

## Case Study Raporu

### ADIM 0: LİTERATÜR TARAMASI VE MEVCUT PROJELERİN İNCELENMESİ

- SOH (State of Health) ve SOC (State of Charge) için verinin nasıl işlendiği ve hangi modellerin kullanıldığı analiz edildi.
- Mevcut kodlar incelendi:
  - Manus'un interaktif websitesi: <https://jzbzrilu.manus.space/> (bu siteden çokça faydalandı)
  - Manus Replay Linki: <https://manus.im/share/bbELiqCMAgfS4AXhtWQc1Y?replay=1>
  - Literatür Taraması ve Roadmap Oluşturma (Manus): <https://manus.im/share/Yd5U9FmYfGNd0woq8llyIA?replay=1>

Manus, kısıtlı zamanımız olduğu için research anlamında büyük ölçüde yardımcı oldu.

### ADIM 1: VERİSETİNİ İNDİRME

Veriseti manuel olarak indirildi. B0005.mat, B0006.mat, B0018.mat dosyaları kullanıldı.

### ADIM 2: VERİSETİNİ İNCELEME VE .MAT FORMATINDAN CSV FORMATINA GEÇİŞ

- İlgili Notebook: notebooks/1\_initial\_data\_inspection.ipynb
- Referans Repo: <https://github.com/MoHoss007/Li-Ion-Battery-RUL-SOH-Prediction>
  - Dataseti .mat formatından .csv formatına geçirmedi bu repodan yardım alındı.

### ADIM 3: DATA PREPARATION (VERİ HAZIRLAMA)

- İlgili Notebook: notebooks/2\_data\_preparation.ipynb

#### 3.1 Eksik Değerler Tespiti

Eksik değerler tespit edildi.

#### 3.2 Veri Tipleri Kontrolü

Veri tipleri kontrol edildi.

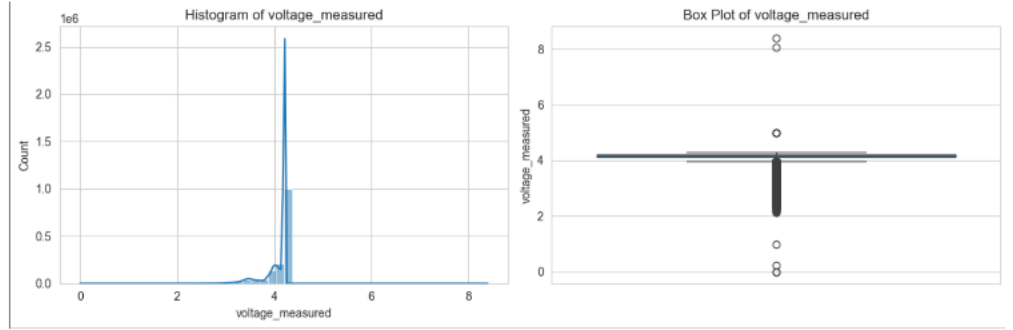
#### 3.3 Tekrarlanan Değerler Kontrolü

Tekrarlanan (duplicate) değerler kontrol edildi.

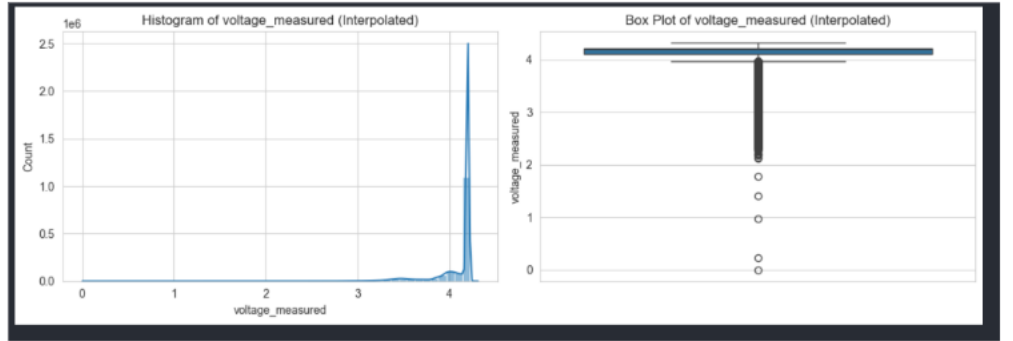
#### 3.4 Outlier (Aykırı Değer) Analizi

- Histogramlar ve Boxplotlar incelendi.
- Voltaj değerlerindeki outlier'lar temizlendi: 0.0V - 4.5V aralığının dışındaki 7 değer tespit edilip veri setinden çıkarıldı (drop edildi).
- Daha sonra eksik veriler (çıkarılan outlier'ların yerine) Linear Interpolation tekniğiyle dolduruldu.

before



after



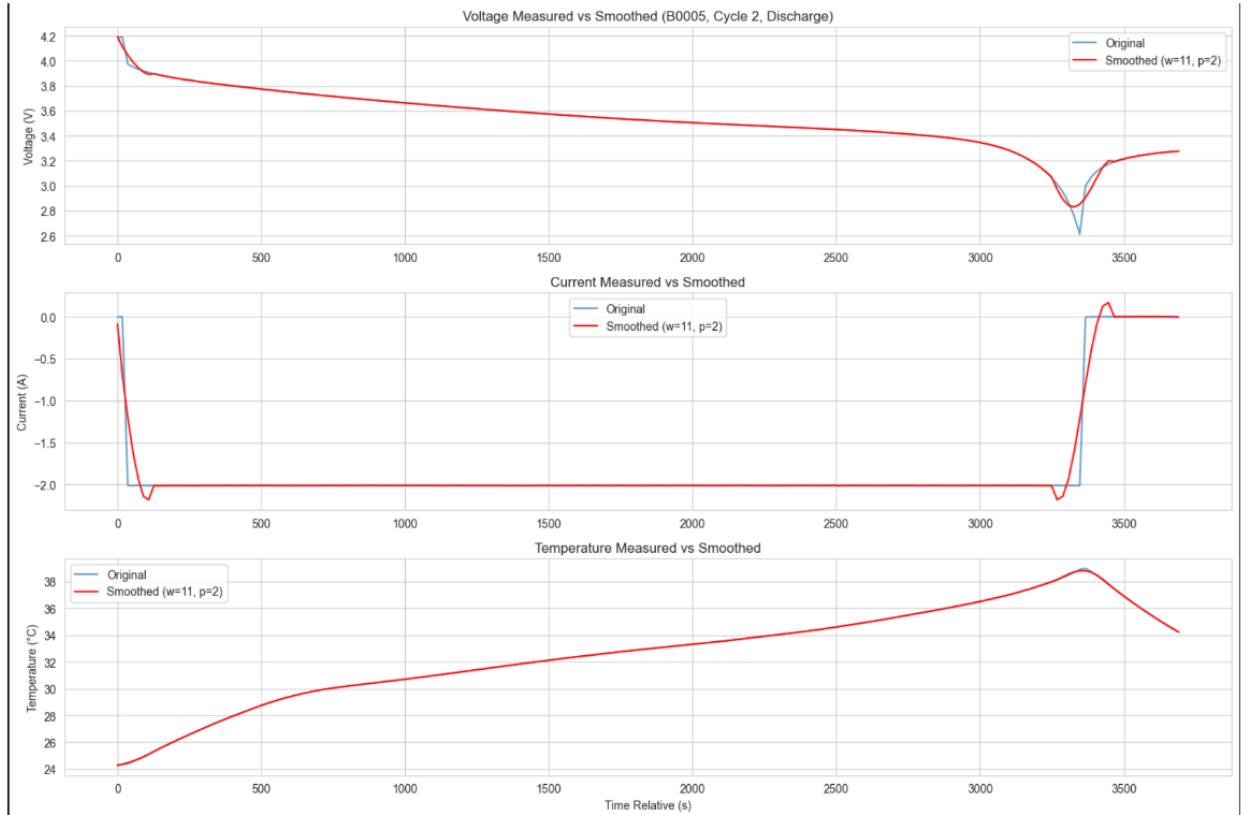
### 3.5 SOH (State of Health) Değerlerinin Hesaplanması

- SOH değerleri hesaplandı.
- Batarya konusuna biraz yabancı olduğu için SOH ve SOC arasındaki farkı anlamak zaman aldı. Bu noktada Gemini'dan bolca yardım alındı.
- Ayrıca, verisetini model için hazırlarken SOH değerlerinin nasıl kullanılacağı konusunda başlangıçta kafa karışıklığı yaşandı. Daha sonra detaylı literatür taraması ve mevcut projelerin incelenmesiyle mevcut yapıya karar verildi.

$$SOH = \frac{C_{measured}}{C_{nominal}} \times 100$$

### 3.6 Smoothing için Savitzky-Golay (Sav-Gol) Filtresi

- Grafiklerde voltajda bazı ani sıçramalar (spike'lar) gözlemlendi.
- Bu ani değişimlerin modelin performansını olumsuz etkilememesi için Savitzky-Golay filtresi kullanıldı.
- Bu yaklaşım, "*Data-driven SOH Estimation of Lithium-ion Batteries Based on Savitzky-Golay Filtering and SSA-SVR Model*" makalesinde kullanıldığı ve model performansında artış gözlemlendiği belirtildiği için tercih edildi.



### 3.7 Verinin CSV Olarak Kaydedilmesi

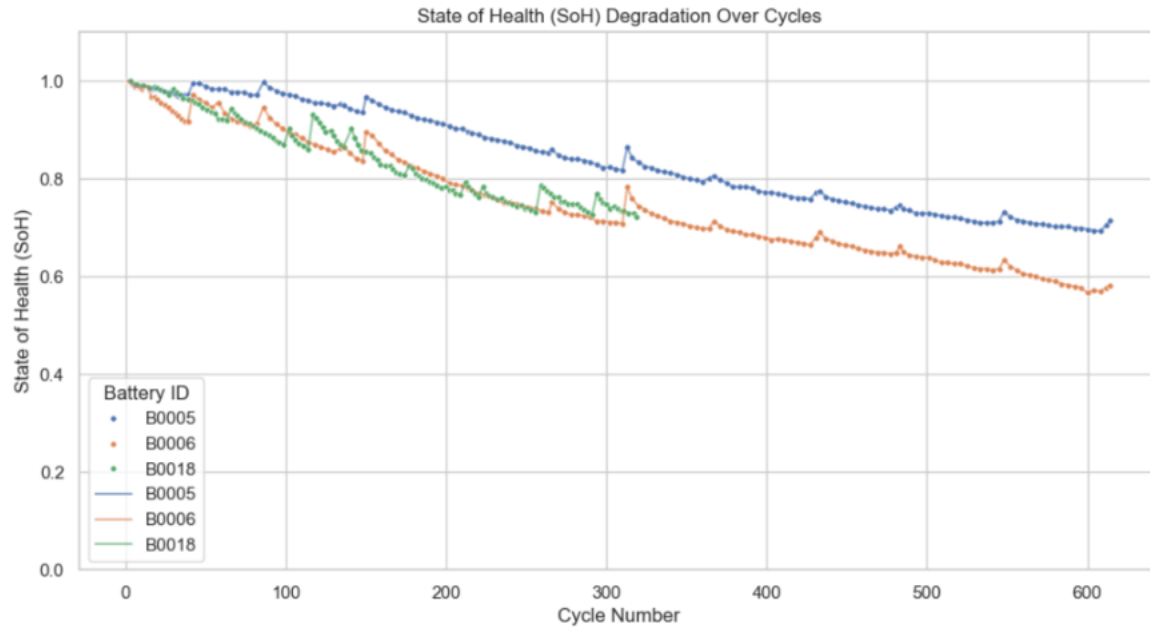
Son olarak işlenen veri tekrar .csv formatında kaydedildi.

## 4. EDA (EXPLORATORY DATA ANALYSIS - KEŞİFÇİ VERİ ANALİZİ)

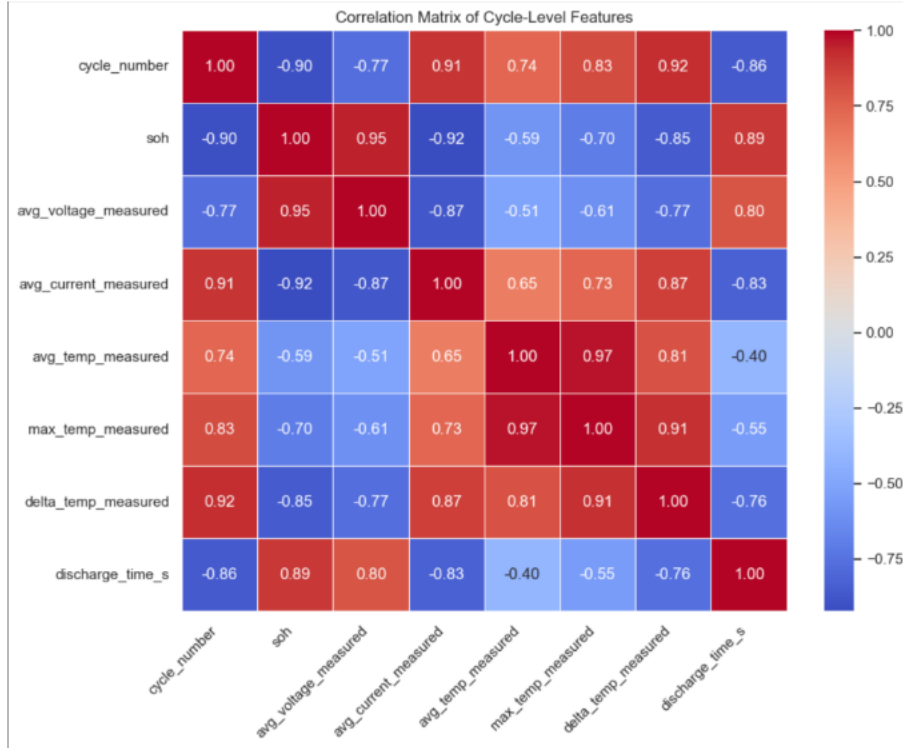
- İlgili Notebook: notebooks/3\_exploratory\_data\_analysis.ipynb

### 4.1 Bataryadaki SoH Değerinin Cycle'lara Göre İncelenmesi

Bataryanın sağlık durumunun (SoH) şarj/deşarj döngü sayısı (cycle) arttıkça nasıl bozulduğu incelendi.



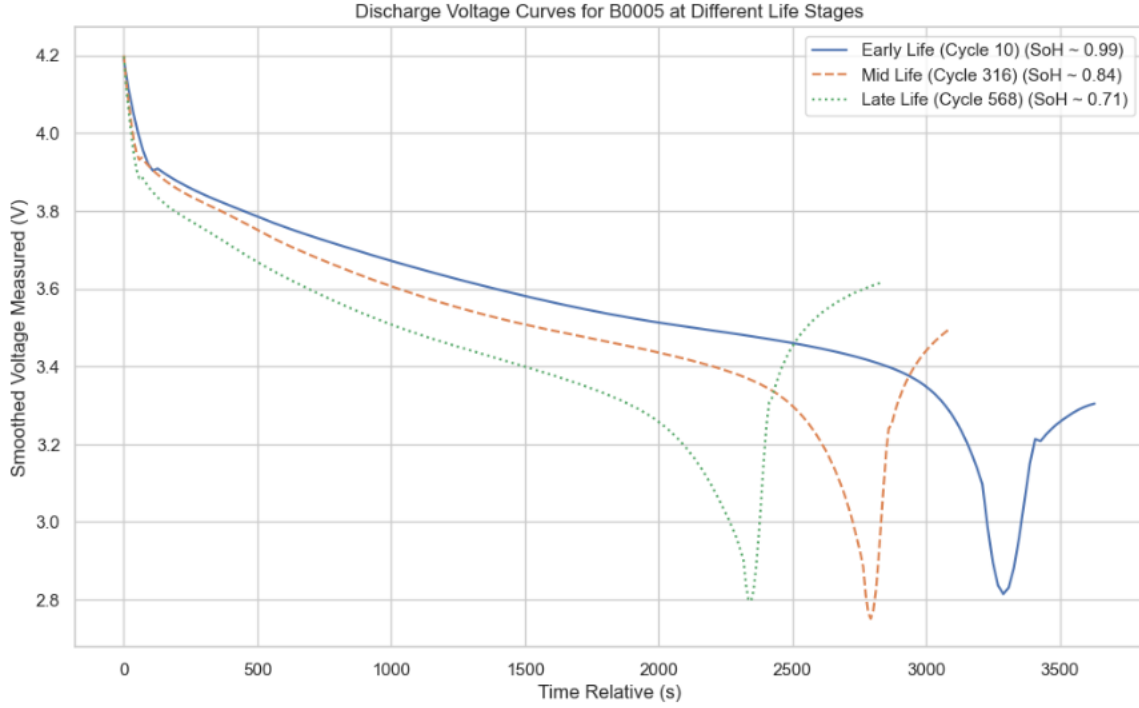
## 4.2 Korelasyon Analizi



- **soh ile cycle\_number (-0.90):** Beklendiği gibi çok güçlü negatif korelasyon. Döngü sayısı arttıkça bataryanın sağlığı azalmaktadır.
- **soh ile avg\_voltage\_measured (+0.95):** Son derece güçlü pozitif korelasyon. Bu, batarya bozuldukça (SoH düştükçe), deşarj sırasındaki ortalama voltajın önemli ölçüde azaldığını göstermektedir. Bu durum, avg\_voltage\_measured'ın SoH tahmini için çok umut verici bir öznelilik (feature) olduğunu işaret ediyor.
- **soh ile discharge\_time\_s (+0.89):** Çok güçlü pozitif korelasyon. Sağlıklı bataryaların aynı akım altında tamamen deşarj olması daha uzun sürer. SoH düştükçe, bataryanın belirlenen kesme voltajına ulaşma süresi kısalmır. Bu da modelleme için mükemmel bir potansiyel özneliliktir.
- **soh ile avg\_current\_measured (-0.92):** Çok güçlü negatif korelasyon.
- **soh ile delta\_temp\_measured (-0.85):** Güçlü negatif korelasyon. Sağlıklı bataryalar, bozulmuş bataryalara kıyasla deşarj sırasında daha küçük bir sıcaklık artışı yaşarlar. Bozulmuş bataryaların iç direnci genellikle daha yüksek olduğundan, deşarj sırasında daha fazla ısı üretirler. Bu nedenle delta\_temp\_measured (deşarj sırasındaki sıcaklık değişimi) SoH tahmini için çok kullanışlı bir özneliliktir.
- **soh ile max\_temp\_measured (-0.70):** Orta derecede güçlü negatif korelasyon. Bozulmuş bataryalar deşarj sırasında daha yüksek maksimum sıcaklıklara ulaşma eğilimindedir.
- **soh ile avg\_temp\_measured (-0.59):** Orta derecede negatif korelasyon. Benzer şekilde, bozulmuş bataryalar deşarj sırasında genellikle daha yüksek bir ortalama sıcaklığa sahip olurlar.

#### 4.3 Deşarj Voltajının Farklı SoH Seviyelerine Göre Grafikleştirilmesi

Deşarj sırasındaki voltaj eğrilerinin farklı SoH seviyeleri için nasıl değiştiği görselleştirildi.



#### 4.4 EDA Bulgularına Göre Feature Engineering ve Verinin Kaydedilmesi

EDA bulguları ışığında yeni öz nitelikler türetildi (feature engineering) ve son veri seti bataryalara göre ayrı ayrı kaydedildi.

### 5. Model Geliştirme

- İlgili Notebook: `Notebooks/4_model_development.ipynb`

#### 5.1 Verinin Train/Validation/Test Olarak Ayrılması

Bu bir zaman serisi (time-series) tahmin görevi olduğu için veri kronolojik olarak train ve test setlerine ayrıldı. Veri %70 train, %15 validation ve %15 test olacak şekilde bölündü.

#### 5.2 Featureların Hazırlanması

Featurelar SoH tahmini için hazırlandı.

### 5.3 Random Forest Modeli Testi 1

```
Training Random Forest Regressor for B0005...  
Training complete in 1.81 seconds.
```

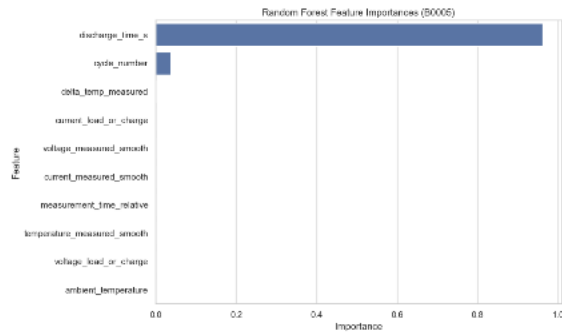
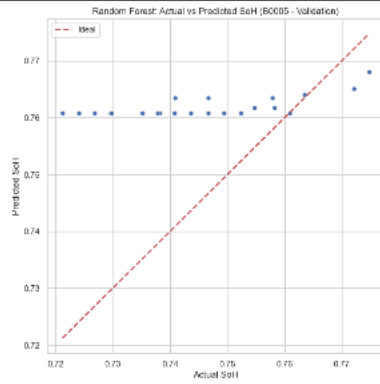
```
Evaluating on Validation Set...
```

```
Validation MAE: 0.0193
```

```
Validation RMSE: 0.0227
```

```
Validation R2: -1.3613
```

```
OOB Score (R2): 1.0000
```



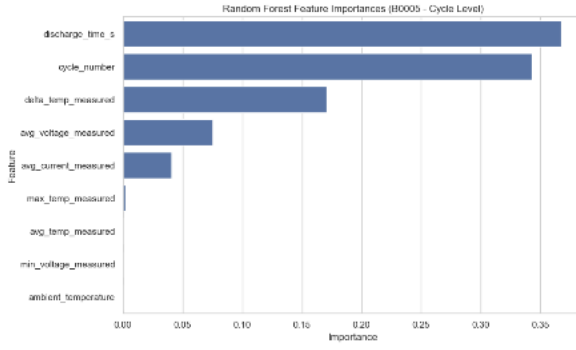
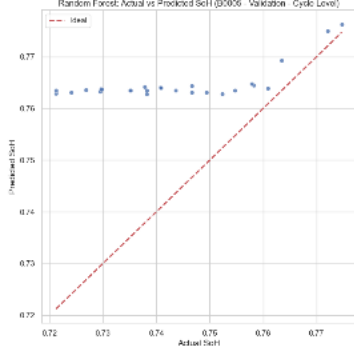
- Model, training ve validation setlerinde güzel sonuçlar verirken test setinde çok kötü sonuçlar verdi. Bu durum **Overfitting** göstergesidir.
- Muhtemel Sebepler:
  1. Veri anlamsızlığı ya da fazlalığı.
  2. Data leakage (veri sızması).
  3. Hyperparametrelerin optimal olmaması.

## 5.4 Random Forest Modeli Testi 2

```
Training Random Forest Regressor for B0005 (CYCLE LEVEL)...  
Training complete in 0.08 seconds.
```

```
Evaluating on Validation Set (CYCLE LEVEL)...
```

```
Validation MAE: 0.0213  
Validation RMSE: 0.0248  
Validation R2: -1.8315  
OOB Score (R2): 0.9947
```



- Veri seti deşarj döngülerine göre tekrar ayarlandı.
- Hyperparametreler düzenlendi.
- Yine overfitting sorunuyla karşılaşıldı.

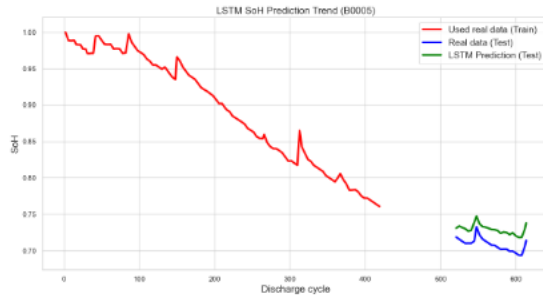
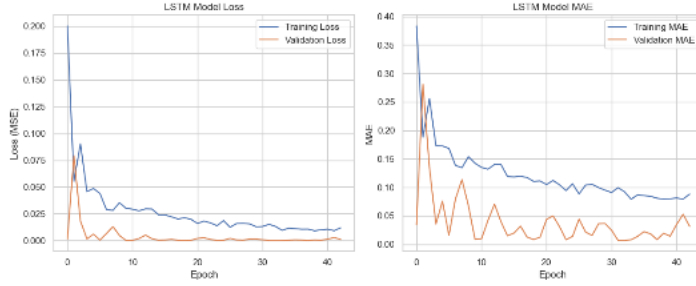
## 5.5 XGBoost Regressor Testi

Bu model denemelerinde de overfitting ile karşılaşıldı. Model çıktıları notebook'da görüntülenebilir. Yine aynı sonuçlar olduğu için çıktıyı buraya koymadım.



## 5.6 LSTM Testleri

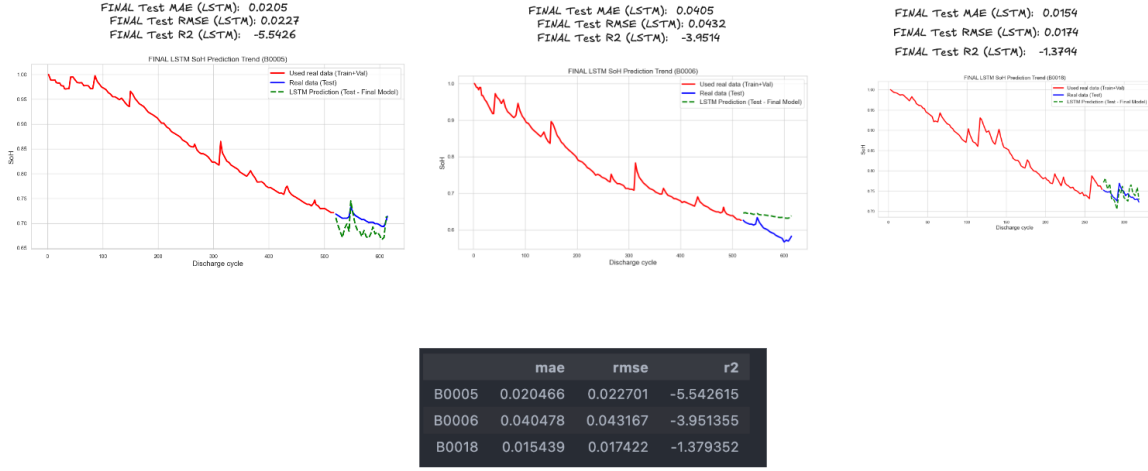
Layer (type)	Output Shape	Param #
masking (Masking)	(None, 373, 4)	0
lstm (LSTM)	(None, 30)	11,000
dropout (Dropout)	(None, 30)	0
dense (Dense)	(None, 1)	31



```
Evaluating LSTM on Test Set...
1/1 ----- 0s 85ms/step
Test MAE (LSTM): 0.0208
Test RMSE (LSTM): 0.0211
Test R2 (LSTM): -4.6323
```

- Veri, LSTM modelinin girdisine uygun olması için sekanslar (sequences) halinde yeniden düzenlendi.
- LSTM, overfitting sorununu çözdü ve gayet iyi sonuçlar verdi.
- **Not:** Grafik incelendiğinde validation seti için ayrılan boşluk fark edildi. Modelin RF ve XGBoost'da test setinde düşük sonuçlar vermesinin sebebinin bu olabileceği düşünüldü, ancak tekrar test edildiğinde bununla alakalı olmadığı anlaşıldı.

## 5.7 LSTM'in Diğer Bataryalar İçin Test Edilmesi



LSTM modeli diğer batarya verileri (B0006, B0018) için de test edildi ve diğer modellere oranla daha başarılı sonuçlar alındı.

- **SONUÇLAR** - Her batarya için ayrı ayrı oluşturulan ve eğitilen modeller kaydedildi. Bu modeller daha sonra REST API'da kullanılmak üzere saklandı.

## 5.8 EXTRA WORK: CNN-BiLSTM-AM Modeli

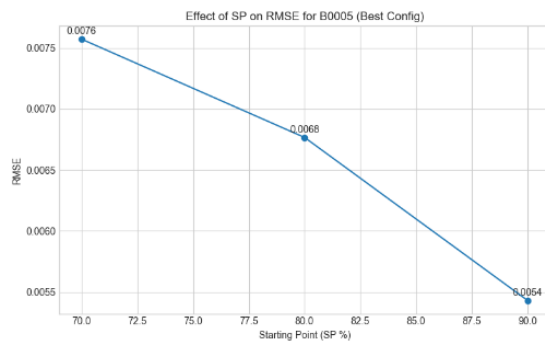
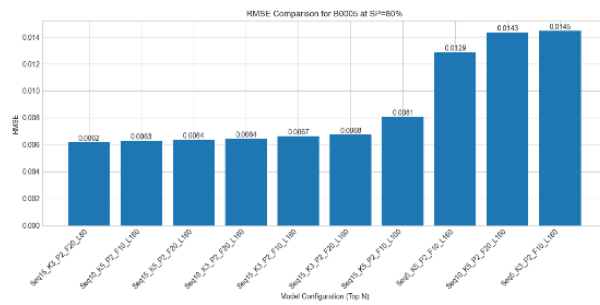
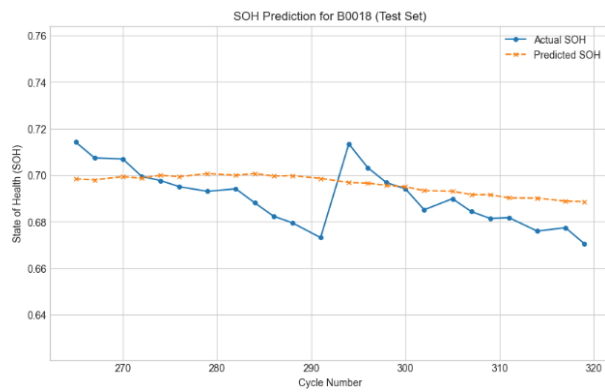
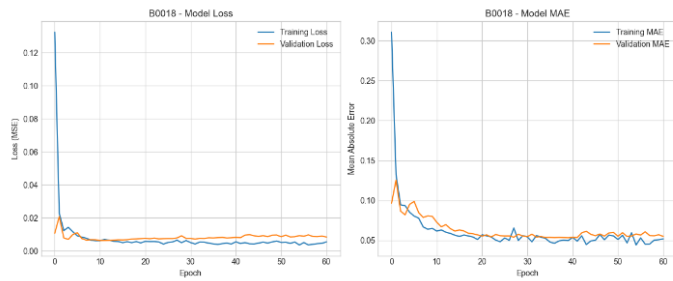
- İlgili Notebook: 4.1\_CNN\_BiLSTM\_AM.ipynb
- Referans Makale: "State-of-Health Prediction of Lithium-Ion Batteries Based on CNN-BiLSTM-AM" (Tian et al., Batteries 2022)
- Bu makaledeki model (CNN-BiLSTM-Attention Mechanism) ve sonuçları incelendiğinde başarılı bulundu.
- Makalede kullanılan Sağlık Göstergesi (Health Indicator - HI): **TIEDVD (Time Interval of Equal Discharging Voltage Difference)** - Deşarj sırasında 3.8V ile 3.4V arasındaki Eşit Deşarj Gerilimi Farkı Zaman Aralığı. Kabaca bu değer öznitelik olarak kullanıldı.
- Model, makalede belirtildiği gibi kodlandı (build edildi).
- Ekstra hyperparameter tuning (hiperparametre optimizasyonu) yapıldı ve farklı başlangıç noktaları (SP - starting point) denendi.
- En iyi hiperparametreler ve konfigürasyonlar bulundu. Sonuçlar notebook'un sonunda tablo olarak listelendi.
- Bu model, geliştirilen modeller arasında en iyi sonucu veren model oldu.
- Ancak, bu modelin API'ye adaptasyonu sırasında bazı sıkıntılarla karşılaşıldı ve zaman kısıtlaması nedeniyle bu kısma daha fazla odaklanılmadı. (Bu kısımda biraz daha vaktim olsaydı çözebileceğime inanıyorum).
- Dökümantasyon yazma işi son kısma bırakıldı. (Şu an tarih 11.04.2024 ve saat 03:38, sanırım doğru kararı vermişim.)
- Makaledeki modelin koda dönüştürülmesi noktasında **Gemini 2.5 Pro** inanılmaz derecede yardımcı oldu.

Table 1: Performance Metrics for Different Batteries and Starting Points

Battery	Prediction SP	Model	RMSE	MAE	MAPE (%)
B0005	70 (60%)	cnn-bilstm-am	0.0226	0.02058	3.10
	80 (70%)	cnn-bilstm-am	0.0150	0.01332	2.02
	90 (80%)	cnn-bilstm-am	0.0116	0.01018	1.56
B0006	70 (60%)	cnn-bilstm-am	0.0367	0.03086	5.03
	80 (70%)	cnn-bilstm-am	0.0359	0.03172	5.25
	90 (80%)	cnn-bilstm-am	0.0353	0.03333	5.62
B0018	70 (60%)	cnn-bilstm-am	0.0159	0.01349	1.94
	80 (70%)	cnn-bilstm-am	0.0132	0.01100	1.59
	90 (80%)	cnn-bilstm-am	0.0132	0.01141	1.66

Layer (type)	Output Shape	Param #
input_layer_16 (InputLayer)	(None, 15, 1)	0
conv1d_16 (Conv1D)	(None, 15, 20)	80
max_pooling1d_16 (MaxPooling1D)	(None, 8, 20)	0
dropout_32 (Dropout)	(None, 8, 20)	0
bidirectional_16 (Bidirectional)	(None, 8, 320)	231,800
dropout_33 (Dropout)	(None, 8, 320)	0
simple_attention_16 (SimpleAttention)	(None, 320)	183,840
dense_16 (Dense)	(None, 1)	321





## 6. API & DEMO & DOCKERIZATION

### 6.1 API (Backend)

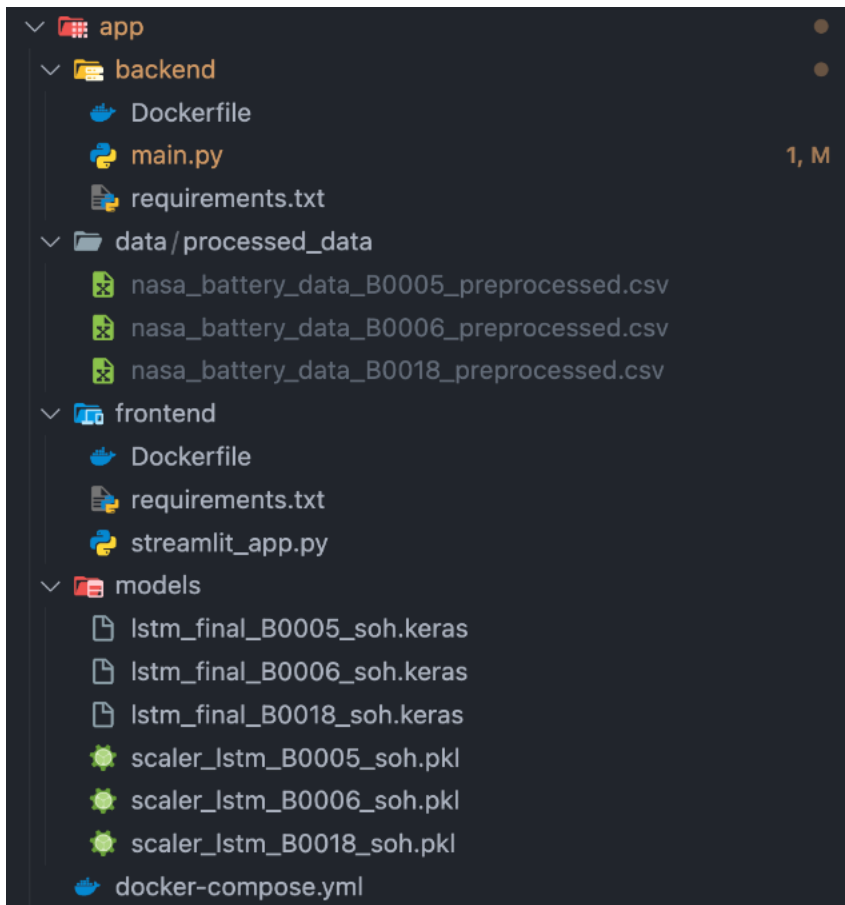
- **Framework:** FastAPI (container içinde çalışıyor)
- **Port:** Host üzerinde 5001 portunda yayınlanır (container içi port 5000'e eşlenmiştir).
- **Volumes:**
  - /models: Batarya tahminleri için ML modellerini içerir.
  - /data: Batarya veri setlerini içerir.
- **Endpoint:** Batarya SOH (State of Health) tahmini için bir endpoint sağlar.

### 6.2 DEMO APP (Frontend)

- **Framework:** Streamlit kullanılarak oluşturulmuştur.
- **Port:** 8501 portu üzerinde çalışır.
- **Özellikler:**
  - Batarya tahminleri için interaktif web arayüzü.
  - Veri görselleştirme (data visualization) bileşenleri.

### 6.3 DOCKERIZATION

- **Araç:** Docker Compose kullanılarak multi-container kurulumu yapıldı.
- **Bileşenler (Containers):**
  - backend: Python/FastAPI servisi.
  - frontend: Streamlit web uygulaması.
- **Ağ Yapılandırması:**
  - Servisler arası iletişim için özel bir bridge network (battery\_net) oluşturuldu.
  - Bu sayede izole edilmiş container iletişimi sağlandı.
- **Volume Bağlantıları (Volume Mounts):**
  - models ve data dizinlerine container'lar arası paylaşımlı erişim sağlandı.
- **Bağımlılıklar:**
  - Frontend container'ı, backend servisine bağımlıdır (çalışmak için backend'e ihtiyaç duyar).
- **Kurulum:** Tek komutla kolay kurulum: `docker-compose up`



## Battery SoH Prediction API 0.1.0 OAS 3.1

/openapi.json

### default

**GET** / API Root/Health Check

**POST** /predict/soh/{battery\_id} Predict SoH for a given cycle

Predicts the State of Health (SoH) for a specific discharge cycle of a given battery.

- battery\_id**: ID of the battery (e.g., B0005, B0006, B0018).
- request\_body**: JSON containing the **cycle\_number**.

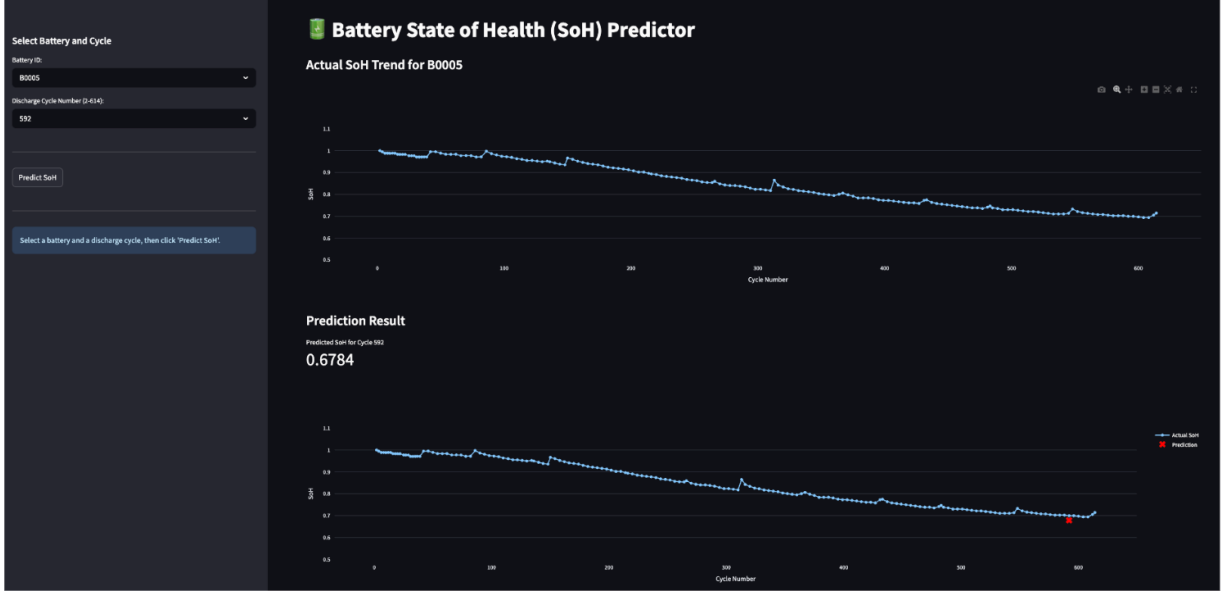
**Parameters** Try it out

Name	Description
<b>battery_id</b> <span>required</span> string (path)	<input type="text" value="battery_id"/>

**Request body** required application/json

**Example Value** | **Schema**

```
{  
  "cycle_number": 0  
}
```



## KARŞILAŞILAN TEMEL ZORLUKLAR:

- 1- Senaryoya (SoH ve SoC) ve bataryalarla çalışmaya aşina olmamak ve tahminlerin nasıl yapılacağı konusunda yeterli bilgiye sahip olmamak. Bu yüzden Data Prep. kısmında ve Model Development kısmında birtakım sıkıntılarla karşılaştım. Eksikleri gidermek için bolca research yaptım. Ancak vakit kısıtlı ve konu kompleks olduğu için ve ilerledikçe daha da kompleksleştiği için bazı kısımları tamamlamada güçlük yaşadım.
- 2- Arayüzün geliştirilmesi konusunda biraz kendi çapımda bir senaryo yarattım. Buradaki temel amacın geliştirilen modelin bir backend aracılığıyla kullanıcıya sunulması olduğu için gerçek dünya senaryolarında daha detaylı tasklarla birlikte daha planlı ve düzenli bir backend ve arayüz oluşturabilirim diye düşündüm.
- 3- Yine vaktim kalmadığı için genel olarak SoH predictionu üzerine yoğunlaştım. Araştırmalarım sonucunda SoC senaryosu için çok fazla kaynağa da ulaşamadım.
- 4- Cycle-based veriyi time-series yapısına nasıl adapte edeceğim konusunda çok fazla tereddüt ettim ve fikir değiştirdim. Bu da beni oldukça yavaşlattı.
- 5- Model development konusunda özellikle ML modellerinde overfit sorununu çözmek için yine vaktim kalmadı. Burada belki EDA kısmında kullanmam gereken featurelarla biraz fazla oynamış olabilirim. Daha basit düşünmem gerekebilirdi.

## GERÇEK DÜNYA ENTEGRASYONU ÖNERİLERİ:

### **MLOps:**

Otomatik Eğitim ve CI/CD Manuel adımları (veri indirme, işleme, model eğitimi, API güncelleme) otomatikleştiren bir CI/CD pipeline'i kurulabilir.

Model ve Veri Versiyonlama: DVC (Data Version Control) gibi araçlarla veri setlerinin ve modellerin daha sistematik versiyonlanması, eğitim parametrelerinin ve metriklerinin kaydedilmesi (MLflow Tracking, Neptune.ai vb.) tekrarlanabilirliğini açısından daha iyi olacaktır.

Microsoft'un BatteryML repositorisi(<https://github.com/microsoft/BatteryML>), bu MLOps prensiplerini batarya degradasyon analizi özelinde uygulayan güzel bir open-source kaynak. Kapsamlı MLOps uygulamaları, otomatik eğitim pipeline'ları ve model versiyon kontrolü gibi özellikleri sunuyor.

### **Edge Deployment:**

Geliştirilen SoH tahmin modeli, doğrudan bir BMS yazılımına entegre edilebilir. Bu, BMS'in daha doğru kalan ömür tahminleri yapmasını, şarj/deşarj stratejilerini optimize etmesini ve bakım ihtiyaçlarını öngörmesini sağlar. API üzerinden haberleşme bir seçenek olabileceği gibi, modelin daha optimize edilmiş bir versiyonunun (örn. TensorFlow Lite, ONNX Runtime ile) doğrudan BMS'ye gömülmesi (edge deployment) de olası bir seçenek.

### **Modelin İyileştirilmesi ve Genişletilmesi:**

SoC Tahmininin Eklenmesi: Zaman ve kaynaklar elverdiğinde, SoH'a ek olarak Şarj Durumu (SoC) tahmin yeteneğinin de eklenmesi, çözümün değerini önemli ölçüde artıracaktır. Bu, ayrı modeller veya SoH ve SoC'yi birlikte tahmin eden modelleri gerektirebilir.

### **Model Optimizasyonu ve A/B Testi:**

Özellikle CNN-BiLSTM-AM gibi daha karmaşık modellerin API entegrasyonu tamamlanmalı ve kaynak kısıtı olan sistemler (örn. edge cihazlar) için model optimizasyonu (quantization, pruning) yapılmalıdır. Yeni modeller veya hiperparametre setleri canlıya alınmadan önce A/B testleri ile mevcut modellerle karşılaştırılarak performansları doğrulanmalıdır.

### **Sürekli Öğrenme ve Yeniden Eğitim:**

İzleme sisteminden gelen uyarılara veya belirlenen periyotlara göre modellerin yeni verilerle otomatik olarak yeniden eğitilmesi yani retraining veya güncellenmesi (incremental learning) için stratejiler belirlenmelidir. Bu, modelin zamanla değişen



batarya davranışlarına adapte olmasını sağlar ve overfitting'in uzun vadeli etkilerini azaltır.

### **Veri Toplama ve İşleme Stratejisi:**

Gerçek Zamanlı Veri Akışı: Manuel veri indirme yerine, hedef sistemlerden (BMS, sensörler) gerçek zamanlı veya periyodik olarak veri akışını sağlayacak bir altyapı kurulmalıdır.

### **KULLANILAN YAPAY ZEKA MODELLERİ**

- GEMINI 2.5 PRO EXP(çoğunlukla code generation kısımlarında kullanıldı)
- MANUS (Research kısmında çok faydalı oldu)
- PERPLEXITY (Research kısmı için kullanıldı)
- GROK (Research kısmı için kullanıldı)
- CHATGPT (Case studydeki senaryoyu ve bilmediğim kavramları anlamada kullandım.)