SİBER GÜVENLİKTE VERİ MADENCİLİĞİ

ÖDEV-1 RAPOR

Büşra Kizilaslan (8)

Hasan Emir Kara (24)

Kyoto 2006+ dataset:

Kyoto 2006+ veri kümesi, başlangıçta 2006 Kasım'dan 2009 Ağustos'a kadar gerçek trafik verilerinin üç yılının kullanıldığı bir siber güvenlik veri kümesidir. Daha sonra 2006 Kasım'dan 2015 Aralık'a kadar ek verilerle güncellenmiştir. Bu veri kümesi genellikle İzinsiz Giriş Tespit Sistemleri (IDS) ve ağ güvenliği alanındaki araştırmalar ve analizler için kullanılır.

Özellikler: Veri kümesi toplamda 24 özellik içerir. Bu özelliklerden 14'ü, ağ giriş tespiti alanında bilinen bir veri kümesi olan KDD Cup '99 veri kümesinden türetilmiş istatistiksel özelliklerdir.[Table 1]

| Index | Feature name | Description |
|---|---|---|
| 1 | Duration | Bağlantı süresi saniyeler cinsinden |
| 2 | Service | Bağlantı hizmetinin türü örneğin HTTP,telnet |
| 3 | Source bytes | Source IP adresinin gönderdiği veri baytının sayısı |
| 4 | Destination bytes | Hedef IP adresinin gönderdiği veri baytının sayısı |
| 5 | Count | Geçmiş iki saniye içinde mevcut bağlantının kaynak ve hedef IP adreslerine sahip olan benzer bağlantıların sayısı |
| 6 | Same_srv_rate | Count özelliğinde aynı hizmete yapılan bağlantıların yüzdesi |
| 7 | Serror_rate | Count özelliği “ SYN” hatası olan bağlantıların yüzdesi |
| 8 | Srv_serror_rate | Srvcount (geçmiş iki saniye içinde hizmet türü aynı olan bağlantıların sayısı) özelliğinde SYN hatası olan bağlantıların yüzdesi |
| 9 | Dst_host_count | Mevcut bağlantının hedef IP adresi ile aynı olan son 100 bağlantı arasında, kaynak IP adresi de şu anki bağlantı ile aynı olan |

| | | |
|---|---|---|
| | | bağlantı sayısı. |
| 10 | Dst_host_srv_count | Şu anki bağlantının hedef IP adresi ile aynı olan son 100 bağlantı içinde, hizmet türü de şu anki bağlantı ile aynı olan bağlantıların sayısı. |
| 11 | Dst_host_same_src_port_rate | Dsthostcount özelliğinde kaynak portun mevcut bağlantının kaynak portu ile aynı olan bağlantıların yüzdesi |
| 12 | Dst_host_serror_rate | Dst_host_count özelliğinde 'SYN' hataları bulunan bağlantıların yüzdesi. |
| 13 | Dst_host_srv_serror_rate | Dst_host_srv_count özelliğinde 'SYN' hataları bulunan bağlantıların yüzdesi. |
| 14 | Flag | Her belirli bağlantı sonlandığında bir özel yazılır.Farklı bağlantılar için farklı durumlar olabilir ve bu durumlar bağlantı sonlandığında gözlemlenir ve bu özelliği kaydedilir. |

Ayrıca, IDS ağ performansını analiz ve değerlendirmek için özel olarak tasarlanmış 10 ek özellik bulunmaktadır.[Table 2]

| Index | Feature name | Description |
|---|---|---|
| 1 | IDS_detection | IDS'nin bağlantı için bir uyarı tetikleyip tetiklemediğine dair bir sayısal değeri saklar. '0' herhangi bir alarmın tetiklenmediğini ve '0' hariç farklı türdeki uyarıları temsil ederler.Parantez,aynı uyarının sayısını gösterir. |
| 2 | Malware_detection | Bağlantıda kötü amaçlı yazılımın(malware) tespit edilip edilmediğini temsil etmke için kullanılır; '0' herhangi bir saldırı bulunmadığını gösterir ancak bu özellikte sıfır olmayan bir sayısal değer bulunduğuunda,tespit edilen belirli bir saldırıyı temsil eder.Malware tespiti için 'clamAV' adlı bir yazılım kullanılı.Parantez,bağlantının varlığının tespiti sırasında aynı kötü amaçlı yazılımın toplam gözlemini temsil etmek için kullanılır |
| 3 | Ashula_detection | Bir bağlantıda kabuk(shell) kodları ve saldırı kodlarının kullanılıp kullanılmadığını belirtir. '0' herhangi bir kabukkodu veya saldırı kodunun gözlemlendiğini ifade eder.Aksine bir durum olduğunda sıfır olmayan bir sayı içerir ve her sayı farklı türdeki kabuk kodları için kullanılı.Parantez ise aynı kabuk kodu veya saldırı kodunun sayısını gösterir. |

| | | |
|---|---|---|
| 4 | Label | Bu veri kümesinin sınıf etiketi özelliğidir.Oturumun saldırıya uğrayıp uğramadığını belirtir.1 normal,-1bilinen bir saldırının gözlemlendiğini,-2 bilinmeyen bir saldırının gözlemlendiğini ifade eder. |
| 5 | Source_IP_Address | Oturumda kullanılan kaynak IP adresini ifade eder.IPv4 üzerindeki orijinal IP adresi benzersiz yerel IPV6 unicast adreslerinden birine sansürlenmiştir.Aynı özel IP adresleri yalnızca aynı ağ için geçerlidir.Aynı ağiçinde iki özrl IP adresi aynı ise bu IPv4 üzerindeki IP adreslerinin de aynı olduğu anlamına gelir. |
| 6 | Destination_Port_Number | Oturum tarafından kullanılan kaynak port numarasını içerir |
| 7 | Destinaiton_IP_Address | Oturumun IP aderslerini içerir.IP adesi,bazı güvenlik nedenlerinden dolayı karşılık gelen IPv4 özgü yerel bir IPv6 adresidir ve gizlenmiş veya sansürlenmiş bir biçimde olabilir. |
| 8 | Destination_Port_Number | Oturum tarafından kullanılan hedef port numarasını içerir. |
| 9 | Start_Time | Oturumun ne zaman başladığını belirtir. |
| 10 | Protocol type ("Duration") | Protokol Tipi (TCP, UDP...) ("Bağlantının toplam süresini temsil eder".) |

Makalelerden Topladığımız veriler üzerine 10 numaralı özellik Duration olarak geçmektedir fakat makalelerin yanlış bilgi verdiğini varsayarak bu özelliği

KDD Cup ’99 dataset 'in içindeki özelliklerde protokol type olarak geçtiğinden dolayı 10 numaralı özelliğimizi bu şekilde belirttik

Kyoto 2006+ veri kümesi, Kyoto Üniversitesi'nin çeşitli türdeki honeypotlardan elde ettiği gerçek trafik verilerine dayanan bir veri kümesidir. Bu veri kümesinin kaynağı, Kyoto Üniversitesi'nin ağ güvenliği araştırmaları için kendi honeypot sistemi olan Kyoto Honeypot System (KHS)'dir. KHS, 2006 yılından beri dünyanın dört bir yanından gelen ağ trafiğini toplamakta ve analiz etmektedir. Bu veri seti, bilgiye yetkisiz kullanım girişimlerini tespit eden honeypotlar, darknet sensörleri, e-posta sunucusu, web tarayıcı ve diğer bilgisayar ağ güvenliği mekanizmaları kullanılarak yakalanmıştır. KHS, ağ saldırılarını tespit etmek ve sınıflandırmak için çeşitli yöntemler kullanmaktadır.

Kyoto 2006+ dataset saldırı türleri

Kyoto 2006+ veri kümesinde, ağ trafiğindeki normal ve anormal davranışları ayırt etmek için kullanılabilecek çeşitli türde saldırılar bulunmaktadır

- DoS (Denial of Service
- Probe (Sonda)
- R2L (Remote to Local)
- U2R (User to Root)

Kyoto 2006+ veri kümesinin siber güvenliğe olan ilgisi, bu veri kümesinin ağ saldırısı tespit sistemleri (NIDS) geliştirmek ve değerlendirmek için kullanılabilmesidir. NIDS, ağ trafiğini izleyerek potansiyel tehditleri belirleyen ve önleyen bir yazılım veya donanım aracıdır. NIDS, ağ güvenliği için hayati bir öneme sahiptir, çünkü ağ saldırıları gün geçtikçe daha karmaşık ve zararlı hale gelmektedir. Kyoto 2006+ veri kümesi, NIDS'in performansını ölçmek ve iyileştirmek için kullanılabilir. Ayrıca, Kyoto 2006+ veri kümesi, ağ saldırılarının kaynaklarını, hedeflerini ve etkilerini analiz etmek için de kullanılabilir.

Yapılan İşlemler

-TXT Belgesi Olarak İndirilen Dataset Excel İle CSV formatına Dönüştürüldü

-Gerekli Kütüphaneler Eklendi

-Dataset Tanımlandı

-Datasete Ait Olan Feature'lar İçin Araştırma Yapılıp Eklendi

-Dataset'e Feature'ler Tanımlandı

-Dataset Gösterildi

-Dataset için önemli olan Feature'lar Sayısal İstatistik Gösterildi

-PCA ile fazlalık olan sütunlar 2 sütuna indirgendi

-PCA Öncesi ve Sonrası İstatistikler Görüntülendi

-Label İstatistiği Görsel Grafik Üzerinde Gösterildi

Gerekli Kütüphaneler Eklendi

Kodlar Ve Yapılan İşlemler

```
import pandas as pd
import numpy as np
import sys
import sklearn
import io
import random
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
# Dataset Tanımlanması
dataset = 'https://raw.githubusercontent.com/WarFires/Kyoto2006-Dataset/main/kyoto2006%2B20151011.csv'
# Değerler
features = ["duration","service","src_bytes","dst_bytes",
    "count","serror_rate","srv_serror_rate","same_srv_rate",
    "dst_host_count","dst_host_srv_count","dst_host_same_src_port_rate",
    "dst_host_serror_rate","dst_host_srv_serror_rate","flag","IDS_detection",

"Malware_detection","Ashula_detection","label","Source_IP_Address","Source_Port_Number",
    "Destination_IP_Address", "Destination_Port_Number","Start_Time","Protocoltype"]
#Dataset featureları okutma
df = pd.read_csv(dataset,header=None, names = features)
```

#Label Sayı İstatistiği - Bilinen Ve Bilinmeyen Saldırıların Sayısı Görmek İçin

print('Label distribution Dataset:')

print(df['label'].value_counts())

print()

```
Label distribution Dataset:
-1    88957
 1    11038
-2        5
Name: label, dtype: int64
```

1 : normal

-1: bilinen bir saldırının gözlemlendiğini

-2 : bilinmeyen bir saldırının gözlemlendiğini ifade eder.

#Malware Sayı İstatistiği - Hangi Saldırı Türlerini İçerdiğini Görmek İçin

print('Malware Types')

print(df['Malware_detection'].value_counts())

print()

```
Malware Types
0                            98267
0                             1696
Win.Worm.Downadup-5(1)           6
Win.Worm.Kido-173(1)             4
Win.Dropper.Agent-35454(1)       3
Win.Worm.Downadup-4(1)           3
Win.Worm.Downadup-11(1)          3
Win.Worm.Kido-423(1)             3
Win.Worm.Kido-392(1)             3
Win.Worm.Kido-266(1)             3
Win.Worm.Kido-37(1)              2
Win.Worm.Kido-160(1)             2
Win.Worm.Kido-355(1)             1
Win.Worm.Kido-197(1)             1
Win.Worm.Kido-185(1)             1
Win.Worm.Conficker-178(1)        1
Win.Worm.Downadup-110(1)         1
Name: Malware_detection, dtype: int64
```

#Servis Tipleri İstatistiği - Bağlantıların Kullandığı Servislerin Sayılarını Gösterir

print('Service Types')

print(df['service'].value_counts())

print()

```
Service Types
other    84355
dns       8672
ssh       4954
sip       1352
snmp       359
rdp        148
smtp       102
http        46
dhcp        10
ssl          2
Name: service, dtype: int64
```

#Flag - Bağlantının Durumlarının Sayılarını Görmek İçin

print('Flag')

print(df['flag'].value_counts())

print()

#Bağlanıların kullandıkları Kabuk kodların ve saldırı kodlarının tipleriye beraber sayısını gösterir

print('Ashula Detection')

print(df['Ashula_detection'].value_counts())

print()

#Bağlantıların Kullandıkları Protokol Tiplerinin Sayısını Gösterir

print('Protocoltype')

print(df['Protocoltype'].value_counts())

print()

#Geçmiş iki saniye içinde mevcut bağlantının kaynak ve hedef IP adreslerine sahip olan benzer bağlantıların sayısı

print('count')

print(df['count'].value_counts())

print()

#PCA öncesi istatistikler

summary_stats = df.describe()

print("\nPCA öncesi istatistikler :")

print(summary_stats)

#PCA sonrası istatistikler

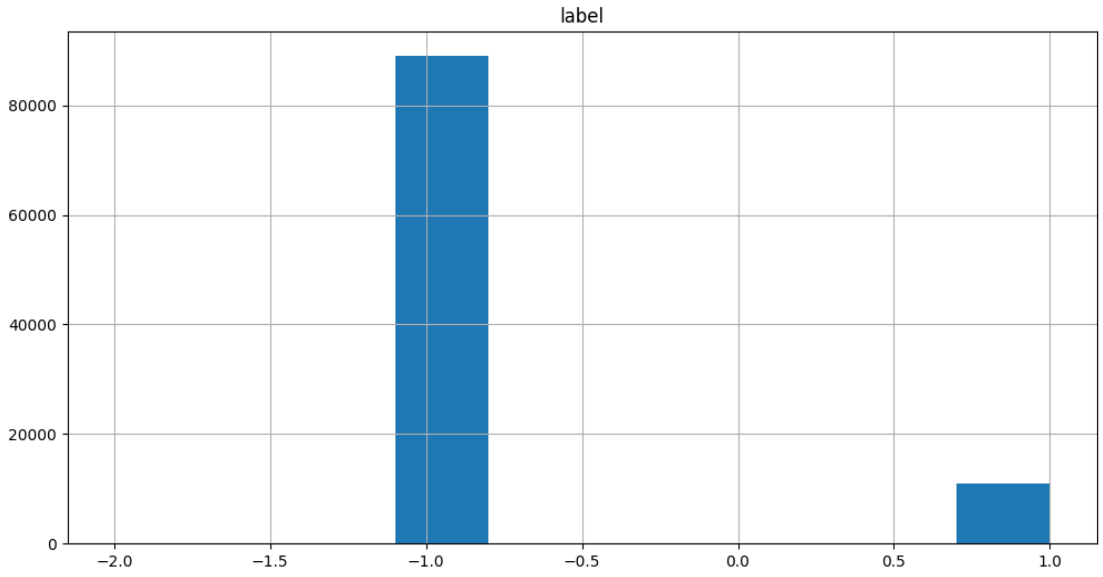
summary_stats_pca = df_pca.describe()

print("\nPCA sonrası istatistikler:")

print(summary_stats_pca)

#Label Görsel Grafiği

df[['label']].hist(bins=10, figsize=(12, 6))



' Denemeye Çalışıp Başarısız Olduğumuz Encoding'

RareEncoding

---

Deneme 1 :

```
nadir_degerler = df['serror_rate'].value_counts()[df['serror_rate'].value_counts() < 1].index


df['serror_rate'] = df['serror_rate'].apply(lambda x: 'Diğer' if x in nadir_degerler else x)


print(df['src_bytes'].value_counts())
```

---

Deneme 2 :

```
#Kütüphane Tanımlama

from feature_engine.encoding import RareLabelEncoder


# Rare Encoder Oluşturma

encoder = RareLabelEncoder(tol=0.03, n_categories=2, variables=['cabin', 'pclass',
'embarked'],

                           replace_with='Rare')


# Encoder Datasete Uydurma

encoder.fit(dataset)

encoder.encoder_dict_

# Datayı Dönüştürme

dataset = encoder.transform(dataset)
```

Normalizasyon - Dataset Start Time Üzerinden Normalize Edilmiş Halde İndirildi Fakat Aşağıdaki Kod ile duration Üzerinden Normalize Edilmeye Çalışıldı

Normalization (e.g., min-max scaling)

scaler = MinMaxScaler()

df[["duration","src_bytes","dst_bytes",

"count","serror_rate"]] = scaler.fit_transform(df[["duration","src_bytes","dst_bytes",
"count","serror_rate"]])

df[["duration","src_bytes","dst_bytes",

"count","serror_rate"]] *= 100