Port Scan Detection With ML

AISECLAB
(Artificial Intelligence Security Defense Lab)
aiseclab.org
CYBERBOT
Danışmanlar
Atakan Ak

DOĞUKAN ESEN ESMANUR HURMA İREM USLU ÖMER FARUK DİLBAZ

***Abstract*—Bu çalışma, bilgisayar ağlarındaki port taraması saldırılarını tespit etmek ve engellemek amacıyla firewall logları kullanılarak gerçekleştirilmiştir. Veri analizi ve makine öğrenmesi yöntemleriyle oluşturulan model, potansiyel tehditleri hızlı bir şekilde tanımlayarak sistem yöneticilerine uyarılar gönderme yeteneğine sahiptir. Bu proje, ağ güvenliğindeki önemli bir adım olarak bilgisayar ağlarını daha güvenli hale getirme hedefini taşımaktadır.**

***Index Terms*—Port Taraması,Ağ Güvenliği,Makine Öğrenmesi, Firewall Logları**

I. ÖZET

Günümüzde, bilgisayar ağları her zamankinden daha fazla hedef haline gelmiştir ve bu da ağ güvenliği önlemlerinin artırılmasını zorunlu kılmıştır. Port taraması, ağ güvenliğini tehdit eden en yaygın saldırı vektörlerinden biridir; bir saldırganın ağdaki hedef sistemlerin açık portlarını tespit etmeye çalışmasını içerir. Bu noktada, firewall logları önemli bir kaynak haline gelmektedir. Bu projenin amacı, firewall logları üzerinden port taraması faaliyetlerini tespit etmek ve önlemek için makine öğrenmesi tekniklerini kullanmaktır. Veri analizi, öznitelik mühendisliği ve makine öğrenmesi algoritmaları bir araya getirilerek, bu tehdidi engellemek hedeflenmektedir. Projemiz, firewall loglarından elde edilen verilerin temizlenmesi ve işlenmesiyle başlayacak, ardından önemli öznitelikler belirlenecektir. Bu aşamada, veri setinin özellikleri göz önünde bulundurularak en etkili makine öğrenmesi algoritması seçilecektir. Elde edilen model, firewall logları üzerinde uygulanarak potansiyel port taramalarının tespiti sağlanacaktır. Bu tespitler, sistem yöneticilerine hızlı bir şekilde uyarılar göndererek, potansiyel saldırılara karşı önlem almalarını sağlayacaktır. Son olarak, proje sonuçları değerlendirilecek, modelin performansı gözden geçirilecek ve gerektiğinde iyileştirmeler yapılacaktır. Bu projenin başarılı bir şekilde gerçekleştirilmesi, ağ güvenliği konusunda önemli bir katkı sağlayabilir ve bilgisayar ağlarını potansiyel tehditlere karşı daha korunaklı hale getirebilir.

II. VERI SETI ÖZELLIKLERI

A. Veri seti adı: normal-dataset.csv, malicious-dataset.csv

B. Veri seti kaynağı: https://github.com/gubertoli/ProbingDataset

C. Veri seti boyutları: normal-dataset.csv ($103094, 41$*), malicious-dataset.csv (*$193315, 42$*)*

D. Kolonların Özellikleri: Kolonların sırası ile içeriği hakkında ve veri tipi ile ilgili açıklamaları şu şekildedir.

- frame-info.encap-type: İleti paketleme tipini belirtir. Bu veri setinde 103094 gözlem bulunmaktadır ve her biri bir tam sayı değerini temsil eder.
- frame-info.time: İletinin zaman damgasını temsil eder. Bu sütunun veri tipi 'object' olarak belirtilmiş, bu nedenle zamanın uygun bir formatta olup olmadığı dikkatle incelenmelidir.
- frame-info.time-epoch: İletinin zaman damgasını epoch formatında gösterir. Bu değer ondalık bir sayıdır ve zamanı belirtir.
- frame-info.number: İletinin sıra numarasını temsil eder. Bu değer bir tam sayı olarak Belirtilmiş ve her bir gözlem için benzersiz bir numarayı ifade eder.
- frame-info-len: Iletinin toplam uzunluğunu belirtir. Bu değer bir tam sayıdır.
- frame-info-cap-len: Yakalanan paketin uzunluğunu temsil eder. Bu sütun, bir tam sayı değeri ile ifade edilmiştir.
- eth-type: Ethernet ¸çerçevesinin türünü belirtir. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-version: IP paketinin versiyonunu belirtir. Bu değer ondalık bir sayıdır.
- ip-hdr-len: IP baslığının uzunluğunu belirtir. Bu değer ondalık bir sayıdır.
- ip-tos: Hizmet tipini belirtir, ancak bu sütunda tüm değerlerin eksik olduğu görülüyor.
- ip-id: IP paketinin kimlik bilgisini temsil eder. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-flags: IP başlığındaki bayrakları belirtir. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-flags-rb: IP başlığındaki" Reserved Bit" değerini belirtir. Bu değer ondalık bir sayıdır.

- ip-flags-df: ”Don’t Fragment” bayrağını belirtir. Bu değer ondalık bir sayıdır.
- ip-flags-mf: ”More Fragments” bayrağını belirtir. Bu değer ondalık bir sayıdır.
- ip-frag-offset: Parçalanan IP paketlerinin ofset değerini belirtir. Bu değer ondalık bir sayıdır.
- ip-ttl: Time To Live (TTL) değerini belirtir. Bu değer ondalık bir sayıdır.
- ip-proto: Tasıma katmanı protokolünün numarasını belirtir. Bu değer ondalık bir sayıdır.
- ip-checksum: IP baslık kontrol toplamını temsil eder. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-src: Kaynak IP adresini temsil eder. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-dst: Hedef IP adresini temsil eder. Bu değer bir nesne (object) olarak belirtilmiş.
- ip-len: IP paketinin toplam uzunluğunu belirtir. Bu değer ondalık bir sayıdır.
- ip-dsfield: Differentiated Services Field değerini belirtir. Bu değer bir nesne (object) olarak belirtilmiş.
- tcp-srcport: Kaynak TCP portunu belirtir. Bu değer ondalık bir sayıdır.
- tcp-dstport: Hedef TCP portunu belirtir. Bu değer ondalık bir sayıdır.
- tcp-seq: TCP başlığındaki dizi numarasını temsil eder. Bu değer ondalık bir sayıdır.
- tcp-ack: Acknowledgement numarasını temsil eder. Bu değer ondalık bir sayıdır.
- tcp-len: TCP segmentinin uzunluğunu belirtir. Bu değer ondalık bir sayıdır.
- tcp-hdr-len: TCP baslığının uzunluğunu belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags: TCP baslığındaki bayrakları belirtir. Bu değer bir nesne (object) olarak belirtilmiş.
- tcp-flags-fin: FIN bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-syn: SYN bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-reset: RESET bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-push: PUSH bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-ack: ACK bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-urg: URGENT bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-flags-cwr: CWR bayrağını belirtir. Bu değer ondalık bir sayıdır.
- tcp-window-size: TCP penceresi boyutunu belirtir. Bu değer ondalık bir sayıdır.
- tcp-checksum: TCP baslık kontrol toplamını temsil eder. Bu değer bir nesne (object) olarak belirtilmiş.
- tcp-urgent-pointer: TCP aciliyet işaretçisini belirtir. Bu değer ondalık bir sayıdır.
- tcp-options-mss-val: TCP baslığındaki Maximum Segment Size (MSS) değerini temsil eder. Bu değer ondalık bir sayıdır.
- Label: Satırın zararlı mı yoksa normal mi olduğunu belirten etiket verisidir.

E. Hedef değişken: Son sütun olan label sütununda 21 farklı veri çeşidi bulunmaktadır ve hedefimiz bu tipleri doğru tahmin etmektir. Bu tipler:

- normal
- unicorn-conn
- zmap
- hping-syn
- masscan
- nmap-ack
- nmap-syn
- nmap-connect
- unicorn-syn
- hping-null
- hping-ack
- nmap-fin
- hping-fin
- nmap-window
- nmap-null
- hping-xmas
- nmap-maimon
- unicorn-fxmas
- unicorn-null
- nmap-xmas
- unicorn-xmas

Bu etiketler, normal işlemler haricinde nmap, unicornscan, hping3, zmap ve masscan araçlarıyla yapılmış taramaların adlandırmalarıdır.

III. VERİ ÖN İŞLEME

df-normal ve df-malicious adında dataframe oluşturuldu “normal-dataset.csv” dosyasından ve ”malicious-dataset.csv” dosyasından okunan verilerle dolduruldu. df-normal veri çerçevesine eksik olan ”label” sütunu eklendi ve bu sütun ”normal” olarak dolduruldu. Bu, normal verilerin etiketlenmesi(hedef değişken) için kullanılır.

pd.concat() fonksiyonu kullanılarak df-normal ve df-malicious veri çerçeveleri birleştirilir ve df adlı yeni bir veri çerçevesi oluşturuldu. Veri çerçevesinin satırları rastgele sıralandırıldı, indeks yapısı sıfırdan başlatıldı ve eski indeks kaldırıldı. Bu, verilerin daha rastgele bir dağılıma sahip olmasını ve analiz veya öğrenme işlemleri için daha uygun hale gelmesini sağlar.

Verilerin dağılımını incelediğimizde normal etiketli verilerin oldukça fazla olduğunu görmekteyiz.

checkdata() isimli eksik değerleri,ilk ve son 5 sütunu,veri tipi bilgilerini,boyutunu alacağımız bir fonksiyon oluşturuldu.Tamamı boş olan kolon silindi.
(grab col names) isimli bir fonksiyon daha oluşturuldu bu fonksiyonla beraber kaç tane kategorik,numeric,cat but car,num but cat olan değişkenlerin sayısını bulduk ve aşağıdaji sonuçları elde ettik.

Observations: 296409
Variables: 41
catcols: 19
numcols: 14
catbutcar: 8
numbutcat: 16

Kategorik ve sayısal değişkenlerin görselleştirme işlemi yapıldı.Hedef değişken olan label ile de kategorik değişkenler arasındaki ilişkiyi göstermek için görselleştirme işlemi yapıldı.

Hedef değişkeni görselleştirip aşağıdaki sonucu aldık.

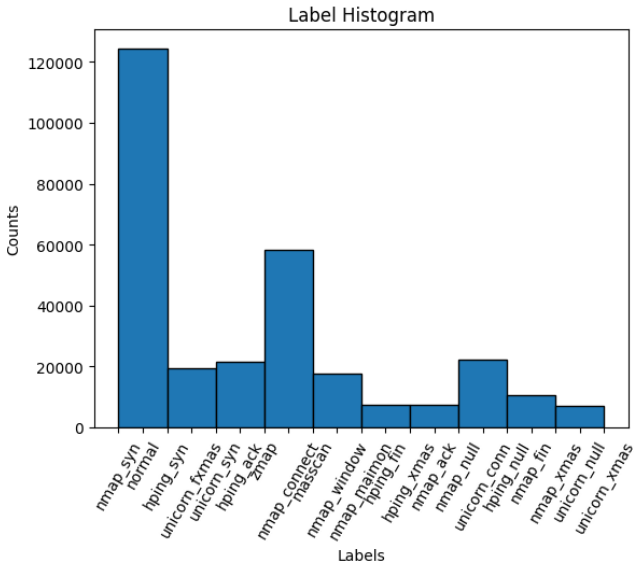

Fig. 1. Hedef değişken görseli.

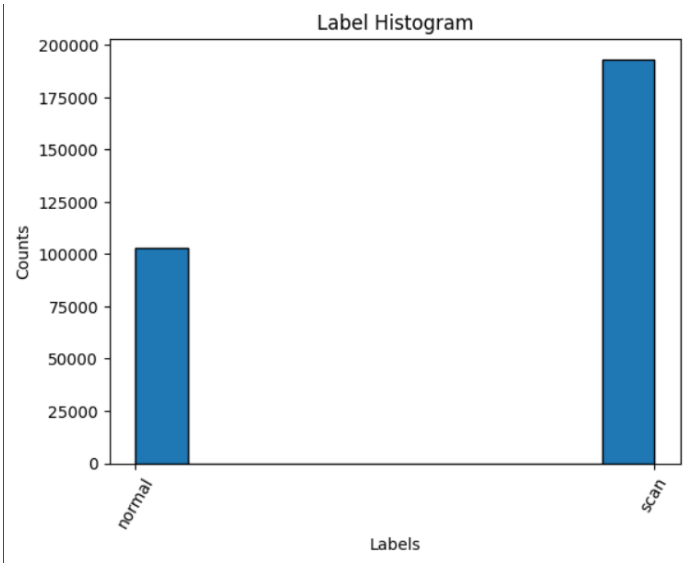

Fig. 2. Hedef değişken görseli.

Zaralı ve normal olmak üzere ikiye ayrılma işlemi yapıldı. 9 farklı zarar türünün birleştirilip tek bir kategori olarak sınıflandırıldı.

IV. YÖNTEM VE ALGORİTMALAR

Bu kısımda çalışmada kullanılan Makine Öğrenmesi yaklaşımlarından bahsedilmiştir.

A. Karar Ağacı: Veri madenciliği ve makine öğrenimi alanlarında kullanılan bir sınıflandırma ve regresyon yöntemidir. Bu algoritma, kararlar ve sonuçları temsil etmek için ağaç yapısı kullanır. Verileri analiz ederken, karar ağacı algoritması veri noktalarını belirli koşullar altında farklı sınıflara veya değerlere ayırır. Ağaç yapısı, veriyi bölme ve sınıflandırma işlemini hiyerarşik bir şekilde görselleştirir. Basit ve yorumlanabilir bir model üretme yeteneği nedeniyle tercih edilir.

B. SVM: Destek Vektör Makineleri (Support Vector Machines veya SVM), özellikle sınıflandırma ve regresyon problemleri için kullanılan güçlü bir makine öğrenimi algoritmasıdır. SVM, özellikle doğrusal ve non-doğrusal sınıflandırma problemlerinde etkili olan birçok avantaja sahiptir. SVM'nin avantajları arasında iyi genelleme yeteneği, doğrusal ve non-doğrusal problemlerde başarılı performans, overfitting'e karşı direnç ve destek vektörlerinin etrafındaki marjini maksimize etme eğilimi sayılabilir.

V. MODEL EĞİTİMİ VE SONUÇ

İlk olarak veri setindeki dengesizlik sorununu ele almak üzere sınıf ağırlıkları belirlenmiştir. Sınıf ağırlıkları, class-weight fonksiyonu tarafından, normal ve zararlı sınıflar arasındaki örnek sayılarına ve bir hiper parametre olan 'mu' değerine bağlı olarak özelleştirilmiştir.

Veri kümesi, test-size=0.15 olacak şekilde eğitim ve test kümelerine bölünmüştür. Eğitim aşamasında, Karar Ağacı sınıflandırıcı modeli önceki adımda belirlenen sınıf ağırlıklarını kullanarak eğitilmiştir.

Modelin performansı, normal ve zararlı durumları ayırt etme yeteneğini değerlendirmek için kullanılan çeşitli metriklerle ölçülmüştür. Test setinde elde edilen doğruluk, hassasiyet, duyarlılık ve F1 skoru gibi metrikler modelin sınıflar arası ayrım kabiliyetini ve genel performansını değerlendirmek için kullanılmıştır.

Ayrıca, eğitim ve test setlerinde elde edilen doğruluk sonuçları ayrı ayrı raporlanmıştır. Eğitim doğruluğu, modelin veri setine ne kadar iyi uyum sağladığını gösterirken, test doğruluğu, modelin yeni verilere ne kadar iyi genelleme yaptığını yansıtmaktadır.

```
Eğitim doğruluk: 0.9968553069278291
Test doğruluk: 0.997400833606519
```

Fig. 3. Eğitim ve test doğruluk sonuçları.

Son olarak, çapraz doğrulama ile elde edilen ortalama doğruluk, modelin performansının daha geniş bir perspektiften değerlendirilmesine olanak tanımaktadır. Bu adım, modelin

genel performansını belirleme konusunda daha güvenilir bir ölçü sunmaktadır.

```
Ortalama Doğruluk: 0.9589497756374092
```

Fig. 4. Ortalama doğruluk.

MODELİN WEB SİTESİNE EKLENMESİ

Günümüzde, veri analizi ve makine öğrenimi modelleri, çeşitli uygulamalarda kullanılmak üzere web servislerine entegre edilmektedir. Bu entegrasyon, kullanıcıların web uygulamaları aracılığıyla bu modellere erişmelerini ve sonuçları alabilmelerini sağlar. Ancak, bu tür bir entegrasyon sırasında güvenlik önlemlerinin alınması son derece önemlidir

Flask gibi bir mikroframework kullanarak, güvenlik açısından ölçeklenebilir bir web servisi oluşturmak mümkündür. Burada kullanıcıdan alınan parametreler sonucunda model tahmin işlemini gerçekleştirdi. Kullanıcıdan alınan parametreler güvensiz olduğu kabul edilerek gerekli doğrulama işlemleri yapıldı. Array veri tipine çevrilerek model tahmini yapıldıktan sonra frontend' e istek gönderildi.

Fig. 5. Kullanıcıdan değerlerin alınması ve çıktı.

SONUÇ VE GELECEK HEDEFLERİ

Bu çalışma, firewall logları üzerinden port taraması saldırılarını etkili bir şekilde tespit etme yeteneğini göstermiştir. Model, güvenlik önlemlerini hızla güçlendirmek için sistem yöneticilerine anında uyarılar gönderebilecek bir güvenlik aracı olarak kullanılabilir.

Gelecekte, bu çalışma temel alınarak daha geniş veri setleri ve farklı saldırı türleri üzerinde genişletilebilir. Daha da önemlisi, modelin hassasiyeti artırılabilir ve gerçek zamanlı tehdit tespiti için otomatik bir sistem olarak uygulanabilir. Ayrıca, farklı makine öğrenmesi algoritmalarının ve derin öğrenme tekniklerinin incelenmesi, modelin performansını daha da artırabilir.

Bu çalışmanın gelecekteki hedefi, bilgisayar ağlarını daha etkili bir şekilde koruyacak ve siber saldırılara karşı daha hızlı ve kesin tepkiler sağlayacak güçlü bir güvenlik çözümü geliştirmektir.