BL226 – Proje Uygulamaları Dersi

1. Ara Rapor

Teslim Tarihi: 04.04.2025
Proje Adı: Yapay Zeka Destekli Siber Saldırı Tespit ve Önleme Sistemi (AI-IDS/IPS)

1. Grup Bilgileri

Adı Soyadı	Görevi
Yusuf Orçan	Proje Yöneticisi, Model Geliştirici
Mehmet Can Oğuz	Ağ Trafik Analizi, Veritabanı Yönetimi
Samed Kabak	Veri Hazırlama, Görselleştirme ve Web Arayüzü

Not: Mehmet Can, Yusuf Orçan'ın görevlerini devralarak model geliştirme sürecine destek verecektir.

2. Proje Amacı ve Kapsamı

Bu projenin amacı, yapay zeka destekli bir sistem geliştirerek ağ trafiği üzerinde gerçekleşen anormal davranışları tespit etmek ve olası siber saldırılara karşı önleyici adımlar atmaktır. IDS/IPS (Intrusion Detection/Prevention System) yaklaşımı ile sistem, güvenlik açıklarını analiz ederek gerçek zamanlı müdahale kabiliyeti sağlayacaktır.

---

3. Şu Ana Kadar Yapılanlar

- **Veri Seti Araştırması ve İndirme:** NSL-KDD ve CICIDS 2017 veri setleri incelendi ve indirildi.
- **Veri Ön İşleme:** Null veriler temizlendi, kategorik veriler sayısallaştırıldı.
- **Temel Model Geliştirme:** Scikit-learn ile Naive Bayes ve Decision Tree algoritmalarıyla ilk modeller denendi.
- **Ağ Trafiği Yakalama:** Wireshark ile örnek trafik verileri elde edildi.
- **Ekip İçi Görev Paylaşımı Yapıldı:** Her üyenin odaklandığı alanlar belirlendi.

---

4. Karşılaşılan Zorluklar

- Veri setlerinin boyutu ve dengesiz sınıf dağılımları sebebiyle model eğitimi süresi uzadı.
- CICIDS 2017 veri setinin karmaşık yapısı nedeniyle analiz süreci zorlaştı.
- Ağ trafiğinin canlı olarak analiz edilmesi için gerekli ortam (virtual network) kurulumu zaman aldı.

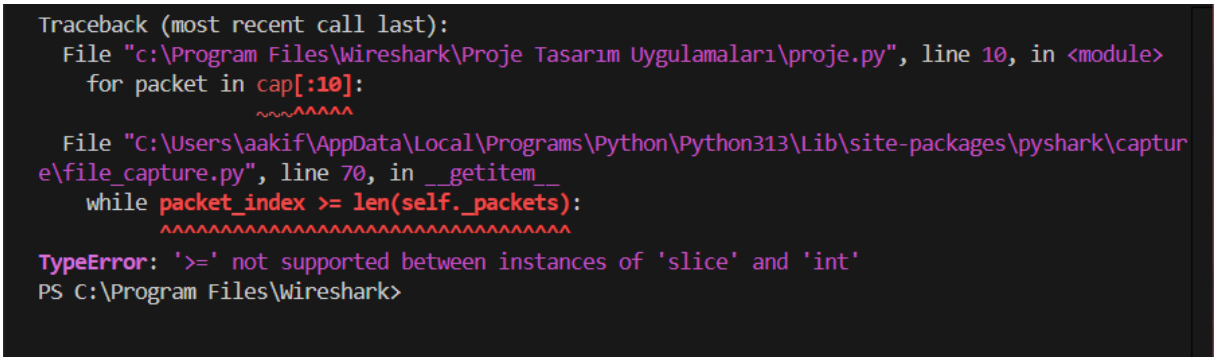
---

5. Gelecek Adımlar

- Derin öğrenme temelli modellerin (LSTM, Autoencoder) geliştirilmesi.
- Flask tabanlı web arayüzünün tamamlanması.
- MongoDB veritabanı ile kayıt sistemi oluşturulması.
- Gerçek zamanlı ağ trafiği analizi entegrasyonu yapılması.
- Test ve performans karşılaştırmalarının yapılması.

---

6. Ekran Görüntüleri



```
xe "c:/Program Files/Wireshark/Proje Tasarım Uygulamaları/pickle.py"
Traceback (most recent call last):
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\pickle.py", line
 1, in <module>
    import pickle
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\pickle.py", line
 4, in <module>
    with open("model.pkl", "rb") as model_file:
         ~~~~^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'model.pkl'
PS C:\Program Files\Wireshark>
```

```
dir : Cannot find path 'C:\Program Files\Wireshark\model.pkl' because it does
 not exist.
At line:1 char:1
+ dir model.pkl
+ ~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Program Files\Wireshark\mo
   del.pkl:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.Get
   ChildItemCommand
```

```
xe "c:/Program Files/Wireshark/Proje Tasarım Uygulamaları/model.py"
Traceback (most recent call last):
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\model.py", line
1, in <module>
    import pickle
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\pickle.py", line
 4, in <module>
    with open("model.pkl", "rb") as model_file:
         ~~~~^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'model.pkl'
PS C:\Program Files\Wireshark>
```

Python
psql
powershell
Python: app
Python: pick...
Python: mo...

```
6 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/generic.py in __array__(self, dtype, copy)
   2151     ) -> np.ndarray:
   2152         values = self._values
-> 2153         arr = np.asarray(values, dtype=dtype)
   2154         if (
   2155             astype_is_view(values.dtype, arr.dtype)

ValueError: could not convert string to float: 'normal'
```

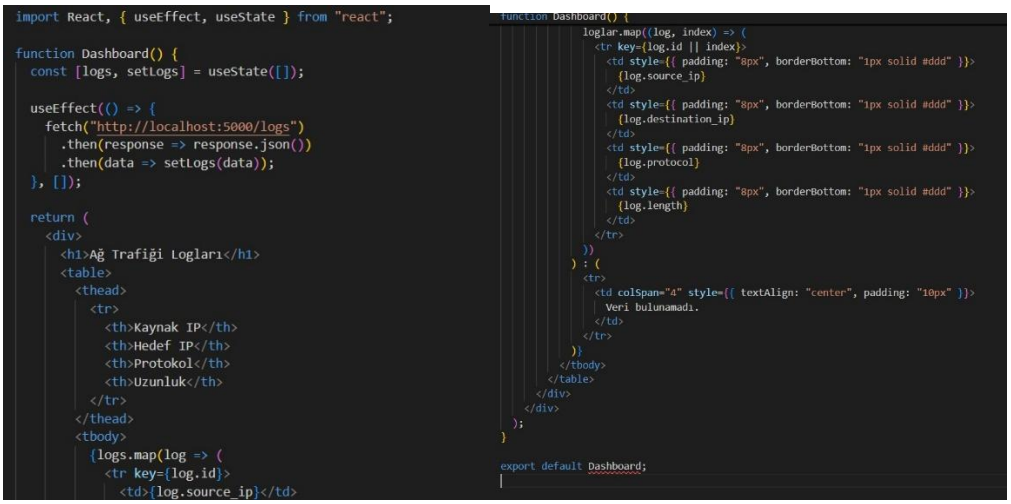
```
Traceback (most recent call last):
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\app.py", line 21, in <module>
    model = pickle.load(model_file)
_pickle.UnpicklingError: invalid load key, '\x0c'.
PS C:\Program Files\Wireshark> dir random_forest_model.pkl
dir : Cannot find path 'C:\Program Files\Wireshark\random_forest_model.pkl' because it does not exist.
At line:1 char:1
+ dir random_forest_model.pkl
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Program File...orest_model.pkl:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand
```

```
Traceback (most recent call last):
  File "c:\Program Files\Wireshark\Proje Tasarım Uygulamaları\app.py", line 21, in <module>
    model = pickle.load(model_file)
_pickle.UnpicklingError: invalid load key, '\x0c'.
PS C:\Program Files\Wireshark>
```

```
  File "C:\Users\aakif\AppData\Local\Programs\Python\Python313\Lib\site-packages\requests\api.py", line 59, in request
    return session.request(method=method, url=url, **kwargs)
           ~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\aakif\AppData\Local\Programs\Python\Python313\Lib\site-packages\requests\sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
  File "C:\Users\aakif\AppData\Local\Programs\Python\Python313\Lib\site-packages\requests\sessions.py", line 703, in send
    r = adapter.send(request, **kwargs)
  File "C:\Users\aakif\AppData\Local\Programs\Python\Python313\Lib\site-packages\requests\adapters.py", line 700, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPConnectionPool(host='127.0.0.1', port=5000): Max retries exceeded with url: /predict (Caused by NewConnectionError(
'<urllib3.connection.HTTPConnection object at 0x000001ED48E4F0E0>: Failed to establish a new connection: [WinError 10061] Hedef makine etkin olarak reddetti
ğinden bağlantı kurulamadı'))
PS C:\Program Files\Wireshark>
```

```jsx
import React, { useEffect, useState } from "react";

function Dashboard() {
  // logs: API'den çekilen log verilerini saklayan state
  // error: API isteğinde hata olup olmadığını saklayan state
  const [loglar, setLoglar] = useState([]);
  const [hata, setHata] = useState(null);

  useEffect(() => {
    // API'den log verilerini çeken fonksiyon
    const loglariGetir = async () => {
      try {
        const yanit = await fetch("http://localhost:5000/logs"); // API'ye istek at
        if (!yanit.ok) {
          throw new Error(`HTTP Hatası: ${yanit.status}`); // Hata durumu kontrolü
        }
        const veri = await yanit.json(); // JSON formatına çevir
        setLoglar(veri); // State'i güncelle
      } catch (err) {
        setHata(err.message); // Hata mesajını state'e kaydet
      }
    };

    loglariGetir(); // İlk yüklemede veriyi getir

    // Belirli aralıklarla API'den veri çek (5 saniyede bir)
    const interval = setInterval(loglariGetir, 5000);

    return () => clearInterval(interval); // Bileşen kaldırıldığında interval'i temizle
  }, []);

  return (
    <div style={{ padding: "20px", fontFamily: "Arial, sans-serif" }}>
      <h1>Ağ Trafiği Logları</h1>
```

```jsx
{/* Hata varsa ekrana yazdır */}
{hata && <p style={{ color: "red" }}>Hata: {hata}</p>}

<div style={{ overflowX: "auto" }}>
  <table
    style={{
      width: "100%",
      borderCollapse: "collapse",
      marginTop: "10px",
    }}
  >
    <thead>
      <tr style={{ backgroundColor: "#f4f4f4", textAlign: "left" }}>
        <th style={{ padding: "8px", borderBottom: "2px solid #ddd" }}>
          Kaynak IP
        </th>
        <th style={{ padding: "8px", borderBottom: "2px solid #ddd" }}>
          Hedef IP
        </th>
        <th style={{ padding: "8px", borderBottom: "2px solid #ddd" }}>
          Protokol
        </th>
        <th style={{ padding: "8px", borderBottom: "2px solid #ddd" }}>
          Uzunluk
        </th>
      </tr>
    </thead>
    <tbody>
      {/* loglar dizisinde veri varsa her birini tabloya ekle, yoksa "Veri bulunamadı" mesajı göster */}
      {loglar.length > 0 ? (
        loglar.map((log, index) => (
          <tr key={log.id || index}>
            <td style={{ padding: "8px", borderBottom: "1px solid #ddd" }}>
              {log.source_ip}
```



```jsx
import React, { useEffect, useState } from "react";

function Dashboard() {
  const [logs, setLogs] = useState([]);

  useEffect(() => {
    fetch("http://localhost:5000/logs")
      .then(response => response.json())
      .then(data => setLogs(data));
  }, []);

  return (
    <div>
      <h1>Ağ Trafiği Logları</h1>
      <table>
        <thead>
          <tr>
            <th>Kaynak IP</th>
            <th>Hedef IP</th>
            <th>Protokol</th>
            <th>Uzunluk</th>
          </tr>
        </thead>
        <tbody>
          {logs.map(log => (
            <tr key={log.id}>
              <td>{log.source_ip}</td>
```

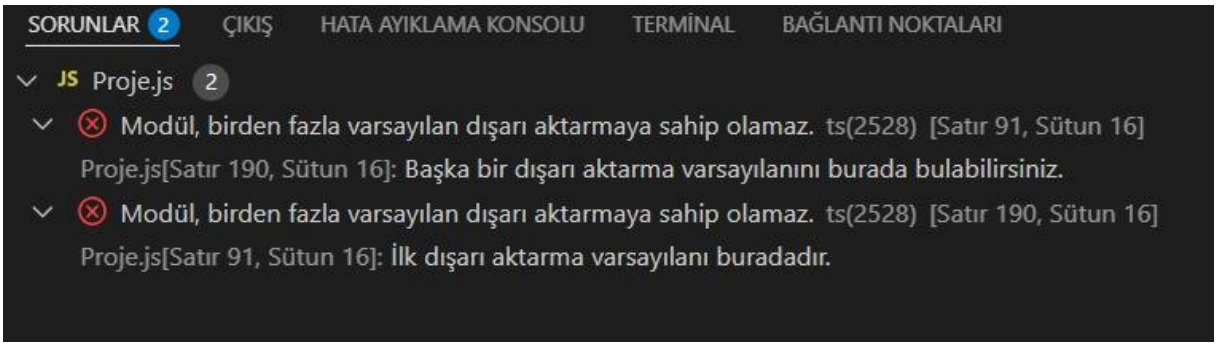
```jsx
function Dashboard() {
                loglar.map((log, index) => (
                  <tr key={log.id || index}>
                    <td style={{ padding: "8px", borderBottom: "1px solid #ddd" }}>
                      {log.source_ip}
                    </td>
                    <td style={{ padding: "8px", borderBottom: "1px solid #ddd" }}>
                      {log.destination_ip}
                    </td>
                    <td style={{ padding: "8px", borderBottom: "1px solid #ddd" }}>
                      {log.protocol}
                    </td>
                    <td style={{ padding: "8px", borderBottom: "1px solid #ddd" }}>
                      {log.length}
                    </td>
                  </tr>
                ))
              ) : (
                <tr>
                  <td colSpan="4" style={{ textAlign: "center", padding: "10px" }}>
                    Veri bulunamadı.
                  </td>
                </tr>
              )}
            </tbody>
          </table>
        </div>
      </div>
    );
}

export default Dashboard;
```

Ağ Trafiği Logları

Hata: Failed to fetch

Kaynak IP	Hedef IP	Protokol	Uzunluk
Veri bulunamadı.			

```jsx
          <tbody>
            {logs.map(log => (
              <tr key={log.id}>
                <td>{log.source_ip}</td>
                <td>{log.destination_ip}</td>
                <td>{log.protocol}</td>
                <td>{log.length}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    );
}

export default Dashboard;
```



---

7. Ekran Kaydı ve YouTube Linki

Aşağıda, projenin geldiği aşamaları detaylı şekilde anlattığımız ekran kaydının YouTube linki yer almaktadır:

YouTube Linki:

https://youtu.be/8Gf4bFSM7zQ

https://youtu.be/ihGElqhs3IQ