**YED TASK**

**Berkay Kenan Erciyes**

**1) Amaç**

* Hücre deşarj döngülerinden **anlık akım (I[t])** ve **döngü kapasitesi (capacity\_Ah)** öngörüsü.
* Öngörülen kapasite ile:
  + **soc\_cycle** (cycle bazında SoC, ilk kapasiteye göre normalizasyon)
  + **soc\_whole\_t(t)** (global ilk kapasiteye göre zaman ekseninde SoC)
  + **soc\_cyclecap\_t(t)** (tahmin edilen cycle kapasitesine göre zaman ekseninde SoC)  
    değerlerinin hesaplanması.
* Modeli **MQTT** üzerinden başka bir bataryadan gelen verilerle **çevrim dışı/çevrim içi test edebilmek** (cross-battery generalization).

**2) Veri ve EDA (özet)**

Veri setinde 3 benzer batarya (**B0005, B0006, B0018**) ve ek bir batarya (**B0007**) bulunur. EDA’dan çıkan hızlı bulgular:

* Deşarj sırasında **voltaj plato süreleri** ve **akım profili** kapasiteyle ilişkili (kapasite düşüşünde plato kısalıyor).
* **Sıcaklık artışı**, kapasite düşüşü ile korelasyon gösteriyor (yüksek C-rate ve sınırlı soğutma koşullarında daha belirgin).
* **Zaman (time\_s) normalizasyonu** sinyal senkronizasyonunu iyileştiriyor.
* Opsiyonel yük kanalları (**voltage\_load\_V / current\_load\_A**) mevcutsa sinyal ayrışması artıyor; yoksa sıfırlarla doldurma yapılabiliyor (robust input).

(İsteğe bağlı dosyalar: predictions\_raw.csv, predictions\_upsampled.csv, corr\_mat.xlsx, vb. EDA görselleri/tabloları buralarda.)

**3) Model Mimarisi**

**CNN-BiLSTM çok-başlı yapı (multi-head)**

* **CNN kademesi:** 1D Conv( in\_dim→32 →64 ), ReLU, (MaxPool opsiyonel)
* **BiLSTM:** hidden=128 (varsayılan), layers=2, bidirectional
* **Head-I:** (2H→H→1) → her timestep için **I\_hat[t]**
* **Head-cap:** (2H→H→1) → sequence’den **tek skalar capacity\_hat**

Giriş kanalları:

* Zorunlu: **voltage\_V, temperature\_C, time\_s\_norm**
* Opsiyonel: **capacity\_Ah** (sabit sütun) → --load\_cap
* Opsiyonel: **voltage\_load\_V, current\_load\_A** → --use\_load

**4) SOC Hesabı**

compute\_soc(I\_pred, cap\_pred, time\_s, first\_cap) ile:

* q\_drawn\_ah = cumsum(sign\*I\*dt) / 3600
* soc\_cycle = cap\_pred / first\_cap
* soc\_whole\_t = clip( soc\_cycle - q\_drawn\_ah/first\_cap, 0, 1.05 )
* soc\_cyclecap\_t = clip( 1 - q\_drawn\_ah/cap\_pred, 0, 1.05 )

**sign**: median(I\_pred) < 0 ise −1 (deşarj negatif akım konvansiyonuna uyum).

**5) Eğitim/Değerlendirme**

* Kayıp: **MSE(I[t]) + α·MSE(capacity)** (α=10 varsayılan).
* En iyi skor: rmse\_I + α·rmse\_cap min.
* **GridFS metadata**’ya kayıt: mimari (in\_dim, hidden, layers, dropout), **train\_sets**, **test\_set**, **metrics**.

**Örnek metrik (dosyadan):**

/mnt/data/metrics.json → {"rmse": 0.01876}  
Bu değer demo amaçlıdır; proje kodu, **cap ve I için ayrı RMSE** üretip GridFS meta olarak yazabilir (aşağıdaki endpoint ile döndürüyoruz).

**6) MQTT Entegrasyonu**

* **Subscribe:** yed/battery/+/telemetry
* **Publish (tahmin):** yed/battery/{device}/predictions
* Payload 2 şekilde desteklenebilir:
  1. **Referansla** (en pratik): { "dataset": "B0007", "cycle\_id": 42, "mode":"currcap\_w\_cap", "use\_load": false }  
     (Backend Mongo’dan çeker → tahmin → publish)
  2. **Ham dizi** (opsiyonel): voltage\_V[], temperature\_C[], time\_s[], opsiyonel voltage\_load\_V[], current\_load\_A[] (format backend’deki on\_message’ın beklentisine göre)

**7) Cross-Battery Test (Task Koşulu)**

* **Eğitim:** B0005 + B0006 + B0018
* **Test:** **B0007** (frontend seçiminde sadece test set görünsün; backend de güvenlik için reddetsin)
* Bu sayede **başka batarya yayını** geldiğinde modelin genelleme performansı gözlemlenir.

**8) Kısıtlar ve İyileştirme**

* Veri kalitesi: sensör gürültüsü/eksik kanal → robust downsample ve sıfırlama stratejisi uygulandı.
* Daha iyi genel performans için:
  + Zaman serisi uzunluğu uyarlamalı (seq\_len tuning)
  + Head’ler için farklı ağırlıklandırma (α optimize)
  + Öğrenme hızı, weight\_decay, early-stopping
  + Model versiyonlama (birden fazla model saklama, seçim endpoint’i)