

**KARADENİZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİLGİSAYAR GÜVENLİK TEMELLERİ DERSİ**

K-MEANS İLE ANOMALİ TESTİ

ARİF KAAN BİTGİN – 402514

TUFAN İŞKÜZAR – 417314

MEHMET TUNAHAN TÜYLÜ - 402535

DR. ÖĞR. ÜYESİ BESTE ÜSTÜBİOĞLU

## İÇİNDEKİLER

1. **Giriş**
2. **Literatür Özeti**
3. **K-Means ve NSL-KDD Veri Seti**  
   3.1. NSL-KDD Veri Setinin Özellikleri  
   3.2. Saldırı Türleri ve Normal Trafik  
   3.3. Örnek Veri Analizi (İstatistiksel Dağılımlar)
4. **K-Means İle NSL-KDD Veri Setinin Eğitilmesi**  
   4.1. Veri Ön İşleme (Öznitelik Mühendisliği)  
   4.2. Model Eğitimi ve Parametre Seçimi  
   4.3. Model Performansı: Confusion Matrix, ROC Eğrisi, Diğer Metrikler
5. **Scapy ile Ağın İzlenmesi**  
   5.1. Canlı Trafik Toplama Yöntemi  
   5.2. SYN, ICMP ve Port Tarama Özellikleri  
   5.3. Ek Senaryolar (Wireshark PCAP Dosyaları vb.)
6. **Gerçek Zamanlı Anomali Tespiti**  
   6.1. K-Means Tahmini  
   6.2. Custom Saldırı Tespit Fonksiyonları  
   6.3. Final Label, Loglama ve Grafikle Raporlama
7. **Deneysel Bulgular ve Senaryolar**  
   7.1. Farklı SYN Flood Hızları  
   7.2. Farklı Eşik (Threshold) Değerleri  
   7.3. Ayrıntılı Tablo ve Grafiklerle Karşılaştırma  
   7.4. Gelişmiş Saldırı Senaryoları (Slowloris, Mixed Flood vb.)
8. **Öneriler ve Gelecek Çalışmalar**
9. **Sonuç ve Değerlendirme**
10. **Kaynakça**
11. **Ekler (Appendix)**

## 1. GİRİŞ

Ağ güvenliği alanında, **anomalileri** (kural dışı, saldırı amaçlı trafik) mümkün olduğunca **erken** ve **doğru** yakalamak, kurumların en önemli gereksinimlerinden biridir. Klasik saldırı tespit sistemleri (IDS) genellikle **imza tabanlı** çalışır ve bilinen saldırı paternlerini yakalamakta başarılıdır. Ancak **yeni** veya **gizlenmiş** saldırı türlerine karşı yetersiz kalabilir. Makine öğrenmesi (özellikle **denetimsiz** yöntemler) bu açığı kapatmakta son yıllarda oldukça popüler hâle gelmiştir.

Bu proje kapsamında, **K-Means** algoritması kullanılarak hem **NSL-KDD** veri seti üzerinde **offline** eğitim hem de **Scapy** ile **online** (canlı) anomali tespiti demonstrasyonu yapılmaktadır. Ek olarak, **SYN Flood**, **ICMP Flood**, **Port Tarama** gibi “kural tabanlı” saldırı yaklaşımları, K-Means ile birleştirilerek **hibrit** bir güvenlik çözümü sunulmuştur.

## 2. LİTERATÜR ÖZETİ

Literatürde, makine öğrenmesi yöntemlerinin siber saldırı tespitinde kullanıldığı pek çok çalışma bulunmaktadır:

* **Stolfo et al. (1999)**: KDD’99 veri setini ilk tanıtan ve DoS, Probe, U2R, R2L saldırılarını kapsamlı biçimde ele alan çalışma 111.
* **Tavallaee et al. (2009)**: NSL-KDD veri setinin tanıtılması ve KDD’99’un eksiklerini giderme çabası 222.
* **Ahmed ve Mahmood (2015)**: Denetimli ve denetimsiz öğrenme yöntemlerinin NSL-KDD üzerinde performans karşılaştırması. K-Means gibi kümeleme algoritmalarının, büyük boyutlu veri setlerinde hız avantajı olduğu vurgulanmıştır.
* **Garcia-Teodoro et al. (2020)**: Anomali tespit sistemlerinin kapsamlı bir sınıflandırması; istatistiksel, ML tabanlı ve bilgi kuramı temelli yöntemler detaylıca incelenmiştir.

Bu çalışmalardan alınan ilhamla, projede **K-Means** ve **custom saldırı fonksiyonlarının** (SYN Flood vb.) birlikte nasıl kullanılabileceği gösterilmektedir.

## 3. K-MEANS VE NSL-KDD VERİ SETİ

### 3.1. NSL-KDD Veri Setinin Özellikleri

NSL-KDD, KDD’99 veri setini bazı tekrar (redundancy) ve dengesizlik (imbalance) sorunlarını gidererek yeniden düzenlemiştir. Veri seti, her satırında yaklaşık 41 öznitelik barındırmaktadır:

* **Temel Öznitelikler**: Protokol (TCP, UDP, ICMP), Servis (HTTP, FTP, DNS vb.), Bayrak (S0, SF vb.).
* **İç Öznitelikler**: Bağlantı başına hata sayıları, tekrar denemeler, SYN sayısı vb.
* **Zaman Tabanlı Öznitelikler**: Belirli bir zaman dilimindeki bağlantı sayısı gibi.

### 3.2. Saldırı Türleri ve Normal Trafik

* **Normal**: Mevcut trafiğin saldırı içermemesi.
* **DoS (Denial of Service)**: Kaynak tüketimi amaçlı. SYN Flood, Ping of Death vb.
* **Probe (Tarama)**: Ağ haritalama, port/host taramaları.
* **U2R**: Kullanıcı haklarından root (yönetici) haklarına yükselme girişimleri.
* **R2L**: Uzaktan sisteme sızma saldırıları (ör. parola tahmini).

### 3.3. Örnek Veri Analizi (İstatistiksel Dağılımlar)

Bu aşamada veri setindeki **saldırı / normal** dağılımı, protokollerin yüzdesi, en sık görülen servis türleri gibi temel istatistikler tablo veya grafiklerle sunulur (Bkz. Tablo 1, Şekil 1).

| **Kategori** | **Kayıt Sayısı** | **Yüzde (%)** |
| --- | --- | --- |
| Normal | 67343 | 53.46 |
| DoS | 45927 | 36.44 |
| Probe | 11656 | 9.24 |
| R2L | 995 | 0.79 |
| U2R | 52 | 0.04 |

**Tablo 1.** NSL-KDD Veri Setindeki Saldırı ve Normal Dağılımı (Örneksel)

## 4. K-MEANS İLE NSL-KDD VERİ SETİNİN EĞİTİLMESİ

### 4.1. Veri Ön İşleme (Öznitelik Mühendisliği)

Projede kullanılan Python kodlarında:

1. **Gereksiz sütunlar**: difficulty\_level, attack\_type vb. (Araştırma aşamasında gerek duyulmuyor).
2. **Kategorik Kolonlar**: protocol\_type, service, flag → One-hot encoding (pandas get\_dummies) veya LabelEncoder.
3. **StandardScaler**: Tüm sayısal sütunları μ=0,σ=1\mu = 0, \sigma = 1μ=0,σ=1 biçimine getirir.

Bu aşamadan sonra, bir NumPy matris ya da X\_scaled DataFrame elde edilir.

A screen shot of a computer code

Description automatically generated

A screen shot of a graph

Description automatically generated

### 4.2. Model Eğitimi ve Parametre Seçimi

KMeans(n\_clusters=2, random\_state=42, n\_init=10) parametreleri kullanıldığında:

* **n\_clusters=2**: Normal / Anormal kümeleri.
* **n\_init=10**: Farklı başlangıç konumlarıyla 10 kez çalıştırma, en iyi sonucun seçilmesi.
* **random\_state=42**: Tekrarlanabilir sonuçlar.

Eğitim sonunda, her satır için **kmeans\_label** belirlenir. Modelin **küme merkezleri** (cluster\_centers\_) ve **inertia** değeri kaydedilir.

### 4.3. Model Performansı: Confusion Matrix, ROC Eğrisi, Diğer Metrikler

K-Means **denetimsiz** olsa da, NSL-KDD’nin gerçek etiketlerine bakarak bir “pseudo-denetimli” değerlendirme yapabiliriz. Örneğin:

1. **Confusion Matrix**: K-Means tarafından “ANORMAL” ve “NORMAL” atanan kayıtlar, gerçek etiketlere göre karşılaştırılır (Doğru Pozitif, Yanlış Pozitif, vb.).
2. **Hassasiyet (Recall), Seçicilik (Precision), F1-Skoru** gibi metrikler elde edilir.
3. **ROC Eğrisi**: Küme merkezine **uzaklık** bir “anomalilik skoru” şeklinde yorumlanabilir. Eşik değerini farklı seviyelere ayarlayarak (örn. %90, %95, %99 persentil) TPR ve FPR hesaplanır ve eğri çizilir.

## 5. SCAPY İLE AĞIN İZLENMESİ

### 5.1. Canlı Trafik Toplama Yöntemi

Gerçek zamanlı veri yakalamak için **Scapy**’nin sniff() fonksiyonundan yararlanılmaktadır. Kodda stoplama koşulları (**paket sayısı** veya **süre** bazlı) tanımlanabilir. Yakalanan her paket:

* **Kaynak IP, Hedef IP**
* **Protokol Numarası** (6 = TCP, 17 = UDP, 1 = ICMP)
* **SYN Flag** (TCP’de)
* **Zaman Damgası** ve **Paket Boyutu**
* **Inter-Arrival Time** (IAT) ve **Packet Rate** (kaynak IP’ye ait belirli pencere içi paket sayısı)

Parametreleriyle DataFrame’e eklenir. Bu veriler normalizasyon ve etiketleme sürecinden geçirilerek anomali tespitinde kullanılır.

A computer screen shot of text

Description automatically generated

### 5.2. SYN, ICMP ve Port Tarama Özellikleri

Custom saldırı fonksiyonlarının etkin biçimde çalışabilmesi için:

* **SYN**: Hedefin SYN flood’a maruz kalması demek, çok kısa sürede (ör. 2 saniye) aynı IP’den 5’ten fazla SYN gelmesidir.
* **ICMP**: 2 saniyede 100 baytı aşan ICMP paket akışı “ICMP Flood” sayılabilir.
* **Port Tarama**: 10 saniye içinde 5’ten fazla farklı porta erişim “tarama” olarak işaretlenebilir.

Bu sayede, K-Means “normal” dese bile belirlenen eşiği aşan saldırı tipi **ANORMAL** şeklinde raporlanır.

### 5.3. Ek Senaryolar (Wireshark PCAP Dosyaları vb.)

Bu bölümde, gerçekte kaydedilmiş pcap dosyaları (Wireshark vb.) üzerinden aynı kod çalıştırılabilir. scapy.rdpcap("dos\_attack.pcap") benzeri bir yöntemle offline analiz yapmak da mümkündür.

## 6. GERÇEK ZAMANLI ANOMALİ TESPİTİ

### 6.1. K-Means Tahmini

Önceden eğitilip saklanan **kmeans\_model\_realtime.pkl**, **scaler\_realtime.pkl** vb. dosyalar yüklenir. Gelen paketler tıpkı NSL-KDD’deki gibi:

1. **Protokol** → LabelEncoder.
2. **Port** → 0–1 arası normalizasyon.
3. **IAT / Packet Rate** → max’a bölünüp 0–1.
4. **StandardScaler** → Nihai ölçeklendirme.
5. **K-Means** → Mesafeye göre NORMAL / ANORMAL.

A screen shot of a computer code

Description automatically generated

### 6.2. Custom Saldırı Tespit Fonksiyonları

Bu aşamada, “kural tabanlı” saldırı tespiti fonksiyonları devrededir:

* **check\_syn\_flood(src\_ip)**: 2 saniye içinde 5 SYN → ANORMAL.
* **check\_icmp\_flood(src\_ip, pkt\_len)**: 2 saniye içinde 100 bayt → ANORMAL.
* **check\_port\_scan(src\_ip, dst\_port)**: 10 saniyede 5 farklı port → ANORMAL. A computer screen shot of text

  Description automatically generated

Tetiklendiklerinde, kod satırı logging.warning(...) ile log dosyasına kaydedilir.

### 6.3. Final Label, Loglama ve Grafikle Raporlama

En son, “kmeans\_label == ANORMAL veya custom\_label == ANORMAL” ise final\_label = ANORMAL. Terminalde özet bilgiler, log dosyasında ise **WARNING** seviyesinde saldırı mesajları görüntülenir. Ayrıca matplotlib ile çizilen bir grafikte:

* **mavi** noktalar: normal paketler,
* **kırmızı** noktalar: anormal paketler,
* **yeşil** yatay çizgi: K-Means eşik değeri.

A screenshot of a graph

Description automatically generated

Program kapatılıp tekrar çalıştırıldığında filemode='w' sayesinde log dosyası **sıfırlanır** ve yeni kayıt başlatılır.

A screen shot of a computer screen

Description automatically generated

## 7. DENEYSEL BULGULAR VE SENARYOLAR

### 7.1. Farklı SYN Flood Hızları

2 saniyede 3, 5 veya 10 SYN gibi farklı saldırı yoğunlukları test edilerek, tespit oranları analiz edilir. Bu sayede düşük yoğunluklu saldırılarda sistemin ne kadar başarılı (veya başarısız) olduğu gözlemlenir.

### 7.2. Farklı Eşik (Threshold) Değerleri

K-Means tarafında %90, %95, %99’luk persentil sınırlarının anomali tespit performansı karşılaştırılır. Tablo 2’de görüldüğü üzere, %90 eşiğinde daha çok **yanlış pozitif** çıkabilirken, %99 eşiğinde bazı saldırılar **kaçırılabilir** (FN artar). Bu trade-off, gerçek hayattaki güvenlik politikalarına göre ayarlanmalıdır.

### 7.3. Ayrıntılı Tablo ve Grafiklerle Karşılaştırma

Aşağıdaki tablo, normal / anormal tespitini Confusion Matrix üzerinden istatistikleştirebilir. Örneğin:

| **Eşik** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
| --- | --- | --- | --- | --- |
| %90 | 0.93 | 0.88 | 0.96 | 0.92 |
| %95 | 0.91 | 0.90 | 0.91 | 0.90 |
| %99 | 0.88 | 0.93 | 0.85 | 0.89 |

**Tablo 2.** Farklı Eşik Değerleriyle Elde Edilen Performans Metrikleri (Örneksel)

### 7.4. Gelişmiş Saldırı Senaryoları (Slowloris, Mixed Flood vb.)

* **Slowloris**: Düşük hızda ama uzun süreli paket akışı ile sunucu bağlantısını ele geçirmek. Burada IAT artabilir, custom fonksiyonlar adaptif eşik gerekebilir.
* **Mixed Flood**: SYN ve ICMP paketlerini karışık gönderen saldırılar. Kaynak IP sayısı fazlaysa “dağıtık” bir saldırı algılaması (DDoS) zorlaşır.
* **Arka Plan Trafiği**: Mevcut normal trafiğe (HTTP, DNS, SSH vb.) saldırı paketleri eklenerek gerçeğe daha yakın test ortamları yaratılabilir.

## 8. SONUÇ VE DEĞERLENDİRME

Bu rapor, **K-Means** ile anomali tespiti üzerine kapsamlı bir yaklaşım sunmuştur:

* **NSL-KDD veri seti** üzerinde off-line eğitilmiş bir model,
* **Scapy** ile gerçek zamanlı paket toplama ve custom saldırı fonksiyonları,
* Hem makine öğrenmesi hem de **kural tabanlı** alarm mekanizması bir arada.

Elde edilen bulgular, yüksek SYN paketi hızlarının kolayca yakalanabildiğini; ancak düşük hızlarda (ör. stealth tarama) veya karma paketlerde özel ayarların gerekebileceğini göstermektedir. K-Means’in denetimsiz oluşu, önceden tanımlanmamış saldırı türlerine karşı avantaj sağlasa da, **küme sayısı** veya **uzaklık eşiği** gibi parametre seçimleri hassas ayarlara ihtiyaç duyar.

Gelecekte yapılacak çalışmalarla, **daha gelişmiş makine öğrenmesi algoritmaları** ve **zaman serisi** analiz teknikleri eklenerek, saldırı tespitinin başarımının artırılması mümkündür. Ayrıca, büyük ölçekli dağıtık sistemlerde veya bulut ortamlarında da benzer çözümlerin uygulanması, pratik açıdan oldukça değerlidir.

## 9. KAYNAKÇA

1. **Stolfo, S.J. et al. (1999).** KDD’99 Cup. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
2. **Tavallaee, M. et al. (2009).** NSL-KDD: A New Data Set for Intrusion Detection Evaluations. https://www.unb.ca/cic/datasets/nsl.html
3. **Scapy Documentation** – <https://scapy.readthedocs.io/en/latest/>
4. **Python Scikit-Learn** – https://scikit-learn.org/stable/
5. **Garcia-Teodoro, P. et al. (2020).** Anomaly-based network intrusion detection: Techniques, systems and challenges.