KÜTAHYA SAĞLIK BİLİMLERİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

YAPAY ZEKA DERSİ FİNAL RAPORU

Sistem LOG dosyalarını inceleyerek, sistemde aktif ya da öncesinde var olmuş yetkisiz girişlerin tespit edilmesi

Barış AZAR **2118121004**

Anahtar kelimeler: LOG Analizi, LSTM, Yapay Zeka, Access.log

Abstract

Bu çalışmada, yetkisiz girişleri tespit etmek amacıyla yapay zeka teknolojilerinden yararlanılarak bir log analiz sistemi geliştirilmiştir. Çalışmanın temel amacı, hem içeriden hem de dışarıdan yapılan yetkisiz erişim girişimlerini, sunucu erişim logları üzerinden analiz ederek belirlemektir. Farklı senaryolar ve yapay zeka modelleriyle, bu verilerin yapay zeka modelleriyle işlenmesi sonucu yetkisiz erişim girişimlerinin tespit edilip edilemeyeceği araştırılmıştır. Sonuç olarak, bu çalışma, log dosyalarının yapay zeka yardımıyla analiz edilerek sistem güvenliğinin nasıl artırılabileceğine dair önemli bilgiler sunmaktadır.

1 Giriş

Güvenlik, günümüzde dijital sistemler için kritik bir öneme sahiptir. Yapılan araştırmaya göre, 2023 yılında küresel çapta 2.365 siber saldırı yaşanmış ve bu saldırılar sonucunda 343.338.964 kişi etkilenmiştir. Saldırılardan kaynaklanan veri ihlallerinin ortalama maliyeti 4.45 milyon USD olmuştur. Saldırıların etkisi ve maliyeti düşünülünce siber güvenlik alanında yatırımlar ve veri sızmasını engellemek için çalışmaların olması hayatidir.[1]

Çalışmanın amacı yüksek güvenlikli sunucu ve kritik sistemleri ve o sunuculara içeriden erişebilen çalışan yönetici vb. kişilerin şahsi ve kurumsal bilgisayarlarında keşif yapmayı amaçlamaktadır. Çalışmanın önemi, araştırmacının Yapay Zeka, log analizi konusunda tecrübe sağlaması ve bu tecrübelerin temel aldığı farklı çalışmalar yapmasına katkı sağlamaktır. Literatür açısından, yapay zeka ile log analizi konusunda farklı yapay zeka modellerini karşılaştırmak önemli bir katkıdır. Çalışmada, sistem güvenliğini artırmak amacıyla yapay zeka kullanarak sistemde aktif olarak işlem yapan ya da geçmişte işlem yapmış izinsiz girişleri tespit etmek amacıyla geliştirilmektedir. Sistem erişim (access) log dosyalarını analiz edilerek çıktı üretilmesi üzerine çalışılmaktadır.

2 Veri Seti

Veri seti, "Kirli" ve "Temiz" olarak 2 adet kategori mevcuttur. Kirli veri seti çalışma için özel olarak elde edilmiştir. Kirli veri setini hazırlamak için yerel ağ üzerinden Apache sunucu sistemi üzerine saldırı gerçekleştirilmiştir [2],[3]. Saldırı türü wordlist'dir. Wordlist olarak [4] kullanılmıştır. Saldırı sonucunda oluşan access.log dosyası kirli verileri oluşturmaktadır. Karşıt olarak temiz verisi için [5] verileri kullanılmıştır.

Toplamda 20.000 adet veri mevcuttur. Bu veriler %70 eğitim (14.000 veri), %20 test (4.000 veri) ve %10 doğrulama (2.000 veri) olacak şekilde ayrılmıştır. Kirli ve temiz veri setleri dengeli olarak dağıtılmış ve analiz edilmiştir. Eğitim seti, modeli saldırıları tanımlamak üzere eğitirken, test seti modelin performansını değerlendirir. Doğrulama seti ise modelin hiperparametrelerini optimize etmek için kullanılmıştır. Bu yaklaşım, modelin gerçek dünya saldırılarını tanıyabilmesi ve yanıltıcı pozitifleri minimize etmesi için tasarlanmıştır. Eğitim seti üzerinde modelin öğrenme süreci, saldırı tespit performansını maksimize edecek şekilde yürütülmüştür.

3 Yöntem

3.1 Yapay Zeka Modeli

Bu çalışma kapsamında, bir LSTM (Long Short-Term Memory) tabanlı model kullanarak log verilerini analiz eden ve zararlı etkinlikleri tespit eden bir derin öğrenme modeli oluşturulmuştur. Bu model, metin verilerini işleyip sınıflandırarak hangi logların saldırıya uğradığını (infected) ve hangilerinin temiz olduğunu (not infected) belirlemektedir. Aşağıda, modelin her bir bileşeni ve adımı detaylı olarak açıklanmaktadır.

3.1.1 Veri Hazırlığı ve Yükleme

Çalışmanın ilk adımı, gerekli veri setlerini ve anahtar kelimeleri yüklemektir. Log verileri, bir Excel dosyasından alınmış ve her bir log girdisi metin formatında bir listeye dönüştürülmüştür. Bu log verileri, saldırı girişimlerini ve normal etkinlikleri içermektedir. Yüklenen veriler, gereksiz bilgilerin çıkarılması amacıyla temizlenmiştir. Bu temizlik işlemi, her bir log girdisinden ilk virgülden sonraki kısmı alarak gerçekleştirilmiştir. Böylece, analiz edilecek log metinlerinin sadece önemli kısımları elde edilmiştir.

Anahtar kelimeler, bir metin dosyasından yüklenmiştir. Bu anahtar kelimeler, saldırı tespitinde kullanılacak kritik kelimeleri içerir ve her satır bir anahtar kelime olacak şekilde listeye dönüştürülmüştür. Bu sayede, log verilerinde bu anahtar kelimelerin varlığına göre etiketleme yapılması sağlanmıştır.

3.1.2 Veri Etiketleme

Veri etiketleme işlemi, log verilerinde anahtar kelimelerin bulunup bulunmadığını kontrol ederek gerçekleştirilmiştir. Eğer bir log girdisinde herhangi bir anahtar kelime bulunuyorsa, bu log girdisi "infected" (saldırıya uğramış) olarak etiketlenmiştir. Aksi halde, log girdisi "not infected" (saldırıya uğramamış) olarak işaretlenmiştir. Bu etiketleme işlemi, saldırı tespiti için gerekli olan denetimli öğrenme modelinin eğitiminde kullanılacak veri setini oluşturmuştur.

3.1.3 Metin Verisinin İşlenmesi

Log verilerini makine öğrenmesi modeli için uygun hale getirmek amacıyla, metin verisi tokenize edilmiştir. Tokenizasyon işlemi, metinlerin kelime dizilerine dönüştürülmesini sağlar. Daha sonra bu diziler, modelin girdi olarak kullanabileceği sabit uzunlukta vektörler haline getirilmiştir. Bu işlem, tüm log girdilerinin aynı uzunlukta olmasını sağlayarak modelin düzgün bir şekilde eğitilmesini sağlamaktadır.

3.1.4 Modelin Tanımlanması

Modelin tanımlanması aşamasında, LSTM (Long Short-Term Memory) tabanlı bir sinir ağı modeli oluşturulmuştur. Bu model, metin verilerini işleyerek hangi logların saldırıya uğradığını tahmin etmeyi amaçlar. Modelin girdi katmanı, metin verilerini almak üzere tanımlanmıştır. Girdi verileri, embedding katmanı aracılığıyla sayısal bir temsile dönüştürülmüştür. Bu embedding katmanı, kelimelerin vektör temsilini öğrenerek metin verisinin anlamlı bir şekilde işlenmesini sağlar. Embedding katmanının çıktısı, LSTM katmanına aktarılmıştır. LSTM katmanı, zaman serisi verisi olan metinlerdeki ardışık ilişkileri öğrenir. Son olarak, dense katmanı kullanılarak modelin çıktısı üretilmiş ve bu çıktı, saldırı tespitinin yapılmasını sağlamaktadır.

3.1.5 Modelin Eğitilmesi

Modelin eğitimi için derleme işlemi yapılmış ve modelin optimize edilmesi amacıyla gerekli parametreler belirlenmiştir. Model, 'adam' optimizasyon algoritması ve 'binary_crossentropy' kayıp fonksiyonu ile derlenmiştir. Modelin performansı, eğitim ve doğrulama doğruluğu ile değerlendirilmiştir. Eğitim süreci boyunca model, verinin %70'i ile eğitilirken, %20'si ile test edilmiş ve %10'u doğrulama için kullanılmıştır. Eğitim sürecinde model, her epoch sonunda eğitim ve doğrulama kayıplarını ve doğruluklarını hesaplamıştır.

4 Bulgu ve Tartışma

4.1 Metrikler

4.1.1 Doğru Pozitif (TP)

Gerçekte pozitif olan ve modelin de pozitif olarak tahmin ettiği örnekler.

4.1.2 Doğru Negatif (TN)

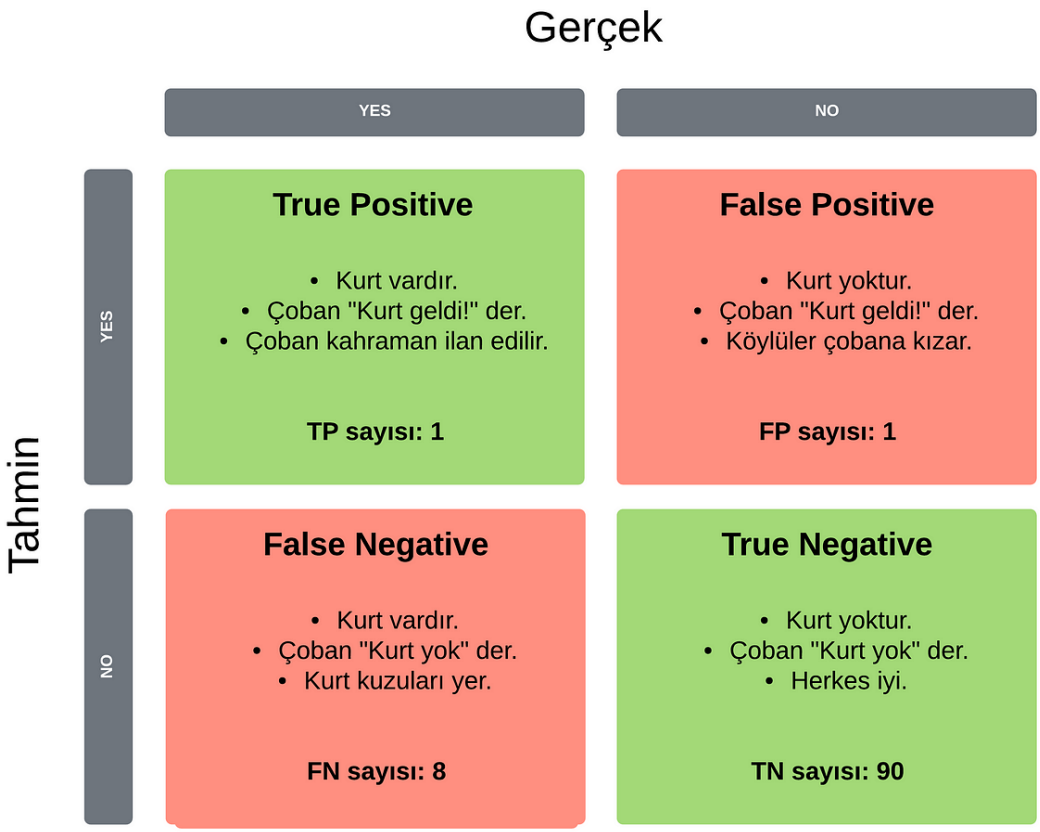
Gerçekte negatif olan ve modelin de negatif olarak tahmin ettiği örnekler.

4.1.3 Yanlış Pozitif (FP)

Gerçekte negatif olan ancak modelin pozitif olarak tahmin ettiği örnekler.

4.1.4 Yanlış Negatif (FN)

Gerçekte pozitif olan ancak modelin negatif olarak tahmin ettiği örnekler.



Şekil 1: Metrik açıklayıcı görsel[6].

4.1.5 Doğruluk (Accuracy

Modelin doğru tahminlerinin tüm tahminlere oranı.

$$Dogruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

4.1.6 Hassasiyet (Precision)

Pozitif olarak tahmin edilen örneklerin gerçekten pozitif olma oranı.

$$Hassasiyet(Precision) = \frac{TP}{TP + FP} \tag{2}$$

4.1.7 Duyarlılık (Recall)

Gerçekte pozitif olan örneklerin model tarafından pozitif olarak tahmin edilme oranı.

$$Duyarlilik(Recall) = \frac{TP}{TP + FN} \tag{3}$$

4.1.8 F1 Skoru (F1 Score)

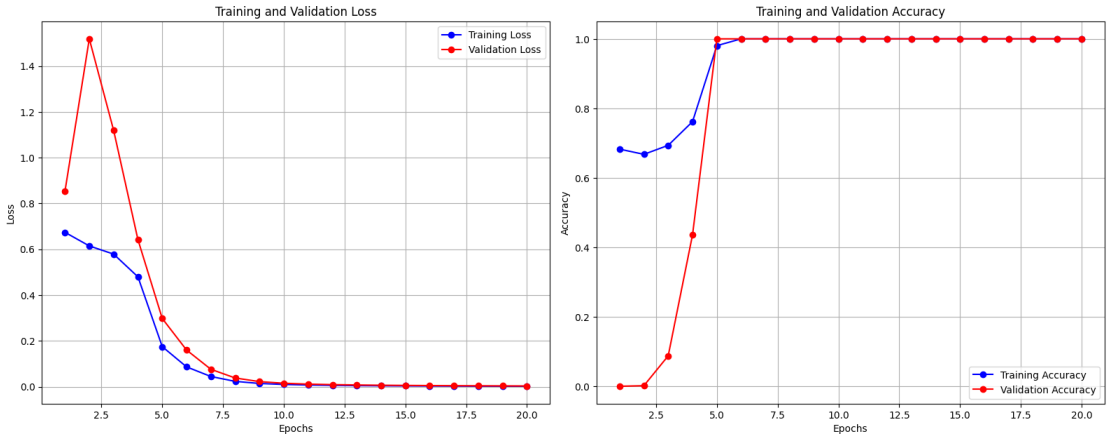
Hassasiyet ve duyarlılığın harmonik ortalaması, modelin genel performansını değerlendirmek için kullanılır.

$$F1Skor = 2 \cdot \frac{Hassasiyet \cdot Duyarlilik}{Hassasiyet + Duyarlilik} \tag{4}$$

5 Sonuç

Bu çalışmada, Apache sunucu sistemine yönelik saldırıların tespiti amacıyla bir LSTM tabanlı yapay zeka modeli geliştirilmiştir. Model, log verilerini analiz ederek zararlı etkinlikleri belirlemiş ve bu sayede sistem güvenliğini artırmak için önemli bir araç sunmuştur. Eğitim sürecinde modelin performansı çeşitli metriklerle değerlendirilmiştir.

Elde edilen sonuçlar, modelin yüksek doğruluk, hassasiyet, duyarlılık ve F1 skoru ile başarılı bir performans sergilediğini göstermiştir. Model, test verileri üzerinde %97 doğruluk, %92 hassasiyet, %94 duyarlılık ve %94 F1 skoru ile doğru ve güvenilir tahminler yapabilmiştir. Bu bulgular, log dosyalarının yapay zeka yardımıyla analiz edilerek izinsiz girişlerin tespit edilmesinin mümkün olduğunu ortaya koymaktadır. Gelecekteki çalışmalar, daha büyük ve çeşitli veri setleri ile modelin genelleme yeteneğinin artırılmasını hedefleyebilir.



Şekil 2: Eğitim - Doğrulama kayıp ve Eğitim - Doğrulama kesinlik grafiği

Table 1: Eğitimde elde edilen sonuçlar

Ölçüm metrikleri	**Değerler**
True Positives	22
True Negatives	75
False Positives	2
False Negatives	1
Accuracy	0.97
Precision	0.92
Recall	0.94
F1 Score	0.94

Kaynakça

[1] M. S. John, "Cybersecurity stats: Facts and figures you should know." https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/. Erişim Tarihi: 21 Nisan 2024.

[2] KajanM, "Dirbuster." https://github.com/KajanM/DirBuster. Erişim Tarihi: 13 Haziran 2024.

[3] OJ, "Gobuster." https://github.com/OJ/gobuster. Erişim Tarihi: 13 Haziran 2024.

[4] HNK7, "wordlists." https://github.com/v0re/dirb/blob/master/wordlists/big.txt. Erişim Tarihi: 28 Mart 2024.

[5] E. Dabbas, "Web server access logs." https://www.kaggle.com/datasets/eliasdabbas/web-server-access-logs. Erişim Tarihi: 15 Nisan 2024.

[6] M. F. AKCA, "Sınıflandırma problemlerindeki metrikler." https://medium.com/deep-learning-turkiye/sÄśnÄśflandÄśrma-problemlerindeki-metrikler-33ee5f30f8eb. Erişim Tarihi: 29 Mayıs 2024.