• **Ders/Dönem:** FET312 Derin Öğrenme / 2025 Güz Dönemi.

• **Proje Başlığı:** Derin Sinir Ağları ile Yapay Zeka Tabanlı Saldırı Tespit Sistemi Geliştirilmesi.

• **Ekip Bilgileri:** Ala Buveıdanı 24040301051 alabuveidani@stu.topkapi.tr

• **GitHub/Repo Bağlantısı:** https://github.com/AlaBuveidani/NeuralWall.

DERİN ÖĞRENME PROJE İLERLEME RAPORU

1) Problem Tanımı & Motivasyon

İş / Bilimsel Soru

Ağ sistemleri üzerinden geçen trafiğin artmasıyla birlikte, zararlı bağlantıların manuel veya kural tabanlı yöntemlerle tespit edilmesi zorlaşmıştır. Bu durum, ağ güvenliğini tehdit eden saldırıların erken aşamada tespit edilememesine yol açmaktadır.

Bu projede temel araştırma sorusu şudur:
“Güvenlik duvarı üzerinden geçen bir ağ bağlantısı, derin öğrenme yöntemleri kullanılarak normal mi yoksa saldırı mı olarak doğru şekilde sınıflandırılabilir mi?”

Görev Türü

- **İkili Sınıflandırma (Binary Classification)**

Hedef Değişken

- **Orijinal veri seti çok sınıflı bir yapıya sahiptir. Bu projede problem sadeleştirilerek ikili sınıflandırmaya dönüştürülmüştür:**
  - **Normal (0): Güvenli ağ bağlantıları**
  - **Attack (1): Zararlı veya saldırı içeren bağlantılar (pozitif sınıf)**

Başarı Kriterleri

- Accuracy ≥ %90
- F1-Score ≥ 0.85
- Attack sınıfı için yüksek Recall

2) Proje Yönetimi

Zaman Çizelgesi (Milestones)

- **1. Hafta:** Veri seti seçimi ve problem tanımı
- **2. Hafta:** Veri keşfi (EDA) ve ön işleme
- **3–4. Hafta:** Base model geliştirme

- **5–6. Hafta:** Derin öğrenme modeli (MLP) tasarımı
- **7. Hafta:** Performans değerlendirme
- **8. Hafta:** Rapor ve sunum

Roller ve Sorumluluklar

- Veri Ön İşleme: Öğrenci 1
- Base Model: Öğrenci 1
- Derin Öğrenme Modeli: Öğrenci 1
- Raporlama: Öğrenci 1

Çıktılar

- Proje raporu (PDF)
- Jupyter Notebook kodları
- Eğitim ve test sonuçları
- Sunum slaytları

3) İlgili Çalışmalar (Mini Literatür Taraması)

Literatürde ağ saldırı tespiti için makine öğrenmesi ve derin öğrenme yöntemlerinin yaygın olarak kullanıldığı görülmektedir. Özellikle MLP, CNN ve RNN tabanlı yaklaşımlar, geleneksel yöntemlere kıyasla daha yüksek doğruluk oranları sunmaktadır.

Bu proje, **Internet Firewall Data** veri seti üzerinde **binary sınıflandırma yaklaşımını** benimseyerek, saldırı tespit problemine daha sade ve uygulanabilir bir çözüm sunmayı hedeflemektedir.

4) Veri Açıklaması ve Yönetimi

Veri Kümesi Açıklaması

- **Adı:** Internet Firewall Data
- **Kaynak:** UCI Machine Learning Repository
- **Toplam Kayıt:** 65.532
- **Özellik Sayısı:** 12
- **Lisans:** Akademik kullanım
- **Veri Türü: Sayısal**

Kullanılan 12 Özellik (Features)

1. **Source Port**
2. **Destination Port**
3. **NAT Source Port**
4. **NAT Destination Port**
5. **Protocol**
6. **IP Protocol**
7. **Bytes**
8. **Packets**
9. **Duration**
10. **NAT Bytes**
11. **NAT Packets**
12. **Flags**

Sınıf Dağılımı (Binary)

- **Normal:** ≈ %70
- **Attack:** ≈ %30

Veri seti dengesiz (imbalanced) bir yapıya sahiptir.

Etik, Gizlilik ve Önyargı

Veri seti anonimleştirilmiş ağ trafiği kayıtlarından oluşmakta olup kişisel veri içermemektedir.

5) Class Imbalance Yönetimi

Sınıf dengesizliği problemini azaltmak amacıyla aşağıdaki yöntemler uygulanacaktır:

- **Stratified Train-Test Split**
- **Class Weighting:** Kayıp fonksiyonunda Attack sınıfına daha yüksek ağırlık verilmesi
- **Recall Odaklı Değerlendirme**

Bu yaklaşımlar, saldırıların yanlış negatif olarak sınıflandırılmasını azaltmayı hedeflemektedir.

6) Yöntemler ve Mimari

Base Modeller

Derin öğrenme modeli öncesinde aşağıdaki temel modeller geliştirilmiştir:

- **Logistic Regression:** Referans performans modeli
- **Random Forest:** Doğrusal olmayan ilişkilerin yakalanması

- Support Vector Machine (SVM): Yüksek boyutlu veriler için karşılaştırma modeli

Derin Öğrenme Modeli (MLP)

- Giriş Katmanı: 12 nöron
- Gizli Katmanlar:
  - Dense(64) – ReLU
  - Dense(32) – ReLU
- Dropout: 0.3
- Çıkış Katmanı: Dense(1) – Sigmoid
- **Kayıp Fonksiyonu:** Binary Cross-Entropy
- **Optimizasyon:** Adam (lr = 0.001)

7) Deney Tasarımı

- **Train / Test Oranı:** %80 / %20
- **Validasyon:** Hold-out
- **Değerlendirme Metrikleri:**
  Accuracy, Precision, Recall, F1-Score, Confusion Matrix

8) Kullanılan Araçlar ve Frameworkler

- **Python:** 3.x
- **Kütüphaneler:**

    - NumPy
    - Pandas
    - scikit-learn
    - PyTorch
- **Ortam:** Google Colab / Local GPU

8) Temel Kaynaklar (Core References)

- I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- R. Buczak, E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys*, 2016.
- S. Sharafaldin et al., "Toward Generating a New Intrusion Detection Dataset," *ICISSP*, 2018.
- J. Kim, H. Kim, "An Effective Intrusion Detection Classifier Using LSTM," *Applied Sciences*, 2019.
- C. Zhang et al., "Network Intrusion Detection Using Deep Learning," *IEEE Access*, 2020.