Yapay Sinir Ağları İle Anomali Tespiti

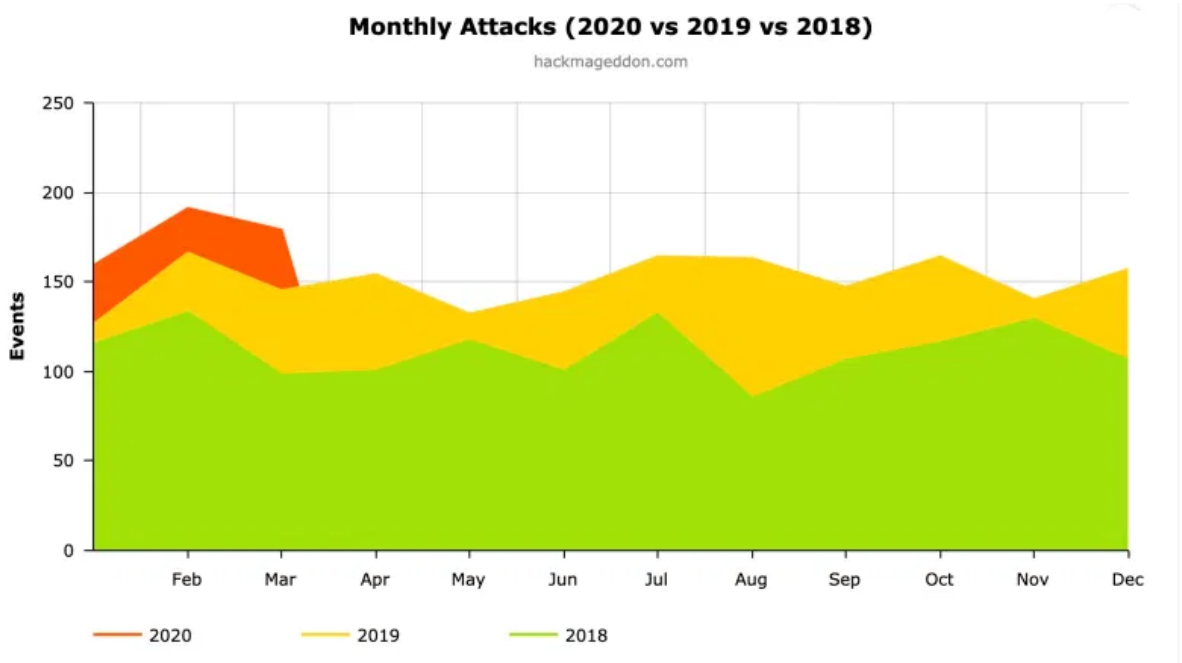
Melih Can Akbulut 151002016 İÖ

Mühendislik Fakültesi, Bilgisayar Müh. Bölümü, Düzce Üniversitesi, Düzce – Türkiye

Özet : Her geçen yıl siber saldırılar küresel ekonomi için bir numaralı tehdit olma yolunda ilerliyor. Devletler, büyük şirketler, küçük diyebileceğimiz şirketler dahil olmak üzere çoğunun siber güvenlik departmanı bulunmaktadır. Bulunmayan şirketler ise siber güvenlik firmalarından danışmanlık almaktadır. Açıklanan verilere göre 2017 yılında siber suç, küresel ekonomiye 600 milyar dolara 2018 yılında ise mali hasar yıllık %50 artışla 1 trilyon doları aştı [7]. Bunun yanı sıra siber zorbalık her geçen yıl artmakta ve önüne geçilmesi gereken bir sorun haline gelmektedir. Bu çalışmada ise Siber Güvenlik de Yapay Zeka Uygulamaları konu başlığı altında Ağ Trafik Analizi ile anomali tabanlı Saldırı Tespit / Engelleme Sistemleri (IDS/IPS) geliştirmiştir. Çalışmada kullanılan veri kümesi, 2015 yılında New South Wales Üniversitesi tarafından Avusturalya Siber Güvenlik Merkezi'nin laboratuvarlarında 49 özellik oluşturmak için on iki algoritma geliştirerek oluşturduğu veri seti olan UNSW-NB15 [6] veri kümesidir. Geliştirilen bu yapay sinir ağı test sonuçlarında %92 doğruluk ile tahmin yaptığı gözlenmiştir.

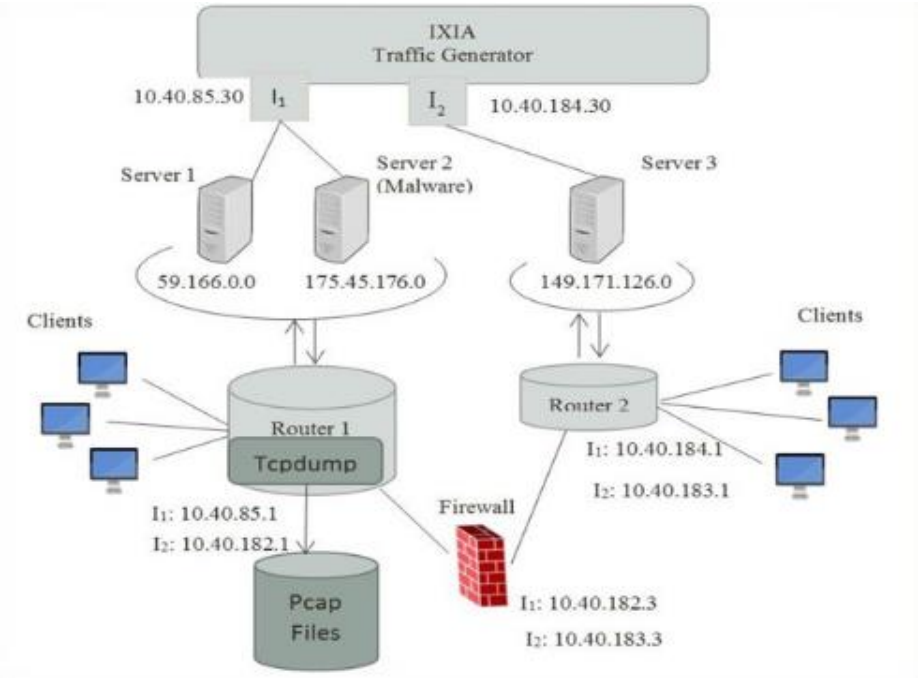
1. GİRİŞ

Siber güvenlik kavramı ve bilgisayarların güvenliğe ihtiyacı 1986 yılında kamu ve kişisel fark etmeksizin kişilerin bilgisayarlarına yoğun bir şekilde saldırı yapılması Amerikan hükümetini harekete geçirdi. IBM 1989 yılında ilk anti virüs programını yazdı. 1994 yılında web tarayıcıları üretildi. Bunların peşine güvenlik duvarları ve belirli işletim sistemlerinin bazı arka kapıları ve bir çok güvenlik açığı kapatıldı. Hızlı bir tarihçenin ardından aslında siber güvenlik her zaman kullandığımız bir alt yapı idi. Bunlara örnek olarak virüs programları, güvenlik duvarları, e-posta kutumuzda ki spam postalara kadar daha detay verecek olursak bir yerel ağ üzerinde port yönetimlerine kadar aslında siber güvenlik tamamen hayatımıza girmiş durumda. Fakat siber zorbaların mottosu olan 'Her sistemin bir arka kapısı vardır, hiçbir sistem kusursuz değildir' ile her geçen gün başka arka kapılar başka sızma yöntemleri geliştirilmektedir. Bununla beraber bu hızlı gelişime ayak uydurmaya çalışan siber güvenlik kavramıdır. Şekil 1'de [8] açıklanan son verilere ve yıllara göre siber saldırı oranları yer almaktadır.



Şekil 1 : Yıllara Göre Siber Saldırı

Günümüzde siber güvenlik virüs korumaları, güvenlik duvarları, ağ üzerindeki veya bilgisayarların gerekli zafiyet açıkları, e-postaların incelenmesi ve gerekli eğitimlerin sağlanması gibi birçok alt başlıkta incelenebilir ve tedbir alınmalıdır ki bunlar sadece üstün körü bahsedebileceğimiz güvenliklerdir. Bu çalışmada kullanılan veri setinde Avusturalya Siber Güvenlik Merkezinin bir ağ modeli üzerinde belirli saldırıların gerçekleştiği ağ trafiğini izleyerek oluşturulan UNSW-NB15 veri setidir. Aşağıda bu veri seti için hazırlanan ağ modeli görseli verilmiştir



Şekil 2 : UNSW-NB15 İçin Oluşturulan Ağ Modeli

Bu çalışmada, bir yapay sinir ağı modeli oluşturduktan sonra amaç hazırlanan bu veri setinde yer alan dokuz farklı saldırı türünden veriler ile ağın eğitim ve test işlemlerini gerçekleştirdikten sonra en yüksek doğruluk ile ağ üzerinde ki işlemin saldırı olup olmadığını tahmin etmektir.

2. LİTERATÜR ÖZETİ

KDDcup99[1] veri seti ile geçmiş yıllarda yapılan bir Yapay Sinir Ağları ile Saldırı Tespit Sistemi geliştirilmiş. KDDcup99 veri setinin %10'luk kısmı kullanılmış ve Matlab r2012b ile kodlanmış bir çalışma mevcut. Kısaca KDDcup99 veri setinden de bahsetmek gerekirse askeri ağ ortamında simüle edilen ve çok çeşitli izinsiz girişleri içeren ve Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması için hazırlanan ve kullanılan veri setidir. Yapılan bu çalışmada Yapay Sinir Ağları kullanılarak gerekli nöron ve iterasyon sayılarının değiştirilerek denenmesi sonucunda %99'un üzerinde bir başarı sağlanmıştır. [2]

İnternetin yaygınlaşması ve ağa bağlı cihazların artmasından kaynaklı gelişen sorunlarımızın en önemlisi siber güvenliktir. Bir bilgisayarın ağına saldırı olup olmadığını yüksek başarı oranı ile tespit edilebilir bunun yanı sıra makineye öğrettiğimiz saldırı türlerinin sisteme zarar vermeye çalışıp çalışmadığını da ayırt edebileceğinden makine öğrenme algoritmaları hala geliştirilmeye çalışmakta ve saldırı tespit sistemleri de makine öğrenmesi algoritmaları ile geliştirilmektedir. Bu çalışmada ise CSE-CIC-IDS2018 veri seti kümesi kullanılmıştır. Benim için burada yapılan en önemli çalışma veri seti üzerinde bütün 79 özniteliğin tespit edilmek istenilen saldırı türüne göre öznitelik azaltma çalışmayı yapılmış ve 5 farklı veri seti ile ağı eğiterek sadece 79 özniteliğin sahip olduğu veri setinden daha yüksek bir doğruluk oranı elde edilmiştir. Yapılan çalışmaya ek olarak kdd-99 veri seti üzerinde destek vektör makineleri ve yapay sinir ağlarının doğruluk oranı test edilmiş ve yapay sinir ağlarının saldırı tespit doğruluğu destek vekör makinelerinden daha iyi bir performans sağladığı görülmüştür. Bu çalışmanın proje konumla hemen hemen çok benzemekte fakat makaledeki çalışmada DDOS, Bot, BruteForce, DOS şeklinde 4 farklı tehdit sınıflandırıcısı oluşturulmuştur. Kullanacağım veri seti ve planladığım çalışma, amaç olarak bu saldırı türlerinden daha fazlası için kontrol sağlanması amaçlanmaktadır. [3]

Bu çalışmada, anomali tabanlı saldırı tespit sistemlerinin makine öğrenmesi metodolojileri ile daha önce öğrenilmemiş saldırılar üzerinde daha az yanlış alarm vermekte olduğu ve daha iyi performans ile çalıştığı gözlemlenmiştir. Anomali tespitini en üst düzeyde doğrulukla ve daha hassas çalışması için denetimli, yarı denetimli, denetimsiz makine öğrenmesi yaklaşımları kullanılmıştır. Makalede daha önce saldırı tespit sistemleri üzerine çalışılmış birden fazla araştırmacının hangi algoritma ve hangi yaklaşımlar ile çalıştığını anlatmıştır. Yine bu makalede de destek vektör ile yapay sinir ağı da karşılaştırılmıştır. Destek vektörü yine geride kalmıştır. Ayrıca, kendi önerdikleri binom sınıflandırma kullanılarak 40 öznitelik üç gizli katman ile daha az sayıda öznitelik kullanılarak performansı karşılaştırılmış ve daha öz nitelik ile daha kesin sonuçlar elde edilmiş bir çalışma incelenmiştir. Yine bu makalede de NSL-KDD, KDD99, ADFA, UNSW-NB15 veri setleri incelenmiştir. Bu makale diğer makalelere göre daha çok ilgimi çekme sebebi benim kullandığım veri seti kullanılmıştır. Ayrıca her bir katmanda 10 nöron olacak şekilde 10 gizli katman kullanılmış

ve 10 epoch ile eğitilmiştir. Önerdikleri metod %99.5 doğruluk ile çalışırken yapay sinir ağları %81.5 doğruluk ile çalıştığı gözlemlenmiştir. [4]

Bu çalışmada, çok katmanlı perceptron kullanılmış ve geri yayılım algoritması ile geliştirilmiştir. İmzalara dayalı 'daha önceden öğretilmiş bir saldırı şeklini' anomali tespit yöntemleri, düşük yanlış alarm oranları ile çalışırken bilinmeyen saldırılar için ise ağımıza sürekli güncel veriler ile destek sağlamalıyız. Makalede NewFlow protokolü ile alınan veriler analiz edilmiştir. Bunun yanı sıra yapay sinir ağında aktivasyon fonksiyonu olarak sigmoid fonksiyon kullanılmıştır. Eğitim sırasında ağa ileri doğru yayılan girişler sunularak YSA'nın ürettiği çıkış ile istenen çıkış karşılaştırıldıktan sonra çıkış farklı ise geri yayılım ile çalışılmıştır. [5]

3. MATERYAL VE YÖNTEM

3.1 Kullanılan Veritabanı

Uygulamada kullanılan veri tabanı New South Wales Üniversitesi üzerinde paylaşılan UNSW-NB15 veri kümesidir. Bu veri seti Avusturalya Siber Güvenlik Merkezinde (ACCS) laboratuvar ortamında 9 farklı saldırı türü ve normal ağ paketlerinin yakalanarak 12 farklı algoritma ile oluşturulmuştur. Ağ üzerinde gerçek ve sentetik saldırı melezleri oluşturulmuştur. Bu veri setinde depolanan 2 milyon 540 bin veri bulunmaktadır. Bu verileri işlemesi ve bir ağa eğitimi kolay olmayacağından ayrı ayrı hem eğitim için ayrılmış 175.341 kayıt test için ise ayrılmış 82.332 kaydın csv formatındaki dosyaları paylaşıma sunulmuştur. Veri setinde bir ağ üzerinde tehdit yaratabilecek 49 farklı özniteliğe yer verilmiştir. Veri setinde 1 sınıf bulunmaktadır '1' için saldırı var '0' için saldırı yok şeklinde hazırlanmıştır. Özniteliklerden bazıları sözel olarak verilmiştir bazıları saniye bazlı sayısal olarak hazırlanmıştır. Aşağıdaki şekilde UNSW-NB15 için öznitelikler ve açıklamaları yer almaktadır.

Attribute Number	Feature	Description
42	ct_srv_dst	No. of connections that contain the same service (#14) and destination address (#3) in 100 connections according to the last time (#26).
43	ct_dst_ltm	No. of connections of the same destination address (#3) in 100 connections according to the last time (#26).
44	ct_src_ ltm	No. of connections of the same source address (#1) in 100 connections according to the last time (#26).
45	ct_src_dport_ltm	No of connections of the same source address (#1) and the destination port (#4) in 100 connections according to the last time (#26).
46	ct_dst_sport_ltm	No of connections of the same destination address (#3) and the source port (#2) in 100 connections according to the last time (#26).
47	ct_dst_src_ltm	No of connections of the same source (#1) and the destination (#3) address in in 100 connections according to the last time (#26).
48	attack_cat	The name of each attack category.
49	Label	0 for normal and 1 for attack records

Attribute Number	Feature	Description
1	srcip	Source IP address
2	sport	Source port number
3	dstip	Destination IP address
4	dsport	Destination port number
5	proto	Transaction protocol
6	state	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Record total duration
8	sbytes	Source to destination transaction bytes
9	dbytes	Destination to source transaction bytes
10	sttl	Source to destination time to live value
11	dttl	Destination to source time to live value
12	sloss	Source packets retransmitted or dropped
13	dloss	Destination packets retransmitted or dropped
14	service	http, ftp, smtp, ssh, dns, ftp-data ,irc and () if not much used service
15	Sload	Source bits per second
16	Dload	Destination bits per second
17	Spkts	Source to destination packet count
18	Dpkts	Destination to source packet count
19	swin	Source TCP window advertisement value
20	dwin	Destination TCP window advertisement value
21	stcpb	Source TCP base sequence number
22	dtcpb	Destination TCP base sequence number
23	smeansz	Mean of the packet size transmitted by the source
24	dmeansz	Mean of the packet size transmitted by the destination
25	trans_depth	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	Actual uncompressed content size of the data transferred from the server's http service
27	Sjit	Source jitter (mSec)
28	Djit	Destination jitter (mSec)
29	Stime	record start time
30	Ltime	record last time
31	Sintpkt	Source interpacket arrival time (mSec)
32	Dintpkt	Destination interpacket arrival time (mSec)
33	tcprtt	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	If source (#1) and destination (#3) IP addresses equal and port numbers (#2)(#4) equal then, this variable takes value 1 else 0
37	ct_state_ttl	No. for each state (#6) according to specific range of values for source/destination time to live (#10) (#11).
38	ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of flows that has a command in ftp session.
41	ct_srv_src	No. of connections that contain the same service (#14) and source address (#1) in 100 connections according to the last time (#26).

3.2 Yapay Sinir Ağı İle Anomali Tespiti

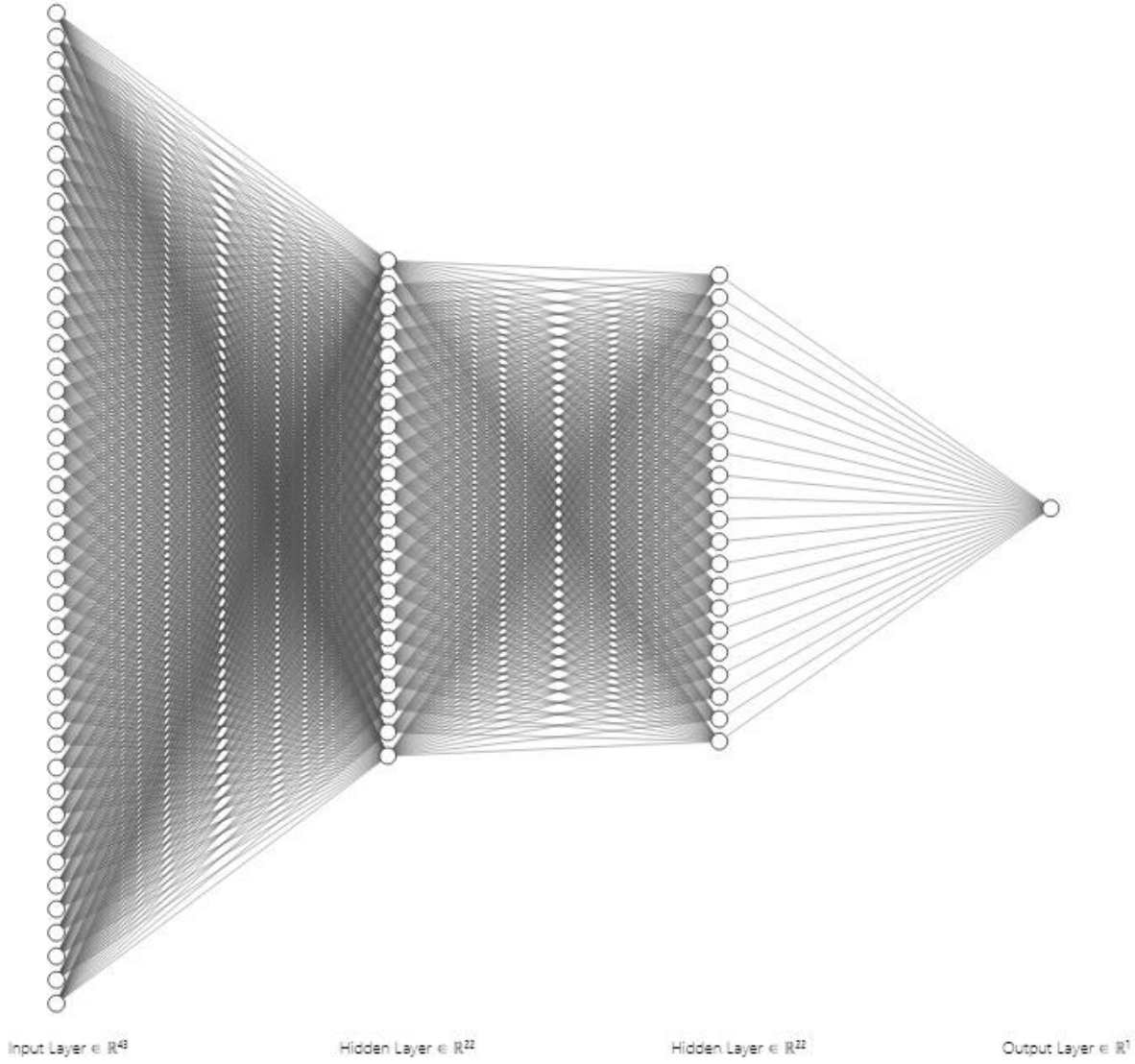
Bu çalışma Python 3.8.5 programlama dili ile Jupyter Notebook üzerinde Tensorflow, Keras, Scikit-Learn, Pandas kütüphaneleri kullanılarak kodlanmıştır. Yapay sinir ağları ve geri yayılım algoritması kullanılarak hazırlanan bu projede yapay sinir ağının test verileri sonucunda %92 doğruluk ile çalıştığı gözlemlenmiştir.

Veri tabanında belirtilen 49 öznitelik için hazırlanan eğitim ve test veri setinde hazırlanan kurum tarafından gerekli normalizasyon işlemleri ile öznitelik sayısı 44'e düşürülmüştür. Yapay sinir ağı modeli oluşturulmadan önce ID kısmı sadece kaç veri olduğu bilgisini verdiğinden ve hesaplamada hiçbir fark yaratmayacağından göz ardı edilmiş ve işlenecek veriden çıkartılmıştır. Ayrıca veri setinin Label kısmı da sınıf niteliği taşıdığı için giriş olarak yer alamayacağından veri setinden çıkartılmıştır. Bu işlemler sonucunda yapay sinir ağımız için gerekli giriş katmanı nöron sayısı 43 olarak belirlenmiş olup gerekli çıkış katmanı nöron sayısı da iki sınıf ile ayrıldığından 1 nöron olarak belirlenmiştir. Bu işlemler sonrasında gerekli denemeler ile iki gizli katman kullanılmıştır birinci gizli katmanda 22 nöron ikinci gizli katmanda ise 22 nöron kullanılmıştır böylece kullanılacak yapay sinir ağı modeli oluşturulmuştur.

Kodlama aşamasına geçildiğinde öncelikle verinin normalizasyon işlemleri yapılmıştır. Veri setinde bulunan 4 sözel veri öncelikli olarak sayısal verilere dönüştürülmüş sonrasında bütün veri, tahmin ve eğitimin daha ince hesaplanması düşüncesi ile 0-1 arasına normalize edilmiştir. Bunun sonucunda bütün veriler yapay sinir ağına sayısal olarak sunulabilecektir. Bu işlemlerden sonra gerekli kodlama işlemleri ile 43,22,22,1 şeklinde ki yapay sinir ağımız oluşturulmuştur. Oluşturulan yapay sinir ağında 1. gizli katmanda ve 2 gizli katmanda 'relu' aktivasyon fonksiyonu, çıkış katmanı için ise 'sigmoid' aktivasyon fonksiyonu kullanılmıştır. Durdurma kriterimiz olarak 10 epoch seçilmiştir. Sonrasında normalize edilen veri ağa sunulmuş ve test işlemi gerçekleştirilmiştir ve ağın %92 doğruluk ile tahmin işlemi yaptığı gözlemlenmiştir. Aşağıda kullanılan yapay sinir ağı topolojisinin şekli ve eğitim, test sırasında gözlemlenen doğruluk oranları verilmiştir.

```
Epoch 1/10
5480/5480 [==============================] - 10s 2ms/step - loss: 0.1244 - accuracy: 0.9452
Epoch 2/10
5480/5480 [==============================] - 9s 2ms/step - loss: 0.0865 - accuracy: 0.9664: 0s - los
Epoch 3/10
5480/5480 [==============================] - 10s 2ms/step - loss: 0.0704 - accuracy: 0.9753
Epoch 4/10
5480/5480 [==============================] - 9s 2ms/step - loss: 0.0387 - accuracy: 0.9869
Epoch 5/10
5480/5480 [==============================] - 9s 2ms/step - loss: 0.0183 - accuracy: 0.9940
Epoch 6/10
5480/5480 [==============================] - 11s 2ms/step - loss: 0.0073 - accuracy: 0.9981 0s - loss: 0.0074 - accuracy:  - ET
A: 0s - l
Epoch 7/10
5480/5480 [==============================] - 10s 2ms/step - loss: 0.0037 - accuracy: 0.9991
Epoch 8/10
5480/5480 [==============================] - 12s 2ms/step - loss: 0.0024 - accuracy: 0.9993
Epoch 9/10
5480/5480 [==============================] - 10s 2ms/step - loss: 0.0018 - accuracy: 0.9995
Epoch 10/10
5480/5480 [==============================] - 9s 2ms/step - loss: 0.0016 - accuracy: 0.9996
```

Şekil 3 : Ağın Eğitimi



Şekil 4 : Kullanılan Yapay Sinir Ağ Modeli

4. SONUÇLAR VE TARTIŞMA

Bu çalışmada, kullanılan veri tabanı öznitelikleri ile oluşturulan optimum yapay sinir ağı ile siber saldırı türlerinin çoğunun tanıyan bir tahmin mekanizması oluşturulmuştur. Yapılan eğitim ve test sonuçlarında ağ %92 doğruluk ile tahmin yapabilmektedir. Doğruluk oranını arttırmak için farklı aktivasyon fonksiyonları, farklı normalizasyon işlemlerinin yanı sıra farklı makine öğrenmesi algoritmaları ile de eğitim ve test aşamaları uygulanabilir. Bunun yanı sıra çok daha farklı ara katman nöron sayıları ile de deneme yanılma yöntemiyle daha yüksek doğruluk oranları elde edilebilir. Her ne kadar kesin bir şekilde artacak diyemeyecek olsak da denenebilir.

```
2573/2573 [==============================] - 3s 1ms/step - loss: 0.2366 - accuracy: 0.9295
[0.23662008345127106, 0.9295292496681213]
```

Şekil 5 : Eğitim Sonucu Gözlemlenen Doğruluk Oranı

5. REFERANSLAR

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[2] Mehmet Y., Abdullah Ç., Baha Ş., İdris B. "Yapay Sinir Ağları ile Ağ Üzerinde Saldırı Tespiti ve Paralel Optimizasyonu", (2014)

[3] Mehmet S. K., Metin T., Muhammed A. A., "Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespiti Modeli Uygulaması", (2020)

[4] Liu Z., Ghulam M., Li B., Luo J., Zhu Y., Lin Z., " Modeling Network Intrusion Detection System Using Feed-Forward Neural Network Using UNSW-NB15 Dataset ", (2019)

[5] Andropov S., Budko M., Budko Mi., Guirik A., "Network Anomaly Detection using Artificial Neural Networks ". (2017)

[6] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[7] https://www.technology.org/2019/07/17/biggest-cyber-attacks-and-their-cost-for-the-global-economy/

[8] https://www.hackmageddon.com/category/security/cyber-attacks-statistics/

Esra ŞATIR – Artificial Neural Network

Makale araştırma ve makale seçimi için, https://www.sciencedirect.com/

Makale araştırma ve makale seçimi için, https://ieeexplore.ieee.org/Xplore/home.jsp

6. EK-KODLAR

Normalizasyon kodları ;

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

df = pd.read_csv('unsw_egitim.csv')
dft = pd.read_csv('unsw_test.csv')


df.drop("service", axis='columns')
df.drop("proto", axis='columns')
```

```
df.drop("state", axis='columns')
df.drop("attack_cat", axis='columns')

dft.drop("service", axis='columns')
dft.drop("proto", axis='columns')
dft.drop("state", axis='columns')
dft.drop("attack_cat", axis='columns')

l2 = LabelEncoder()
label1 = l2.fit_transform(df['service'])
df["service"] = label1
label2 = l2.fit_transform(df['proto'])
df["proto"] = label2
label3 = l2.fit_transform(df['state'])
df["state"] = label3
label4 = l2.fit_transform(df['attack_cat'])
df["attack_cat"] = label4

label5 = l2.fit_transform(dft['service'])
dft["service"] = label5
label6 = l2.fit_transform(dft['proto'])
dft["proto"] = label6
label7 = l2.fit_transform(dft['state'])
dft["state"] = label7
label8 = l2.fit_transform(dft['attack_cat'])
dft["attack_cat"] = label8

Egitim = df
Test = dft

Y_Egitim = Test[['dur', 'proto', 'service', 'state', 'spkts', 'dpkts',
'sbytes',
       'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss',
'dloss',
       'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb',
'dwin',
       'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth',
       'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm',
       'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm',
       'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm',
       'ct_srv_dst', 'is_sm_ips_ports', 'attack_cat']]
X_Egitim = Egitim[['dur', 'proto', 'service', 'state', 'spkts', 'dpkts',
'sbytes',
       'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss',
'dloss',
       'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb',
'dwin',
       'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth',
       'response_body_len', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm',
       'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm',
       'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm',
       'ct_srv_dst', 'is_sm_ips_ports', 'attack_cat']]

X_Label = Egitim[['label']].astype('float64')
Y_Label = Test[['label']].astype('float64')

X_Egitim = X_Egitim.apply(lambda x : (x-x.min(axis=0) ) / (x.max(axis=0) -
x.min(axis=0)))
Y_Egitim = Y_Egitim.apply(lambda x : (x-x.min(axis=0) ) / (x.max(axis=0) -
x.min(axis=0)))
```

Eğitim ve Test Kodları ;

```
import keras
import tensorflow as tf
from Normalizasyon import *
from keras.models import Sequential
from keras.layers import Dense

classifier = Sequential()

classifier.add(Dense(22,kernel_initializer="normal", activation = 'relu',
input_dim=43))

classifier.add(Dense(22,kernel_initializer="normal", activation = 'relu'))

classifier.add(Dense(1,kernel_initializer="normal", activation =
'sigmoid'))

classifier.compile(optimizer= 'adam', loss = 'binary_crossentropy', metrics
= ['accuracy'])

classifier.fit(X_Egitim, X_Label ,epochs = 10)

classifier.evaluate(Y_Egitim,Y_Label)
```