1	22	9	0	0	1
10	1	22	9	0	0	0
11	1	22	9	0	0	0
12	1	22	9	0	0	0
13	1	22	9	0	0	0
14	1	22	9	0	0	0
15	1	22	9	0	0	0
16	1	22	9	0	0	0
17	1	22	9	0	0	0
18	1	22	9	0	0	0

Şekil 1.8. Kategorik veriler sayısal verilere çevrildikten sonraki hali

Verisetimizi modele uygun hale getirmek için projenin ilk aşamasında eğitim verisi %70, test verisi %30 olarak parçalanmıştır ve eğitim verileri ile eğitim aşaması tamamlanıp model oluşturulmuştur. İkinci aşamasında, k-fold Cross Validation yöntemi ile tüm veri 3 ile 10 parçaya bölünerek teker teker her parça üzerinden eğitim ve test verisi olarak kullanılarak değerlendirilmiştir.

1.4. Kullanılan Ortam

1.4.1. Jupyter Notebook

Jupyter Notebook aldığımız notları ve hesaplamaları beraber tutmak için , Python kodları yazma olanağı sağlayan, görselleştirme özelliği bulunan yararlı bir araçtır.

1.5. Kullanılan Diller

Bu çalışmada **Python** programlama dili kullanılmıştır. Python, nesne yönelimli, yorumlamalı, birimsel ve etikileşimli yüksek seviyeli bir programlama dilidir.

1.6. Kullanılan Algoritma

1.6.1. K-NN Algoritması

Makine öğrenmesi algoritmaları arasında en basiti olarak belirtilmektedir. Modeli oluşturmak, yalnızca eğitim datasetinin saklanmasından oluşur. Yeni bir veri noktası için bir tahmin yapmak için algoritma, eğitim verisetindeki ''en yakın komşuları'' içindeki en yakın veri noktalarını bulur. Çoğunluk olan taraf seçilir. Öngörüde bulunmak istediğimiz noktaya en yakın eğitim verilerini dikkate alır. Bütün eğiitim seti içindekiler ile arasındaki uzaklık bulunur. En yakın k

örnek seçilir. Hem classifivation hem regression için kullanıılr. Bu çalışmada, problem sınıflandırma problemi olduğu için K-NN algoritması sınıflandırma amaçlı kullanılmıştır.

2. SONUÇ VE DEĞERLENDİRME

Bu çalışmada, KDD Cup'99 veri kümeleri kullanılarak büyük veri madenciliğinin sınıflandırma tekniklerinden olan k-NN tekniği algoritması ile bilgisayar ağlarında alışılmışın dışında bir durum olup olmadığı hakkında analizler yapılmış ve bu analizlerin nasıl gerçekleştirildiği üzerinde durulmuştur.

Modelin başarısını ölçerken confusion matrix diye adlandırılan karışıklık matrisinden yararlanılır. Confusion matrix, gerçek değerlerin bilindiği test verisi üzerinde bir sınıflandırma modelinin doğruluğunu, performansını ölçmek için kullanılmaktadır. Çalışmada doğruluk ölçümü doğruluk oranı (TP+ TN) / Toplam formulüne dayanarak bulunmuştur [8].

		GERÇEK DEĞERLER	
		POZİTİF	NEGATİF
TAHMİN EDİLEN DEĞERLER	POZİTİF	Gerçek Pozitif True Positive – TP	Sahte Pozitif False Positive – FP
	NEGATİF	Sahte Negatif False Negative – FN	Gerçek Negatif True Negative – TN

Şekil 2.1. Karışıklık matrisi

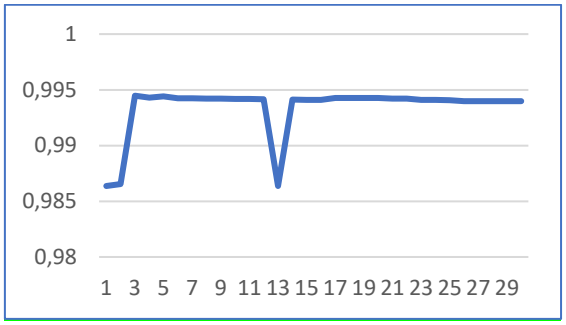
k-NN algoritmasında parametre olarak en yakın komşu sayısı "k" değeri belirleniyor. Bu çalışmada k değerine 1'den 30' a kadar değerler verilmiştir. Çalışma sonucundaki modelin doğruluk ya da başarı oranı şekil 2.2' de görülmektedir. Şekilden de anlaşılacağı üzere k parametresinin değeri arttıkça modelimizin başarısı ufak tefek artıp azalma göstermektedir. En yüksek başarı oranı k= 3 iken %0.994473 olarak sonuçlanmıştır ve son olarak k=26 değerinde %0.993993' te sabitlenmiştir.

```
k=1 0.9863703558306293
k=2 0.9865298386156894
k=3 0.9944733081023389
k=4 0.9942954234574641
k=5 0.994418102522895
k=6 0.9942463518312917
k=7 0.9942402178780202
k=8 0.9942218160182056
k=9 0.9942095481116625
k=10 0.9941911462518479
k=11 0.9941911462518479
k=12 0.9941604764854901
```

```
k=13 0.9863703558306293
k=14 0.994135940672404
k=15 0.9941114048593178
k=16 0.9940991369527747
k=17 0.9942586197378348
k=18 0.9942770215976495
k=19 0.9942708876443779
k=20 0.9942647536911063
k=21 0.994215682064934
k=22 0.9942095481116625
k=23 0.9940991369527747
k=24 0.9940991369527747
k=25 0.9940746011396885
k=26 0.9939948597471584
k=27 0.9939948597471584
k=28 0.9939948597471584
k=29 0.9939948597471584
k=30 0.9939948597471584
```

Şekil 2.2. Modelin başarı oranlarının çıktıları

Şekil 2.3 ’ te modelin başarısının grafiksel olarak gösterilmektedir.



Şekil 2.3. Modelin başarı oranlarının grafiksel hali

Model iyi bir başarı göstermektedir. Model, güvenlik seviyesi çok yüksek olması gereken ve yanlış alarmlarla uğraşacak yeterli elemanı olan, kritik öneme sahip ağ işletim merkezleri için uygundur.

Bu çalışmada başvurulan KDD Cup’99 veri kümesi yaygın olarak şimdiki saldırı tespit sistemlerinde kullanılmakta ama bu 1999’un verisi olduğu için ve ağ teknolojileri ve saldırı metotları sürekli değiştiğinden bu, şimdiki gerçek ağ durumunu yansıtamayabilir. Bu yüzden gelecekte yapılacak çalışmalarda, günümüzdeki ağ durumunu daha doğru bir şekilde yansıtabilmek için daha yeni bilgiye sahip olunup, test edilip ve kıyaslama yapılması önerilmektedir. Bu çalışmada karşılaşılan güçlük olarak toplanan verilerin işlenmesinin uzun sürmesi olarak söylenebilir.

REFERANSLAR

1. Budak, İ, Şen, B ve Yıldırım, M.Z., "Lojistik regresyon ile bilgisayar ağlarında anomali tespiti", XIII. Akademik Bilişim Konferansı, Antalya (2013).
2. Güven, E. N., "Zeki saldırı sistemlerinin incelenmesi, tasarımı ve gerçekleĢtirilmesi", Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, 1-91 (2007).
3. M. A. Aydın, "Bilgisayar Ağlarında Saldırı Tespiti için İstatistiksel Yöntem Kullanılması", İTÜ Yüksek Lisans Tezi, 2005.
4. Cabrera, B.D., Cabrera, L. Lewis and R.K. Mehra, "Detection and classification of intrusions and faults using sequence of system calls", ACM SIGMOD record, 30(4): 25-34 (2001).
5. Bace, R., Mell, P., "Intrusion detection systems", Technical Report, National Institute of Standards and Technology, NIST SP300-31, Scotts Valley, CA, 5-46 (2001).
6. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "Off-line intrusion detection evaluation data" http://www.ll.mit.edu/IST/ideval/ (2009).
7. Mukkamala, S., Janoski, G., Sung, A., "Intrusion detection using neural networks and support vector machines", IEEE International Joint Conference on Neural Networks, IEEE Computer Society Press, 1702-1707 (2002).
8. İnternet: https://veribilimcisi.com/2017/07/18/karisiklik-matrisi-nedir/(Erişim Tarihi: 24.12.2018)