

# LoPy4 Tabanlı Kablosuz Ağ Kesintisi Tahmin Sistemi

Makine Öğrenmesi ile Proaktif Ağ İzleme ve Erken Uyarı

Duha Keskin

Miraç Bayoğlu

Yunus Alim Avşar

Ahmet Güldaş

Eren Bezek

Doç. Dr. İzzet Fatih Şentürk

BM-5 Kablosuz Ağlar • 2025-2026 Güz Dönemi

# Problem Tanımı ve Motivasyon

Kablosuz ağlarda kesinti problemi ve proaktif çözüm yaklaşımı

## ! Mevcut Problemler

- Kablosuz ağlar fiziksel engeller ve parazitlerden etkilenir
- Bağlı kesintileri **öngörelemez** ve anı gerçekleşir
- Reaktif yaklaşımlar kesinti olduktan sonra müdahale eder
- Kritik uygulamalarda kesintiler yüksek maliyetlidir

## ✓ Çözüm: Proaktif Tahmin

- Sinyal kalitesi metrikleri sürekli izlenir
- Makine öğrenmesi ile kesinti **önceden tahmin** edilir
- Erken uyarı ile önlem alma fırsatı sağlanır
- Sistem güvenilirliği ve kullanıcı deneyimi artar

## ★ Proje Hedefleri

1

### Veri Toplama

RSSI ve RTT metriklerini gerçek zamanlı ölçme

2

### Kesinti Tahmini

ML modeli ile bağlı kesintilerini önceden tespit

3

### Görselleştirme

Web dashboard ile anlık izleme ve analiz

4

### Erken Uyarı

Farklı seviyelerde alarm ve bildirim sistemi

# Sistem Mimarisi

Üç bileşenli IoT mimarisi ve veri akış yapısı



# Donanım: Pycom LoPy4

Çoklu bağıntılı destekli IoT geliştirme kartı

## Teknik Özellikler

Özellik	Değer
İşlemci	ESP32 Dual-Core 240MHz
RAM	520 KB SRAM
Flesh	4 MB
Wi-Fi	802.11 b/g/n
Bluetooth	BLE 4.2
LoRa	868/915 MHz

## Neden LoPy4?

- Çoklu bağlantı (Wi-Fi, BLE, LoRa)
- MicroPython ile hızlı geliştirme
- Düşük güç tüketimi
- Gerçek zamanlı RSSI erişimi

## Cihaz Rollerleri

### AP Access Point

- Wi-Fi eşi oluşturur **AP Mode**
- İki TCP sunucusu (Port 12345, 12346)
- Client verilerini PC'ye aktarır
- LED ile durum gösterir

### CL Client (İstemci)

- AP ağına bağlanır **STA Mode**
- RSSI değerini ölçer (Sinyal gücü)
- RTT hesapları (Geçikme)
- Her saniye veri gönderir

# İstemci (Client) Cihazı

RSSI ve RTT ölçüm döngüsü

## Ölçüm Döngüsü

1 Ağ Tarama  
Wi-Fi ağları taranır, AP bulunur

2 RSSI Okuma  
Sinyal gücü (dBm) değeri alınır

3 TCP Bağlantısı (T1)  
AP'ye bağlantı açılır, zaman kaydedilir

4 Veri Gönder & Kapat (T2)  
RSSI gönderilir, ACK alınır

5 RTT Hesaplama  
 $RTT = T2 - T1 \text{ (ms)}$

## RSSI Kalite Tablosu

RSSI ARALIĞI	KALİTE
> -50 dBm	Mükemmel
-50 ~ -60 dBm	İyi
-60 ~ -70 dBm	Orta
-70 ~ -80 dBm	Zayıf
< -80 dBm	Çok Zayıf

## RTT (Round-Trip Time)

- TCP handshake süresi ölçülür
- $\text{Latency} \approx \text{RTT} / 2$

## LED Göstergeleri

- 🔴 Kırmızı: Ağ aranıyor
- 🔵 Mavi: Veri gönderildi
- 🟡 Sarı: RSSI okunemedi

# Erişim Noktası (AP) Cihazı

Çift sunucu mimarisi ve veri aktarımı

## Ağ Yapılandırması

```
# Wi-Fi AP Ayarları  
SSID = "Lopy4-Network"  
Password = "Lopy4pass123"  
IP = "192.168.4.1"  
Channel = 6
```

## Veri Akış Diyagramı



## Çift Sunucu Mimarisi

### Port 12345

Client Sunucusu

- Client verisi alır
- RSSI:değer formatı
- ACK yanıtı döner

### Port 12346

PC Sunucusu

- PC'ye veri ekterir
- STATUS mesajları
- DATA:rsđ,rtt,count

## LED Göstergeleri

- Yeşil: Client bağlı, veri akışı aktif
- Turuncu: Bağlantı yok / veri geçidi

## Timeout Mekanizması

Client verisi **5 saniye** gelmezse:

- STATUS:DISCONNECTED mesajı gönderilir
- LED turuncu yanır

# Makine Öğrenmesi Modeli

Hibrit tahmin sistemi: Kural tabanlı + Random Forest

## Hibrit Tahmin Sistemi



## 13 Özellik (Feature)

### RSSI Özellikleri

- rssi\_mean, rssi\_std
- rssi\_min, rssi\_max
- rssi\_trend, rssi\_delta

### RTT & Kalite

- rtt\_mean, rtt\_std
- rtt\_min, rtt\_max, rtt\_trend
- quality\_mean, quality\_std

## Random Forest Parametreleri

```
n_estimators = 100 # Ağac sayısı  
max_depth = 18  
min_samples_split = 5  
class_weight = "balanced"
```

## Kural Tabanlı Eşikler

METRİK	DİKKAT	UYARI	KRİTİK
RSSI	-60 dBm	-75 dBm	-85 dBm
RTT	100 ms	200 ms	—
Trend	-3 dB/s	-5 dB/s	—

## Uyarı Seviyeleri

- [!] INFO — Bilgilendirme
- [!] DİKKAT — Dikkat edilmeli
- [!] UYARI — Müdahale gerekebilir
- [!!] KRİTİK — Acil müdahale

# Web Dashboard

Gerçek zamanlı izleme ve görselleştirme arayüzü

## Teknoloji Stack



Flask  
Python web framework



SocketIO  
Gerçek zamanlı iletişim



Chart.js  
Grafik görselleştirme



WebSocket  
Anlık veri ﬂoşu

## REST API Endpoints

```
GET / # Dashboard sayfası
GET /api/stats # Tüm istatistikler
GET /api/status # Bağlantı durumu
GET /api/history # Geçmiş veriler
GET /api/warnings # Uyarı listesi
```

## Dashboard Önizleme

localhost:5001

### Dashboard - Kablosuz Ağ İzleme

RSI Graflığı



RTT Graflığı



RSI

-45

RTT

23ms

Kalite

Mükemmel

Ölçüm

1247

Uyarı: 10 sinyal zayıf (3dB'dan aşağı)

12:34:56

# Veri Akışı ve Protokol

Uçtan uca iletişim protokolü ve mesaj formatları

## Uçtan Uca Veri Akışı

1 Client: RSSI Ölçümü  
Sinyal gücü okunur

2 Client → AP: Veri Gönderimi  
TCP ile gönderilir

3 AP: RTT Hesaplama  
Hendekme süresi

4 AP → PC: Veri Aktarımı  
DATA[ssr,rtt,count]

5 PC: ML Tahmin  
Özellik çıktıları

6 Dashboard: Görselleştirme  
WebSocket güncelleme

## Mesaj Formatları

Client → AP (Port 12345)

RSSI: -45

AP → PC (Port 12346)

STATUS: CONNECTED  
DATA: -45, 23, 1247

## CSV Veri Şeması

session\_id, timestamp, measurement\_id,  
event\_type, rssi, rtt, latency, quality

## Zamanlama

### BİLEŞEN

Client ölçümü

### PERİYOT

1 saniye

AP relay

Anlık

ML İşleme

< 10 ms

Timeout

5 saniye

# Sonuç ve Demo

Projenin katkıları ve canlı demonstrasyon

## ✓ Projenin Katkıları

1

### Proaktif Kesinti Tahmini

Bağlı kesintiden önce erken uyarı

2

### Hibrit ML Yaklaşımı

Kural tabanlı + ML kombinasyonu

3

### Gerçek Zamanlı İzleme

Web dashboard ile görselleştirme

4

### Tam Stack Çözüm

Donanım + Yazılım + ML

## ⌚ Kullanım Alanları

- IoT eğ izleme sistemleri
- Endüstriyel kablosuz uygulamalar
- Akıllı bina yönetimi



## Gelecek Gelişmeler

- LoRa protokolü ve çoklu client desteği
- Derin öğrenme modelleri (LSTM)
- Mobil uygulama ve bulut entegrasyonu



## Demo Senaryosu

- AP ve Client cihazlarını başlat
- PC uygulaması + Web dashboard aç
- Cep telefonunuza veri akışını izle
- Kesinti simülasyonu ve uyarı gözlemlé

**Teşekkürler!**

Sorularınız için hazırız