



Mühendis ve Makina / Engineer and Machinery
<https://dergipark.org.tr/tr/pub/muhendismakina>



ELEKTRİKLİ ARABA VERİ KÜMESİ ÜZERİNDE MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ALGORİTMALARI KULLANARAK ŞARJ DURUMU TAHMİNİ

Durmuş Özkan ŞAHİN^{1*}, Mustafa ÇANDIRLI², Abdulkерim ERŞEN³,
Mikail Hüseyin AKAR⁴

¹ Ondokuz Mayıs Üniversitesi, Mühendislik Fak., Bilgisayar Mühendisliği Bölümü, Samsun
ORCID No : <http://orcid.org/0000-0002-0831-7825>

² Ondokuz Mayıs Üniversitesi, Mühendislik Fak., Bilgisayar Mühendisliği Bölümü, Samsun
ORCID No : <http://orcid.org/0009-0004-7711-4513>

³ Ondokuz Mayıs Üniversitesi, Mühendislik Fak., Bilgisayar Mühendisliği Bölümü, Samsun
ORCID No : <http://orcid.org/0009-0005-4138-5634>

⁴ Ondokuz Mayıs Üniversitesi, Mühendislik Fak., Bilgisayar Mühendisliği Bölümü, Samsun
ORCID No : <http://orcid.org/0009-0007-6440-0934>

Anahtar Kelimeler

Öz

Batarya şarj durumu,
batarya SoC , SoC tahmini,
makine öğrenmesi, derin
öğrenme

Lityum-iyon bataryalar sadece elektrikli araçlarda değil birçok farklı cihazda kullanılmaktadır. Bu bataryaların kullanıldığı cihazlarda, batarya şarj durumunun doğru tahmini kritik bir öneme sahiptir. Yapılan çalışmada elektrikli araç bataryalarının şarj durumunun değerini tahmin etmek için dört makine öğrenmesi ve dört derin öğrenme tekniği kullanılmıştır. Makine öğrenmesi teknikleri; doğrusal regresyon, k-en yakın komşu, rastgele orman regresyonu ve Xgboost algoritmalarıdır. Derin öğrenme teknikleri ise uzun kısa süreli bellek, çift yönlü uzun kısa süreli bellek, kapılı tekrarlayan birim ve evrişimli sinir ağı yöntemleridir. Batarya şarj durumu tahmini için giriş verileri olarak bataryanın akım, sıcaklık ve gerilim değerleri, çıkış verisi olarak da batarya şarj durumu değeri kullanılmıştır. Çalışmada kullanılan veri kümesi BMW i3 (60Ah) elektrikli aracının gerçek sürüşlerinden elde edilen batarya verileridir. Bu veri kümesi parça parça sürüş verilerini içermektedir. Çalışma kapsamında parça parça veriler birleştirilmiştir. Ardından veri temizleme işlemi

* durmus.sahin@bil.omu.edu.tr

doi : 10.46399/muhendismakina.1728570

uygulanarak 989670 örnekten oluşan büyük bir veri kümesi oluşturulmuştur. Ham veri kümesine Min-Max normalizasyonu, Medyan normalizasyonu, Sigmoid normalizasyonu ve D-Min-Max normalizasyonu işlemleri uygulanarak dört adet daha veri kümesi elde edilmiştir. Modellerin performansını ölçmek için Kök Ortalama Kare Hata, Ortalama Mutlak Hata ve R^2 hata metrikleri kullanılmıştır. Çalışmadan elde edilen sonuçlara bakıldığında Min-Max normalizasyonu ile oluşturulmuş veri kümesi kullanılarak rastgele orman regresyonu tekniği ve D-min-max normalizasyonu ile oluşturulmuş k-en yakın komşu tekniğinin diğer yöntemlere göre üstün olduğu görülmüştür.

STATE OF CHARGE PREDICTION USING MACHINE LEARNING AND DEEP LEARNING ALGORITHMS ON ELECTRIC CAR DATASET

Keywords

Battery state of charge,
battery SoC, SoC,
prediction, machine
learning, deep learning

Abstract

Lithium-ion batteries are used not only in electric vehicles but also in many different devices. Accurate estimation of battery charge status is of critical importance in devices using these batteries. In the study, four machine learning and four deep learning techniques were used to estimate the state of charge of electric vehicle batteries. Machine learning techniques; linear regression, k-nearest neighbor, random forest regression and Xgboost algorithms. Deep learning techniques are long short-term memory, bidirectional long short-term memory, gated recurrent unit and convolutional neural network methods. For battery charge state estimation, the battery's current, temperature and voltage values were used as input data, and the battery charge state value was used as output data. The data set used in the study is the battery data obtained from real drives of the BMW i3 (60Ah) electric vehicle. This dataset contains piecemeal driving data. Within the scope of the study, piecemeal data was combined. Then, the data cleaning process was applied and a large dataset consisting of 989670 samples was created. Four more data sets were obtained by applying Min-Max normalization, Median normalization, Sigmoid normalization and D-Min-Max normalization processes to the raw data set. With this process, Root Mean Square Error, Mean Absolute Error and R^2 error metrics were used to measure the performance of normalization models. Looking at the results obtained from the study, it was seen that the random forest regression technique using the data set created with Min-Max normalization and the k-nearest neighbor technique created with D-min-max normalization were superior to other methods.

Araştırma Makalesi

Başvuru Tarihi : 20.11.2024

Kabul Tarihi : 14.03.2025

Research Article

Submission Date : 20.11.2024

Accepted Date : 14.03.2025

Extended Abstract

Introduction/ Background

Lithium-ion batteries have become indispensable in a variety of applications, ranging from consumer electronics to electric vehicles (EVs). The accurate estimation of the state of charge (SoC) of these batteries is essential for optimizing performance, ensuring battery longevity, and enhancing user experience. In electric vehicles, an accurate SoC estimation is critical for predicting range and battery health, which directly impacts user decision-making and vehicle efficiency. This study builds upon previous work in the field of battery SoC estimation using machine learning (ML) and deep learning (DL) techniques, which have demonstrated potential for improving the accuracy of predictions compared to traditional methods. However, there is a lack of comprehensive studies comparing the performance of various ML and DL techniques in real-world battery datasets. This paper explores several such techniques to determine the most effective approach for SoC estimation.

Objectives/ Research Purpose

The primary aim of this study is to compare the performance of four ML and four DL techniques in estimating the SoC of electric vehicle batteries. The study also seeks to assess the impact of data normalization on model performance. The methods under investigation include Linear Regression, k-Nearest Neighbor (KNN), Random Forest Regression (RF), and Xgboost (XGB) as the machine learning algorithms, and Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN) as the deep learning models. The purpose is to evaluate which combination of normalization techniques and machine learning/deep learning models provides the most accurate SoC estimates.

Methods/ Methodology

The research utilized a real-world dataset obtained from the BMW i3 electric vehicle (60Ah). The dataset consists of piecemeal driving data, including various parameters such as battery current, temperature, and voltage, which are all critical factors influencing battery charge status. The raw data was preprocessed by combining the piecemeal data, followed by a cleaning process to ensure that any noise or errors were minimized. A total of 989,670 samples were used in the analysis after the preprocessing steps. To evaluate the effect of data normalization, four different normalization techniques were applied to the dataset: Min-Max normalization, Median normalization, Sigmoid normalization, and D-Min-Max normalization. These normalization techniques were tested to ensure that the machine learning models could be effectively trained and that their performance could be accurately measured. For performance evaluation, three error metrics were used: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2 error. These metrics provided a comprehensive measure of model accuracy and prediction reliability.

Results/ Findings

The study presents the performance of the various ML and DL models, which were

evaluated using different normalized datasets. It was found that the RF model, when applied to the dataset normalized using Min-Max scaling, produced the best results in terms of accuracy, followed closely by the KNN model, which performed optimally when the D-Min-Max normalization technique was used. Both models significantly outperformed the other techniques in terms of lower RMSE and MAE values and higher R^2 scores. The DL models, although promising, did not achieve the same level of accuracy as the best performing ML algorithms, particularly when the models were trained on non-normalized data or less optimal normalization schemes.

Discussion and Conclusions

The findings indicate that RF with Min-Max normalization and KNN with D-Min-Max normalization are the most effective combinations for SoC estimation in electric vehicle batteries among the techniques tested. The results suggest that traditional ML techniques can be as effective, if not more so, than deep learning models for this specific task, provided that appropriate preprocessing (normalization) is applied. Additionally, the success of Min-Max and D-Min-Max normalization techniques highlights the importance of data preprocessing in improving the performance of battery SoC estimation models. While the DL models (LSTM, Bi-LSTM, GRU, and CNN) showed some promise, they were outperformed by the RF and KNN models in terms of predictive accuracy. This could be due to the complexity of the DL models, which may require more extensive training data or more fine-tuned hyperparameters to achieve optimal performance. This study contributes to the ongoing research in the field of battery SoC estimation by providing a comparative analysis of various ML and DL models, as well as emphasizing the importance of data normalization in improving model performance. Future work could explore the use of larger and more diverse datasets, further fine-tuning of the models, and the inclusion of additional features (e.g., charging/discharging cycles) to enhance prediction accuracy.

1. Giriş

Batarya, elektrik enerjisini kimyasal enerji olarak depo edebilen, istenildiğinde bu kimyasal enerjiyi elektriksel enerji olarak verebilen bir cihazdır (Winter ve Brodd, 2004). Batarya birçok hücrenin birleşmesiyle oluşan bir cihazdır. Her bir batarya hücresine pil denir. Pil temel olarak üç maddeden oluşur. Anot pilin negatif elektrot tarafıdır. Katot pilin pozitif elektrot tarafıdır. Negatif elektrottan dışarı verilen elektrot, pozitif elektrottan alınır. Elektrolit ise iyon iletimini sağlayan kısımdır (Winter ve Brodd, 2004).

Bataryalar geçmişten günümüze birçok değişime uğramıştır. Kullanılan birçok batarya tipi vardır. Bu çalışmada lityum-iyon bataryalar üzerinde çalışılacaktır. Lityum iyon bataryalar ilk olarak Sony tarafından ticarileştirilmiştir (Nitta, Wu, Lee ve Yushin, 2015). Lityum-iyon bataryalar yüksek özgül enerjiye ve uzun döngü sayılarına sahiptirler. Bakım gerektirmezler ve diğer pillere göre çok sayıda avantajları bulunmaktadır. Diğer pil teknolojilerine göre daha uzun ömürlü olmaları, daha yüksek enerji ve güç yoğunluğu lityum-iyon pillerin en avantajlı özelliklerinden bazılarıdır (Diouf ve Pode, 2015). Sürekli gelişen ve yenilenen dünyada bu bataryalar oldukça önemlidir. Lityum-iyon bataryalar telefonlar, bilgisayarlar ve tabletler gibi birçok cihazda kullanılmaktadır (Nitta ve diğ., 2011; Thomas, Despesse, Bacquet, Fernandez, Lopez, Ramahefa-Andry ve Cassarino, 2021; Oji, Zhou, Ci, Kang, Chen ve Liu, 2021). Bu cihazların yanı sıra elektrikli arabalarda da lityum-iyon bataryalar sıkça tercih edilmektedir.

Bataryaları kullanırken batarya verilerinin durumları ne zaman bakım yapılacağına bilinmesi ve doğru çalışıp çalışmadığının kontrolü oldukça önemli problemlerden bir tanesidir (Oji ve diğ., 2021). Bu çalışmanın ana amacı elektrikli araba verilerinden yararlanılarak doğru şarj durumu (İngilizce: State of Charge: SoC) tahmini yapıp araçların ne zaman şarj olması gerektiğini doğru bir şekilde saptamaktır. Bu doğrultuda makine öğrenmesi ve derin öğrenme teknikleri kullanılarak gelişen teknolojinin merkezinde olan bataryaların SoC değerlerinin yüksek başarımla tahmin edilmesi hedeflenmektedir.

2. Bilimsel Yazın Taraması

Keskin, Günal, Urazel ve Keskin (2022), bataryanın şarj durumu tahmini için Torbalama-Rastgele Orman (RF) yaklaşımına dayalı bir model önermişlerdir. Bu yöntem ile bataryanın SoC değeri anlık akım, gerilim ve sıcaklıkla ilişkilendirilmiştir. Bu çalışmada kullanılan veri seti BMW i3 aracına ait yapılan gerçek zamanlı sürüşlerden elde edilmiştir. Veri boyutunun büyüklüğünden ve batarya karakteristik verilerinin doğrusal olmamasından dolayı min-maks normalizasyonu ile veriler normalleştirilmiştir. Model performansını değerlendirmek

için hata metriği tekniklerinden ortalama mutlak hatası (MAE) tekniği ve kök ortalama kare hatası (RMSE) tekniğini kullanılmışlardır. Önerilen Torbalama-RF yaklaşımının etkinliğini göstermek ve tahmin değerlerini karşılaştırmak amacıyla Doğrusal Regresyon (LR) ve Destek Vektör Makinesi (SVM) yaklaşımları kullanılarak da testler gerçekleştirilmiş olup, Torbalama-RF modelinden elde edilen sonuçlarla karşılaştırılma yapılmıştır. Elde edilen tahmin sonuçlarına göre Torbalama-RF modelinin diğer iki makine öğrenmesi yöntemine göre daha başarılı sonuç verdiği gösterilmiştir.

Serinbaş ve Gülbahçe (2022), SoC tahmin yöntemleri üzerine incelemeler yapmıştır. En çok bilinen SoC tahmin yöntemleri karşılaştırılmış, üstün ve zayıf yönlerinden bahsedilmiştir. Üç farklı yöntem tipi incelenmiş olup bunlar geleneksel yöntemler, durum-uzay modeline dayalı algoritmalar ve zeki algoritmalar. Karşılaştırmaların sonucunda geleneksel yöntemlerin diğer yöntemlere göre basit olduğu ve yüksek doğrulukla sonuçlar verebileceği belirtilmiştir. Fakat modelde oluşan belirsizlikler sonucu tahminin doğruluğunu etkilediğinden bahsedilmiştir. Ayrıca ölçüm süresi uzun olduğundan dolayı dinamik sistemler için uygun olmadığı belirtilmiştir. Durum uzayı tabanlı tahmin yöntemler ise doğrusal olmayan ve dinamik sistemlerde en başarılı sonuçları verdiği belirlenmiştir. Fakat kestirim doğruluğu seçilen elektrik devre modeline göre değişiklik göstermektedir. Zeki algoritmalar ile tahmin yöntemleri veriye dayalı, daha anlaşılır ve kendi kendine öğrenme yeteneğinden dolayı diğer yöntemlere göre daha üstün olduğu çıkarılmıştır. Zeki algoritmalar ile tahmin yöntemlerinin dezavantajı yapılan tahminin doğruluğu için çok sayıda veriye ihtiyaç duyulduğu saptanmıştır.

Taş, Uysal ve Bal (2023), lityum polimer pillerin SoC değerini tahmin etmek için Evrişimli Sinir Ağı (CNN) tabanlı derin öğrenme modeli kullanmıştır. SoC tahmini yapmak için pillerin akım, sıcaklık ve voltaj değerlerini ölçerek yeni bir veri seti elde etmişlerdir. 8 farklı model üzerinde deneysel çalışma göstererek SoC tahmini gerçekleştirilmiştir. CNN1 1000 dense, CNN2 10000 dense, CNN3 50000 dense, CNN4 100000 densedan oluşmaktadır. Ayrıca her bir dense batch size 500 ve 1000 olacak şekilde deneyler yapmışlardır. Yapılan testler sonucunda en uygun optimize edici adadelta seçilmiştir. Modellerde kullanılan Hiperparametre değerleri olarak Optimizer Adadelata, Rho 0.98, Epsilon 0.01, Learning rate 0.01–0.001, Epochs 20, Batch size ise 500–1000 değerindedir. Modellerin karşılaştırılması için MSE, RMSE, MAE, ve hata metrikleri kullanılmıştır. Deneyler sonucunda derin öğrenme kullanılarak yapılan testlerde katman sayısı azaldıkça tahminin doğruluğu da artmakta olduğu tespit edilmiştir. SoC tahmininde lityum-iyon veri kümesi kullanılarak %2.33 RMSE tahmin oranına ulaşılmıştır. Lityum-polimer pillerin veri kümesi kullanılarak SoC tahmininde %2.89 hata oranına ulaşılmıştır. İki hata metriği karşılaştırıldığında yaklaşık olarak tahmin oranlarının benzer olduğu görülmüştür. CNN derin öğrenme modellerinde FLOPS değeri ve parametre

sayısının RMSE hata metriği üzerinde etkisinin olmadığı fakat çalışma sürelerinde artış olduğu gösterilmiştir.

Çavuş ve Sancaktar (2022), batarya sağlık durumunun (SoH) belirlenmesi için makine öğrenmesi tekniklerini kullanmıştır. Bataryanın deşarj olduğu durumdaki akım, gerilim ve kapasite azalması gibi parametreleri kullanmıştır. SoH değerinin belirlenebilmesi için iki farklı makine öğrenme algoritması kullanmıştır. Bu algoritmalar k-En yakın Komşu Yöntemi (KNN) ve Orman Regresyon yöntemidir. Bu algoritmaları Python programlama dilini kullanarak gerçekleştirmiştir. Çalışmada kullanılan veri seti CALCE'nin internette yayınladığı 12 farklı bataryaya ait verilerdir. Her bir bataryaya ait veriler 50 şarj-deşarj döngüsü içermektedir. Bu verileri yaptığı çalışmaya göre düzenlemiştir. Oluşturulan sistemin başarısını ölçmek için iki farklı senaryo oluşturmıştır. Bu senaryolardan ilkinde 12 tane batarya (her batarya için 50 şarj-deşarj döngüsü) %70'i eğitim, %30'u test verisi olarak rastgele bölünmüştür. Oluşturulan ikinci senaryo ise 12 tane bataryanın 1 tanesini test verisi olarak kullanılmış ve bu durum her bir batarya için tekrarlanmıştır. Oluşturulan sistemin başarısını ölçmek için MAE hata metriğini kullanmıştır. Kullanılan iki algoritmada her iki senaryo için de başarılı sonuçlar elde etmiş ancak Orman Regresyon yönteminin KNN yöntemine göre daha başarılı sonuçlar verdiği saptanmıştır. Oluşturulan ikinci şartlara ait senaryo sonuçlarına bakıldığında yalnızca CX2 38 isimli bataryada KNN yönteminin daha başarılı olduğu saptanmıştır.

Vidal, Malysz, Kollmeyer ve Emadi (2020), araştırmalarında araç sektörünün daha yüksek performanslı, gelişmiş bağlantılı, uzun ömürlü ve daha düşük emisyonlu bir alana doğru ilerlediğini fark etmişlerdir. Şarj durumu tahminin araç performansı için kritik nokta olduğunu düşünmüşlerdir. Şarj durumu tahmini için ileri beslemeli sinir ağı, radyal sinir ağı, ekstrem öğrenme makineleri, destek vektör makinesi ve tekrarlayan sinir ağı yöntemlerini kullanılmışlardır. İleri beslemeli sinir ağı ve radyal sinir ağlarının hata yüzdeleri %2'yi geçerken, diğer yöntemler yaklaşık %0.5 hata yüzdesine sahip olduğunu tespit etmişlerdir. Batarya sağlık durumu tahmini için ileri sinir ağı, yinelemeli sinir ağı, radyal sinir ağı, Hamming yapay sinir ağları, destek vektör makinesi, Bayes sinir ağı kullanılmıştır. Neredeyse tüm yöntemlerle %1'in altında hata yüzdesi almışlardır. En iyi sonuç veren iki yöntem ileri beslemeli ve yinelemeli sinir ağları olduğunu tespit etmişlerdir. Sonuç olarak SoC ve SoH tahmini için kullanılan makine öğrenimi yaklaşımlarının uygun olduğu raporlanmıştır. Ayrıca kullanılan algoritmalara verilen verilerin çok önemli olduğu belirtilmiştir.

Rivera-Barrera, Muñoz-Galeano ve Sarmiento-Maldonado (2017), yaptıkları çalışmada SoC tahmin yöntemlerini incelemiş ve zorluklarından bahsetmiştir. SoC durumunun birçok değişkenden etkilenmesi, batarya sisteminin karışık olması

ve doğrusal veriler olmaması başlıca SoC tahmin zorlukları olduğunu belirtmişlerdir. Tahmin yöntemlerini doğrudan yöntemler ve dolaylı yöntemler şeklinde iki başlık altında incelemişlerdir. Doğrudan yöntemlerde yük sayma yöntemi, açık devre gerilimi yöntemi ve empedans ölçümü ile tahmin yöntemini incelemişlerdir. Dolaylı yöntemlerde ise filtreler ve yapay zekâ algoritmalarından bahsetmişlerdir. Sonuç olarak birçok yöntem ile SoC tahmini yapılabileceğini fakat SoC tahmini çok hassas olması gerektiği için kullanılan yöntemin basit olması gerektiğini vurgulamışlardır. Aynı zamanda gerçek zamanlı verileri normalizasyon işlemi ile sınırlandırılmasının daha doğru sonuçlar vereceğinden bahsedilmiştir. Son olarak uygulanacak yöntemin amaca göre tercih edilmesi gerektiğini belirtmişlerdir.

Meng, Ricco, Luo, Swierczynski, Stroe, Stroe ve Teodorescu (2017), tarafından yapılan çalışmada, popüler olan gerçek zamanlı uygulanabilir SoC tahmin yöntemlerini beş kategoride incelenmiş ve karşılaştırılmıştır. Bu yöntemler Coloumb Sayma Yöntemi, Açık Devre Gerilimi, Empedans Spektroskopisine Dayalı Yöntem, Model Tabanlı Yöntemler ve Yapay Sinir Ağı Tabanlı yöntemlerdir. Çalışmada elektrikli araç bataryaları temel alınarak incelenen yöntemler karşılaştırılmıştır. Batarya Yönetim Sistemlerinde dikkate alınması gereken üç faktöre (doğruluk, sağlamlık ve hesaplama maliyeti) göre karşılaştırmalar yapılmıştır. Her yöntem doğruluk açısından belirli durumlarda iyi sonuçlar verebilmektedir. Coloumb Sayma Yönteminin başlangıç SoC değerinin bilinmesi gerektiğinden bu da elektrikli araçlarda artan maliyet sebebiyle gerçekçi kabul edilemediği öne sürülmüştür. Açık Devre Geriliminde sürüş işlemi sırasında akım kesintisi meydana gelebileceğinden dolayı tahminde hatalar olabileceğinden bahsedilmiştir. Empedans Spektroskopisine Dayalı Yöntemin neredeyse çevrimiçi ölçülemeyeceğinden kabul edilemediği gözlemlenmiştir. Model tabanlı yöntemde doğru model seçiminin uygulandığı takdirde geçerli olabileceğinden bahsedilmiştir. Yapay Sinir Ağı Tabanlı yöntemde ise elektrikli araçların sürüşündeki veriler eğitim verilerine benzediği zaman çok yüksek doğruluk oranına ulaşabileceği ama pratik olarak farklı çalışma koşullarında farklı verilerle karşılaşılacağından bu yönden dezavantajlı olduğu öne sürülmüştür. Sonuç olarak çevrimiçi SoC tahmin yöntemlerinde üç faktöre göre önlemler alarak doğru sonuçlar elde edilmesinin mümkün olduğu vurgulanmıştır.

Sevim ve Altın (2023), lityum bataryada SoC ve SoH tahmini için öncelikle bataryayı MATLAB programında modellemiş ve bu bataryadan elde edilen SoC değerini gerçek değer olarak belirlemişlerdir. SoC ve SoH tahmini için yaygın olarak kullanılan Kalman Filtresi kullanılmıştır. Kalman filtresi sonucu elde edilen SoC değeriyle MATLAB programında modellenen bataryanın SoC değeri karşılaştırılmış ve maksimum hata değerinin pm 0.03 olduğu sonucuna varılmıştır. Elde edilen filtre sonuçları arasındaki değerlerin karşılaştırılmasıyla, SoH değerinin tahminindeki hata miktarının da pm 0.03 olduğu dolaylı olarak belirlenmiştir.

Şahin, Sahin ve Tanağardıgil (2021), yaptıkları çalışmada SoC ve SoH tahmini için bir mikroişlemci kart modülü geliştirilmiş ve bu kart aracılığıyla seri bağlı pillerin sensörler yardımıyla elde ettiği akım, sıcaklık, voltaj değerlerini alarak bilgisayara aktarılması sağlanmıştır. Elde edilen değişkenleri de SoC ve SoH tahmini için makine öğrenmesi tekniklerinden 6 tane regresyon modeli kullanılmıştır. Performans analizi için RMSE ve hata metriği kullanılmıştır. Sonuçlar karşılaştırıldığında en başarılı regresyon modeli karar ağacı regresyon modeli olduğu çıkarımında bulunulmuştur.

Eissa ve Chen (2023), yaptığı araştırmada, makine öğrenimi ile batarya şarj durumu ve elektrik araçlar için sürüş menzili tahmini yapılmıştır. Araştırmaya göre menzili ve batarya şarj durumunu etkileyen üç ana faktör vardır: araç parametreleri, kullanım ortamı ve sürücünün davranışları. Nissan Leaf aracı ile gerçek sürüş döngüsü ile veriler araç içi diagnostik sistemine yüklenmiştir. Kernel regresyon metodunu kullanılarak, batarya şarj durumunda %95 doğruluk oranı elde edilmiştir. Bu şarj durumu verilerini kullanarak sürüş menzili tahmininde %2'den düşük hata oranı elde edilmiştir.

Lin (2018), sensör gürültüsünden dolayı SoC tahmin hataları hakkında analizler yapmıştır. Akım ve gerilim sensörleri üzerinde durmuştur. Yapılan bu araştırma sensör gürültüsünün tamamen bitirmenin olanaksız olduğunu belirtmiştir. Ancak Kalman filtresi gibi algoritmaların tahmin edilen gürültü özelliklerini hesaba katarak sensör gürültüsünü, proses gürültüsüne göre azaltabileceğini söylemiştir. Gürültünün sebep olduğu SoC tahmin hatasını teorik olarak en küçük kareler altında ve kalman filtresi altında hesaplamış, karşılaştırmalarını yapmıştır. Bu yapılanları netleştirmek adına deneylerle desteklemiştir. Sonuç olarak veri sayısı ne kadar fazla ise hata oranının daha düşük olduğunu saptamıştır.

El Fallah, Kharbach, Hammouch, Rezzouk ve Jamil (2023), tarafından yapılan araştırmada laboratuvarından veriler alınıp batarya şarj durumu üç farklı sinir ağı ile tahmin edilmiştir. Testler farklı sıcaklık değerleri altında gerçekleşmiştir ve batarya birçok farklı profil ayarları altında yapılmıştır. Her deneyden sonra metotlar arası sonuçlar karşılaştırılmıştır. Derin sinir ağı ortalama hata değeri %0.4'ün altında iken, en yüksek hata oranı %2.5 olarak elde edilmiştir. Sonuç olarak derin sinir ağı yöntemi ile elde edilen tahmin sonuçları doğru olarak ifade edilmiştir.

Chemali, Kollmeyer, Preindl ve Emadi (2018), yaptıkları çalışmada derin öğrenme ile SoC kestirimi yapmışlardır. Lityum iyon pillerden veri toplamışlardır. Bu verileri kendi ortamlarında, farklı koşullarda çalıştırdıkları pillerden almışlardır. Farklı sıcaklıklarda alınan verilerin tahmini yapılmıştır. Ağları eğittikten sonra çıkan sonuç 25 santigrat derecede %1.20 MAE, -20 santigrat derecede ise %2.27 MAE olduğunu tespit etmişlerdir. Daha sonra veri miktarını arttırarak hata değerinin %1 MAE'nin altına düşebileceğini tespit etmişlerdir.

Zahid, Xu, Li, Li ve Li (2018), elektrikli araçların farklı sürüş döngülerinden elde edilen verileri kullanarak pil SoC'sini tahmin etmek için çıkarımlı kümeleme tabanlı nöro-bulanık mimariyi kullanmışlardır. 10 farklı sürüş döngüsü verilerini ADVISOR kullanarak toplamışlardır. 10 sürüş döngüsünün her biri için %0.1'den daha az maksimum mutlak tahmin hatasıyla SoC tahmini yapmıştır. Çıkarımlı kümeleme tabanlı nöro-bulanık mimari, SoC tahmin sonuçlarına bakıldığında diğer makine öğrenme tekniklerine göre daha başarılı sonuçlar elde etmiştir.

Chandran, Patil, Karthick, Ganeshaperumal, Rahim ve Ghosh (2021), araştırmasına göre, lityum-iyon bataryaların şarj durumunu doğru tahmin etmek performans açısından önemlidir. Şarj durumu tahmini için yapay sinir ağı dâhil 6 farklı makine öğrenimi metodu kullanılmıştır. Makineye giriş verisi olarak akım ve voltaj verilmiştir. Ortalama olarak %82 doğruluk oranı elde eden yapay sinir ağı ve Gauss regresyonu, diğer tahmin metodlarına göre daha doğru sonuçlar vermiştir.

Tunç ve Erdem, (2024), batarya sağlık durumu tahmini için farklı derin öğrenme tekniklerinden yararlanmışlardır. Derin öğrenme tekniklerinden LSTM, BİLSTM, GRU ve GRU-AM algoritmaları üzerine yoğunlaşmışlardır. Yapay zekânın gelişmesiyle birlikte makine öğrenimi tekniklerinin batarya ömrünü tahmin etmede daha basit ve daha hızlı olduğunu belirtmişlerdir. Yaptıkları çalışmada NASA'nın lityum iyon pil veri setini kullanmışlardır. Bataryaları şarj/deşarj zamanlarında veri toplama yapılmıştır. Gerilim, akım ve sıcaklık parametrelerine göre bir tahmin yapmışlardır. Önerilen yapı üç adımdan oluşur: Veri ön işleme, eğitim ve tahmin. Veri ön işleme için veri temizleme ve min-max normalizasyonu kullanmışlardır. Eğitim için sırasıyla LSTM, GRU, BİLSTM ve GRU-AM'ye dayalı bir model seçmek için eğitim ve doğrulama setlerini kullanmışlardır. MATLAB programlama dili ile işlemleri gerçekleştirmişlerdir. MAE, RMSE ve MAPE hata metrikleri ile aldıkları sonuçları karşılaştırmışlardır. Önerdikleri GRU-AM yöntemi RMSE'de %14 ile %35, MAE'de %21 ile %36 ve MAPE'de %21 ile %35'lik iyileştirme gösterdiğini belirtmişlerdir. BiLSTM hem gelecek hem de geçmiş bağımlılıkları sebebiyle, LSTM ve GRU ise Dikkat Mekanizmasına sahip olmayışlarıyla önerilen yöntemin gerisinde kaldığını saptamışlardır.

Literatürdeki güncel çalışmaların büyük bir kısmı belirli makine öğrenmesi veya derin öğrenme modellerine odaklanarak SoC tahmini gerçekleştirmiş, ancak farklı model türlerinin geniş kapsamlı bir karşılaştırmasını yapmamıştır. Ayrıca, birçok çalışma sınırlı veri setleriyle veya laboratuvar ortamında oluşturulmuş verilerle deneyler gerçekleştirmiştir. Bu çalışmada ise, gerçek sürüşlerden oluşan büyük ölçekli bir veri kümesi kullanılarak SoC tahmini yapılmış ve dört farklı makine öğrenme yöntemi ile dört farklı derin öğrenme yöntemi karşılaştırılmıştır. Böylece, literatürde eksik olan kapsamlı model değerlendirme ve normalizasyon tekniklerinin etkisini ele alan bir analiz sunulmuştur. Ayrıca, literatürde

genellikle tek bir normalizasyon yöntemi ile çalışılmışken, bu çalışmada farklı normalizasyon tekniklerinin model performansına etkisi de detaylı olarak incelenmiştir. Bu sayede, literatürdeki eksik yönler giderilerek, farklı model ve veri işleme tekniklerinin SoC tahmini üzerindeki etkileri kapsamlı bir şekilde değerlendirilmiştir.

2.1 Motivasyon ve Katkı

Bu çalışmanın temel hedefi, farklı makine öğrenimi ve derin öğrenme tekniklerini kullanarak elektrikli araç bataryalarının SoC tahmin performanslarını kapsamlı bir şekilde karşılaştırmaktır. Literatürde çoğunlukla belirli algoritmaların tek başına incelendiği çalışmalar bulunurken, bu çalışmada hem geleneksel makine öğrenme algoritmaları hem de derin öğrenme modelleri birlikte ele alınarak avantajları ve dezavantajları detaylı şekilde analiz edilmiştir. Ayrıca, büyük ölçekli ve gerçek sürüş verilerinden oluşan bir veri kümesi üzerinde farklı normalizasyon tekniklerinin model başarımına etkisi incelenerek, SoC tahmininde en iyi yöntem ve veri işleme stratejileri belirlenmiştir. Bu sayede, batarya yönetim sistemlerinde daha güvenilir ve optimize edilmiş SoC tahmini yapılmasına katkı sağlanması amaçlanmıştır. Bu çalışmanın ana katkıları şöyle özetlenebilir:

- Farklı sürüş zamanlarının yer aldığı veriler bir araya getirilerek büyük bir veri kümesi üzerinde algoritmalar denenmiştir. Veri kümesi üzerinde yer alan örnek sayısı 989670'dir. Literatürde çok daha büyük veri kümeleriyle gerçekleştirilen çalışmalar bulunmaktadır. Ancak bu çalışmada, sürüş verileri içeren geniş ölçekli bir veri kümesi üzerinde deneyler gerçekleştirilmiş ve elde edilen sonuçlar analiz edilmiştir.
- 4 farklı geleneksel makine öğrenmesi ve 4 farklı derin öğrenme tekniği kullanılarak SoC tahmini gerçekleştirilmiştir.
- 4 farklı normalizasyon tekniği kullanılarak makine öğrenmesi ve derin öğrenme algoritmalarının başarımlarına olan etkileri incelenmiştir.
- RMSE, MAE ve metrikleri olmak üzere 3 farklı metrik ile algoritmaların tahmin ettiği SoC hatası hesaplanmıştır.

2.2 Organizasyon

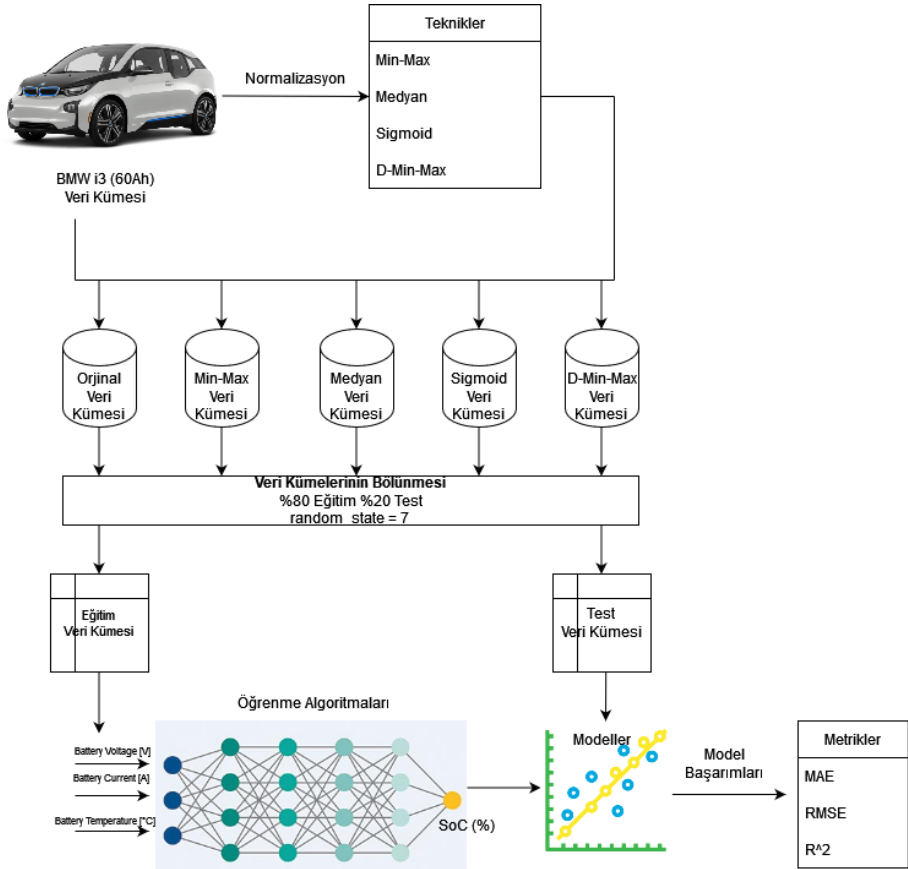
Çalışmanın geri kalan kısımları şöyle organize edilmiştir: Deneysel ayarlamalar bölümünde çalışma alt yapısında kullanılan programlama diline, dile ait kitaplıklara ve veri kümesine değinilecektir. Bu bölümün devamında normalizasyon tekniklerinin matematiksel alt yapısı verilecektir. Ardından çalışmada kullanılan makine öğrenmesi teknikleri kısaca anlatılacaktır. Daha sonra makine öğrenmesi tekniklerinin başarımlarını ölçmede kullanılan metriklere yer verilecektir. Bu bölümün ardından çalışmada elde edilen sonuçlar verilip yorumlanacaktır. Son

olarak çalışma genel olarak değerlendirilip bu alanda çalışma yapmak isteyen araştırmacılara bilgiler verilecektir.

3. Deneysel Ayarlamalar

Bu çalışmada araştırma ve yayın etiğine uyulmuştur.

Bu bölümde çalışmada kullanılan programlama diline, programlama diline ait kitaplıklara ve veri kümesine değinilecektir. Çalışma kapsamında takip edilen adımlar ve geliştirilen SoC tahmin sisteminin mimarisi Şekil 1’de verilmektedir.



Şekil 1. SoC Tahmin Sisteminin Mimarisi

3.1 Kullanılan Programlama Dili ve Kitaplıklar

Bu çalışmada, SoC tahmininde çeşitli makine öğrenimi ve derin öğrenme tekniklerini uygulamak için Python programlama dili kullanılmıştır. Aynı zamanda şu

Python kütüphaneleri ve araçlarından yararlanılmıştır:

- Pandas: Veri manipülasyonu ve analizi için kullanılan güçlü bir kütüphanedir.
- Numpy: Bilimsel hesaplamalar ve çok boyutlu diziler üzerinde verimli çalışma için temel bir kütüphanedir.
- Scikit-learn: Makine öğrenimi algoritmalarının uygulanması için kapsamlı bir kütüphanedir.
- Tensorflow ve Keras: Derin öğrenme modelleri oluşturmak için kullanılan kütüphanelerdir.
- Xgboost: Gradient boosting algoritması için optimize edilmiş, yüksek performanslı bir kütüphanedir.
- Matplotlib ve Seaborn: Veri görselleştirme için kullanılan kütüphanelerdir.

3.2 Kullanılan Veri Kümesi

Bu çalışmada kullanılan veri kümesi BMW i3 (60Ah) aracının gerçek sürüş verilerinden oluşmaktadır. Veri kümesi IEEEDataPort sitesinde bulunan BATTERY AND HEATING DATA IN REAL DRIVING CYCLES adlı çalışmadan alınmıştır (Steinstraeter, Buberger ve Trifonov, 2020). İlk olarak .csv dosya formatında bulunan 70 gezi verisi .xlsx dosya formatına dönüştürülmüştür. Ardından bunlardan "Battery Voltage [V]", "Battery Current [A]", "Battery Temperature [°C]" ve "SoC [%]" kolonları alınıp tek bir Excel dosyasında birleştirilerek 1094793 veriden oluşan veri kümesi elde edilmiştir. SoC değerine sahip olmayan 30748 veri tespit edilip veri kümesinden kaldırılmıştır. Ayrıca tekrarlanan 74375 veri tespit edilmiştir ve bu veriler veri kümesinden kaldırılmıştır. Bu işlemler sonucunda veri setinin tutarlılığı artırılmış, gürültü azaltılmış ve modellerin daha doğru tahminler yapabilmesi için güvenilir bir veri kümesi oluşturulmuştur. Veri temizleme adımları sayesinde, özellikle normalizasyon süreçlerinin daha etkili çalışması ve model tahmin doğruluğunun artması hedeflenmiştir. Veri kümesinin son halinde 989670 tekrarlanmayan veri bulunmaktadır. Son olarak yerine koy özelliğiyle, verilerin noktaları (.) virgül işaretiyle (,) değiştirilmiştir. Bu sayede veri hücrelerinin "Number" formatına dönüştürülmesi sağlanmıştır. Bu adımlar sayesinde veri kümesi Pandas kütüphanesi için uygun hale getirilmektedir. Ham verinin yanında oluşan veri kümesine normalizasyon teknikleri uygulanarak 4 adet normalize edilmiş veri kümesi elde edilmiştir.

Veri kümesinin SoC dağılımına bakıldığında, SoC değerleri 15.4 ile 88.5 arasında değişmektedir. Ortalama SoC değeri 64.88 olup, standart sapma 15.02 olarak hesaplanmıştır. Bu durum, SoC değerlerinin büyük ölçüde 50'nin üzerinde yoğunlaştığını göstermektedir. Nitekim 822187 örnek SoC ≥ 50 seviyesinde bulunurken, 50'nin altındaki örnek sayısı 167483 olarak bulunmuştur. Bu da veri

kümesinde düşük SoC seviyelerine ait örneklerin daha az temsil edildiğini göstermektedir.

Tablo 1. Veri Kümesinin Bir Bölümü

Battery Voltage [V]	Battery Current [A]	Battery Temperature [°C]	SoC [%]
366,70	-6,19	3,00	48,80
350,58	-159,47	10,00	54,50
376,47	-11,27	21,00	58,80
341,28	-95,50	6,00	38,90
380,40	0,40	20,00	64,90
350,38	-108,93	15,00	34,32
370,11	-58,59	6,00	69,20
370,86	-42,71	5,00	75,10
383,83	-18,11	24,00	78,00
349,31	-122,08	12,00	48,60
385,16	4,31	5,00	80,40
380,53	-20,85	13,00	71,50
385,33	-9,20	8,00	79,90
381,44	-9,63	22,00	67,70
357,99	-35,52	15,00	36,20
361,01	-23,74	16,00	37,80
388,81	9,02	20,00	79,90
373,58	-4,82	14,00	62,90
377,61	99,24	32,00	54,19
361,58	-48,95	7,00	50,50

Tablo 1’de veri kümesinin bir bölümü gösterilmektedir. Oluşturulan veri kümesinin %80’i modelleri eğitmek için kullanılırken, %20’si ise model başarımını test etmek için kullanılmaktadır. Adil bir karşılaştırma yapmak için, tüm algoritmaların aynı eğitimi ve aynı test bölümünü kullanması gerektiğinden, Sklearn kütüphanesindeki “train_test_split” modülünün “random_state” parametresi her durumda 7 değeriyle çalıştırılarak deneyler gerçekleştirilmiştir. Bu parametrenin aynı değerde çalıştırılması sonuçların tekrar edilebilirliğini sağlamak ve farklı algoritmalarda aynı veri bölünmelerini kullanarak adil bir karşılaştırma yapmak açısından oldukça önemlidir. Aksi takdirde, farklı değerler ile modelin öğrenme süreci değişebilir ve sonuçlar arasında tutarsızlıklar oluşabilmektedir.

3.3 Normalizasyon Teknikleri

Verilerin giriş ve çıkışlarına belirli ön işlem adımları uygulanarak verilere uygulanacak eğitim işlemi daha verimli hale getirilebilir. Normalizasyon işlemi ham verilere uygulanır ve bu verilerin eğitim için uygun veri setinin hazırlanmasında önemli rol oynar. Eğitim daha hızlı ve daha doğru şekilde gerçekleştirilebilir. Normalizasyon işlemlerinde farklı teknikler kullanılabilir. Literatürde birçok veri normalizasyon çeşidi vardır. Bunlar; Min-Max normalizasyonu, Medyan normalizasyonu, Sigmoid normalizasyonu, Z-Score normalizasyonu ve D-Min-Max normalizasyonu gibi sıralanabilir (Yavuz ve Deveci, 2012).

Yapılan çalışmada 989670 örnekten oluşan veri ayrı ayrı Min-Max normalizasyonu, Medyan normalizasyonu, Sigmoid normalizasyonu ve D-Min-Max normalizasyonu ile normalize edilmiştir. Normalize edilmemiş veri kümesi ile birlikte toplam 5 veri kümesi üzerinde çalışılmıştır. Kullanılan normalizasyon tekniklerinin detayları şöyledir:

Min-Max Normalizasyonu: Bu normalizasyon tekniği, verileri doğrusal bir biçimde normalize eder. Burada minimum, verinin alabileceği en düşük değeri temsil ederken, maksimum ise en yüksek değeri ifade etmektedir (Yavuz ve Deveci, 2012). Min-Max tekniği veriyi genellikle 0 ile 1 arasındaki bir aralığa ölçekler. Ancak, başka bir ölçek aralığı da belirlenebilir. Verileri $[0, 1]$ aralığına ölçeklendirmek için Eşitlik 1 kullanır. Eşitlik 1’de yer alan x' normalize edilmiş değeri, normalize edilmek istenen değeri, $x_{\min}$ ilgili girdideki en küçük değeri ve $x_{\max}$ ise ilgili girdideki en büyük değeri temsil etmektedir.

$$x' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

Medyan Normalizasyonu: Her bir örneğin medyan değerini temel alarak, medyan normalizasyon yöntemi uygulanır (Yavuz ve Deveci, 2012). Bu normalizasyon tekniğinin matematiksel gösterimi Eşitlik 2’de verilmiştir. Eşitlik 2’de yer alan x' ve x_i Eşitlik 1’de tanımlandığı gibidir. $\text{medyan}(\text{data}_i)$ ise girdinin medyan değerini temsil etmektedir.

$$x' = \frac{x_i}{\text{medyan}(\text{data}_i)} \quad (2)$$

Sigmoid Normalizasyonu: Bu teknik verileri $[0, 1]$ veya $[-1, 1]$ aralığına ölçeklemektedir. Sigmoid normalizasyonunun matematiksel gösterimi Eşitlik 3’te gösterilmektedir. Eşitlik 3’te yer alan x' ve x_i Eşitlik 1’de tanımlandığı gibidir. e ise doğal logaritmayı göstermektedir.

$$x' = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} \quad (3)$$

D-Min-Max Normalizasyonu: Verileri istenilen aralıkta ölçeklemek için kullanılan normalizasyon tekniğidir. Bu normalizasyon tekniğinin matematiksel alt yapısı Eşitlik 4'te verilmektedir. Eşitlik 4 kullanılarak veriler [0.1, 0.9] aralığında normalize edilmektedir. Çalışmada da bu eşitlik kullanılarak normalizasyon işlemi gerçekleştirilmektedir. Eşitlik 4'te yer alan x' ve x_i Eşitlik 1'de tanımlandığı gibidir.

$$x' = 0.8 \times \frac{x_i - x_{min}}{x_{max} - x_{min}} + 0.1 \quad (4)$$

Bu yöntemler, literatürde makine öğrenmesi ve derin öğrenme uygulamalarında yaygın olarak kullanıldığı ve farklı veri dağılımlarına karşı avantajlar sunduğu için seçilmiştir (Yavuz ve Deveci, 2012; Xie, Suvarna, Li, Zhu, Cai ve Wang, 2021). Min-Max normalizasyonu, verileri belirli bir aralıkta ölçekleyerek birçok modelin daha istikrarlı öğrenmesini sağlarken, Medyan normalizasyonu özellikle aykırı değerlere karşı daha dayanıklıdır (Yavuz ve Deveci, 2012). Sigmoid normalizasyonu, sinir ağları gibi modellerde öğrenme sürecini iyileştirirken, D-Min-Max normalizasyonu belirli bir ölçek aralığında model eğitim aşamasını kararlı bir hale getirmek için kullanılmaktadır. Bu yöntemlerin seçimi, literatürde yaygın kullanımları ve model performansına olan olumlu etkileri göz önünde bulundurularak yapılmıştır.

3.4 Makine Öğrenmesi ve Derin Öğrenme Teknikleri

Yapılan çalışmada 4 tane geleneksel makine öğrenimi tekniği ve 4 tane derin öğrenme tekniği kullanılmıştır. Çalışmanın amaçlarından biri olan, farklı tekniklerin karşılaştırılması ve en yüksek başarı elde eden makine öğrenimi tekniğinin tespit edilmesi bu sayede gerçekleştirilmektedir. Kullanılan geleneksel makine öğrenimi teknikleri; LR tekniği, KNN, RF tekniği ve Xgboost (XGB) tekniğidir. Derin öğrenme tekniklerinden; LSTM, çift yönlü uzun kısa süreli bellek (BiLSTM), kapılı tekrarlayan birim (GRU) ve CNN algoritmaları kullanılmıştır. Kullanılan algoritmaların alt yapıları şöyle özetlenebilir:

Doğrusal Regresyon: Doğrusal regresyon, x olarak ifade edilen bir veya daha fazla bağımsız değişkenle ilişkilendirilen sayısal bir bağımlı değişken olan y arasındaki ilişkiyi modelleme yöntemidir. Eğer regresyon modelinde yalnızca bir bağımsız değişken varsa, bu model basit doğrusal regresyon olarak adlandırılır. Ancak modelde birden fazla açıklayıcı bağımsız değişken varsa, bu durum çoklu doğrusal regresyon olarak tanımlanır. Doğrusal regresyonda bağımlı değişken sayısal bir değişken olmak zorundadır (Kılıç, 2013). Doğrusal regresyon tekniği-

nin matematiksel gösterimi Eşitlik 5'te gösterilmiştir. Eşitlik 5'te bağımlı değişkeni yani tahmin edilen değeri, $x_1, x_2, \dots, x_n$ bağımsız değişkenleri, B_0 doğrunun y eksenini kestiği noktayı, B_1 ise doğrunun eğimini başka bir ifadeyle bağımsız değişkenin her bir birimlik artışının bağımlı değişkende ne kadar değişiklik oluşturduğunu temsil etmektedir.

$$y = B_0 + B_1 * x_1 + B_2 * x_2 + \dots + B_n * x_n \quad (5)$$

KNN: KNN algoritması hem sınıflandırma hem de regresyon problemlerinde kullanılan basit ama güçlü bir makine öğrenimi algoritmasıdır. KNN, bir veri noktasının sınıfını veya değerini, veri setindeki en yakın komşusuna dayanarak tahmin eder. İlk olarak eğitim ve test verileri oluşturulmaktadır. Ardından k değeri belirlenmektedir. Bu çalışmada KNN algoritmasının k parametresi "Randomized-SearchCV" tekniği kullanılarak aranmıştır. Bu süreçte farklı k değerleri denenmiş ve modelin performansı değerlendirilmiştir. Yapılan deneyler sonucunda k = 9 değerinin en iyi sonucu verdiği gözlemlenmiştir. Bu nedenle, modelin doğruluğunu maksimize etmek ve en iyi tahmin performansını elde etmek için k = 9 olarak belirlenmiştir.

Rastgele Orman Algoritması: RF algoritması, sınıflandırma ve regresyon problemlerinde kullanılabilir. Bu algoritma, birden fazla karar ağacı oluşturarak sınıflandırma işleminde doğruluğu artırmayı amaçlar. RF, birbirinden bağımsız çalışan çok sayıda karar ağacının bir araya gelerek en yüksek puan alan değeri seçme sürecidir. RF algoritması ön tanımlı parametreler ile çalıştırılarak deneyler gerçekleştirilmiştir. "n_estimators" ağaç sayısını gösteren parametredir ve ön tanımlı olarak 100 değerini almaktadır.

XGBoost Algoritması: XGB, eğim artırma tekniğini kullanan bir makine öğrenimi algoritmasıdır ve karar ağaçlarına dayanır (Kelle ve Yüce, 2022). Temel çalışma prensibi, mevcut modelin hatalarını minimize edecek şekilde yeni bir model oluşturmaktır. Bu süreçte, kayıp fonksiyonunun ikinci türevine dayalı bir adım boyutu hesaplanır. Bu adım, yeni modelin çıktısını, mevcut modelin hatalarını en aza indirecek şekilde ayarlamaya yardımcı olur.

LSTM: Hochreiter ve Schmidhuber tarafından tasarlanan tekrarlayan sinir ağının geliştirilmiş bir versiyonudur (Hochreiter ve Schmidhuber, 1997). Zaman serisi ve dizi tahminleri için çok uygundur. LSTM bir bellek hücresi ile verileri uzun vadeli verileri hafızasında tutabilir, bu işlem de eğitimin daha iyi olmasında önemli rol oynar. Bellek hücresi 3 kapı tarafından kontrol edilir: Giriş kapısı, Çıkış kapısı, Unutma kapısı. Bu kapılar, LSTM ağlarının, ağ boyunca akarken bilgileri seçici olarak tutmasına veya atmasına olanak tanır. Bu işlem sayesinde LSTM uzun vadeli bağımlılıkları öğrenebilir. Çalışmada kullanılan LSTM modelinde optimizasyon

yon algoritması olarak Adam (öğrenme oranı = 0.001), aktivasyon fonksiyonu olarak ReLU, kayıp fonksiyonu olarak Ortalama Kare Hata (Mean Squared Error: MSE) kullanılmıştır. Düşürme oranı (dropout) ise 0.5 olarak alınmıştır. Model 20 epoch boyunca, batch boyutu 32 olacak şekilde eğitilmektedir.

BiLSTM: BiLSTM, biri girdiyi ileri yönde işlemek, diğeri ise geri yönde işlemek için olmak üzere iki LSTM katmanı içeren bir dizi modeli için kullanılan derin sinir ağıdır (Shiri, Perumal, Mustapha ve Mohamed, 2023). Bu yaklaşımın arkasındaki anlam, verileri çift yönlü işleyerek, oluşturulacak modelin veriler arasındaki ilişkiyi daha iyi anlayabilmesidir. Çalışmada kullanılan BiLSTM modelinde optimizasyon algoritması olarak Adam (öğrenme oranı = 0.001), aktivasyon fonksiyonu olarak ReLU, kayıp fonksiyonu olarak MSE kullanılmıştır. Düşürme oranı (dropout) ise 0.5 olarak alınmıştır. LSTM katmanı, çift yönlü (Bidirectional) ve 50 nöronlu olarak tanımlanmıştır. Girdi verisi, (3,1) boyutunda olacak şekilde modellenmiştir. Model, 20 epoch boyunca batch boyutu 32 olacak şekilde eğitilmektedir.

GRU: Bazı durumlarda LSTM'ye göre daha avantajlı olan RNN türüdür (Shiri ve diğ., 2023). LSTM'ye göre daha az bellek kullanır ve daha hızlı çalışır. Fakat uzun veri kümelerinde LSTM'nin daha doğru sonuçlar verdiği araştırmalarda görülmüştür. Kullanılan GRU modelinde optimizasyon algoritması olarak Adam (öğrenme oranı = 0.001), aktivasyon fonksiyonu olarak ReLU, kayıp fonksiyonu olarak MSE kullanılmıştır. Model, 10 epoch boyunca batch boyutu 32 ile eğitilmektedir. GRU katmanı, 50 nöronlu olarak tanımlanmış ve aşırı öğrenmeyi önlemek için dropout oranı 0.5 olarak belirlenmiştir. Girdi verisi, (3,1) boyutunda olacak şekilde şekillendirilmiştir.

CNN: Evrimsel sinir ağları, derin öğrenmenin bir alt dalıdır (Li, Liu, Yang, Peng ve Zhou, 2021). CNN genellikle görüntü sınıflandırma problemlerinde kullanılır. 3 ana katmanı vardır: Evrimsel katman, Havuz katmanı, Tam bağlı katman. Evrişim katmanı, evrişimli bir ağı ilk katmanıdır. Evrişimli katmanları ek evrişimli katmanlar veya havuzlama katmanları takip edebilirken, tam bağlı katman son katmandır. Her katmanla birlikte CNN'nin karmaşıklığı artar. Karmaşıklık arttıkça model daha detay ayrıntıları öğrenmeye başlar. Kullanılan 1D CNN modelinde optimizasyon algoritması olarak Adam (öğrenme oranı = 0.001), aktivasyon fonksiyonu olarak ReLU, kayıp fonksiyonu olarak MSE kullanılmıştır. Model, 10 epoch boyunca batch boyutu 32 ile eğitilmektedir. İlk katmanda 64 filtreli, çekirdek boyutu 2 olan bir Conv1D katmanı bulunmaktadır. Ardından, havuzlama işlemi (MaxPooling) için havuz boyutu 2 olarak belirlenmiştir. Tam bağlantılı (dense) katmanda 50 nöron kullanılmıştır ve aşırı öğrenmeyi (overfitting) önlemek için dropout oranı 0.5 olarak belirlenmiştir.

3.5 Hata Metrikleri

Hata metrikleri, bir modelin tahminlerinin gerçek değerlerden ne kadar farklı olduğunu değerlendirmek için kullanılan ölçütlerdir. Bir modelin performansını ölçmek ve karşılaştırmak için kullanılırlar. Hata metrikleri, modelin ne kadar doğru tahminler yaptığını, ne kadar hatası olduğunu ve hangi alanlarda iyileştirme gerektiğini belirlemeye yardımcı olur. Bu çalışmada MAE, RMSE ve hata metrikleri kullanılmıştır. Bu metriklerin matematiksel gösterimleri şöyledir:

MAE: Gerçek değerler ile tahmin edilen değerler arasındaki hataların mutlak değerlerinin ortalamasıdır. Eşitlik 6'da bu metriğin altyapısı verilmiştir. Eşitlik 6'da veri noktalarının sayısını, gerçek değerleri, tahmin edilen değerleri temsil etmektedir.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

RMSE: MSE metriğinin karekökü alınarak elde edilir. RMSE'nin en büyük farkı, büyük hatalara daha fazla ağırlık vermesidir. Bu, hata karelerinin ortalamasını almak yerine doğrudan hata değerlerinin ortalamasını alarak gerçekleşir. Eşitlik 7'de bu metriğin matematiksel gösterimi verilmiştir. n , y_i ve $\hat{y}_i$ Eşitlik 6'da tanımlandığı gibidir.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

R^2 : Modelin açıkladığı varyansın oranını ifade eder. Bu metrik sonucu 1'e ne kadar yakın ise regresyon modeli o kadar başarılıdır anlamına gelmektedir. Eşitlik 8'de bu metriğin matematiksel gösterimi verilmiştir. n , y_i ve $\hat{y}_i$ Eşitlik 6'da tanımlandığı gibidir. $\bar{y}$ ise gerçek değerlerin ortalamasını temsil etmektedir.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (8)$$

RMSE, tahmin hatalarının karelerinin ortalamasının karekökünü alarak büyük hataları daha fazla cezalandıran bir metrik olup, modelin genel hata seviyesini anlamada önemlidir. MAE, tahmin edilen ve gerçek değerler arasındaki mutlak farkların ortalaması olarak, modelin tahminlerinin doğruluğunu daha doğrudan ve yorumlanabilir bir şekilde sunar. R^2 ise modelin bağımsız değişkenlerle hedef değişken arasındaki ilişkiyi ne kadar iyi açıkladığını gösteren bir belirleme katsayısıdır ve modelin genel performansını değerlendirir. Bu üç metrik birlikte kullanılarak, modelin hata düzeyi, tahmin doğruluğu ve genel uyumu kapsam-

lı bir şekilde incelenmiştir. Böylece, hem tahmin hatalarının büyüklüğü hem de modelin veriye ne kadar iyi uyum sağladığı değerlendirilerek daha sağlam bir performans analizi yapılmıştır.

4. Elde Edilen Sonuçlar

Yapılan çalışmada 5 farklı veri kümesi, 4 farklı makine öğrenimi tekniği ve 4 farklı derin öğrenme tekniği ile çalışılmıştır. Yapılan çalışmalar sonucunda 3 farklı hata metriği ile sonuçlar elde edilmiştir. Elde edilen sonuçlar Tablo 2’de verilmiştir.

Tablo 2’de görüldüğü üzere alınan sonuçlar, bu denli büyük bir veri kümesi için memnun edici düzeydedir. Literatürde çok daha büyük veri kümeleriyle gerçekleştirilen çalışmalar bulunmaktadır. Ancak bu çalışmada, sürüş verileri içeren kapsamlı bir veri kümesi üzerinde deneyler gerçekleştirilmiş ve elde edilen sonuçlar değerlendirilmiştir. Elde edilen sonuçlara göre LR tekniği diğer algoritmalara göre başarısız kalmıştır. En başarılı sonuçlar ise KNN ve RF teknikleri ile alınan sonuçlar olmuştur. Normalizasyon işlemi bu veri kümesi için yüksek derecede değişiklik göstermese de normalize edilmiş veri kümeleri ile elde edilen sonuçlar, genel itibarıyla normal verilerden elde edilen sonuçlardan daha başarılı sonuç vermiştir. Özellikle medyan, min-max ve D-min-max normalizasyon teknikleri ile elde edilen sonuçlarda normalizasyonun etkisi görülmüştür.

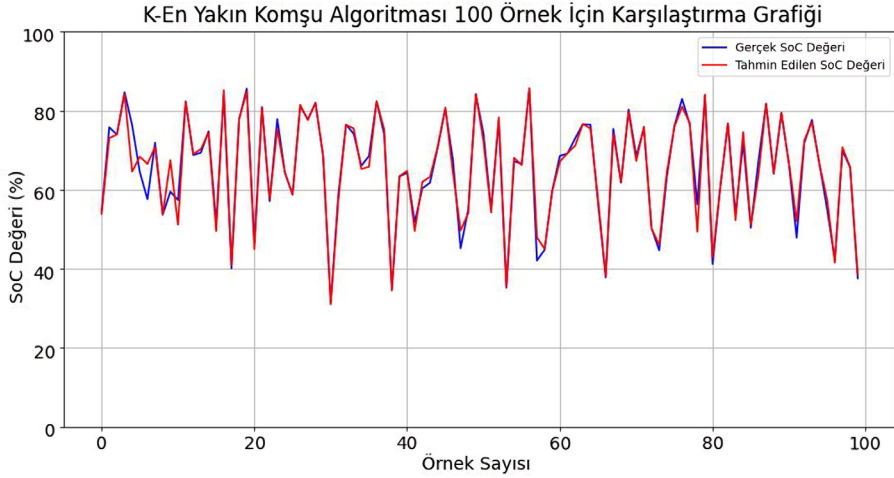
Tablo 2’de koyu ile belirtilmiş sonuçlar, elde edilen en iyi iki sonuçtur. Genel itibarıyla KNN ve RF teknikleri en başarılı sonuçları veren iki makine öğrenimi tekniği olmuştur. KNN tekniğinde en iyi sonuç MinMax ve DMinMax ile edilmiş veri kümesinden alınmıştır. metriğine göre yaklaşık 0.98 oranında başarı elde edilmiştir. RF tekniğinde ise MinMax normalizasyonu ile elde edilmiş veri kümesinde en iyi sonuç alınmıştır. R^2 metriğine göre 0.9769 oranında başarı elde edilmiştir. KNN ve RF algoritmaları en iyi sonucu verdiği için test verisinden rastgele alınan 100 örneğin gerçek SoC ve tahmini SoC değerlerine karşılık gelen karşılaştırılma grafikleri Şekil 2 ve Şekil 3’te verilmiştir.

Genel olarak, derin öğrenme algoritmalarının büyük veri kümelerinde daha iyi performans gösterdiği bilinmektedir. Ancak bu çalışmada, KNN ve RF’nin derin öğrenme yöntemlerine kıyasla daha başarılı sonuçlar vermesi, kullanılan öznitelik kümesi ve model optimizasyon süreçleriyle ilişkilendirilebilir. Öncelikle, bu problemde öznitelikler belirli ve tanımlı olduğundan, güçlü bir özellik mühendisliği sürecine ihtiyaç duyulmamış ve geleneksel makine öğrenme algoritmaları doğrudan etkili bir şekilde çalışmıştır. Bunun yanı sıra, derin öğrenme modellerinin başarılı olabilmesi için hiperparametrelerin dikkatlice ayarlanması büyük önem taşımaktadır. Optimizasyon sürecinde öğrenme oranı, katman sayısı, aktivasyon fonksiyonları gibi faktörler doğru şekilde ayarlanmadığında, model beklenen performansı sergileyemeyebilir. Bu bağlamda, RF ve KNN gibi yöntemler,

daha az parametre gereksinimi ve kolay optimize edilebilir yapıları sayesinde bu problemde daha iyi sonuç vermiştir. Sonuç olarak, derin öğrenmenin genellikle daha üstün performans göstermesi beklenirken, bu çalışmada öznetelik kümesinin tanımlı olması ve hiperparametre optimizasyonunun kritik rolü, klasik yöntemlerin daha iyi performans göstermesine neden olmuştur.

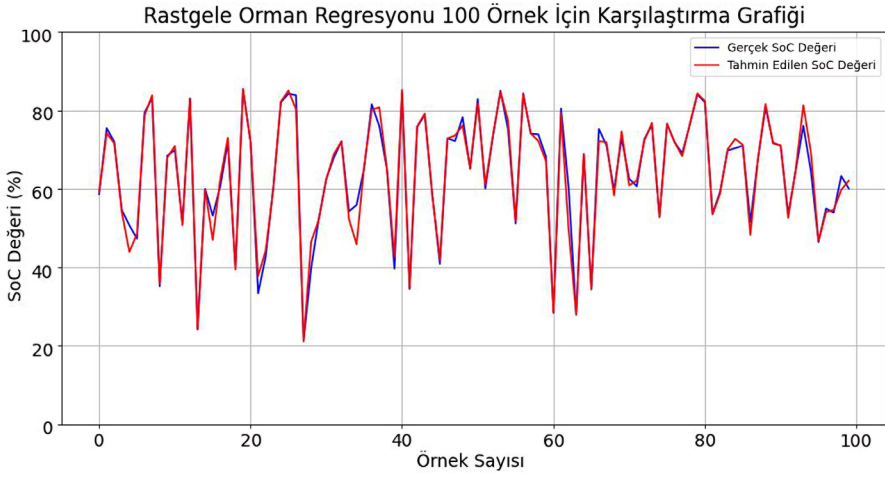
Tablo 2. Tüm Normalizasyon ve Öğrenme Algoritmalarına Ait Sonuçlar

		Normalizasyon Olmadan	Medyan	MinMax	DMinMax	Sigmoid
		0.9737	0.9776	0.9782	0.9782	0.9706
KNN	MAE	1.3981	1.2325	1.2029	1.2029	1.5178
	RMSE	2.4257	2.2366	2.2074	2.2074	2.5677
		0.9769	0.9769	0.9769	0.9769	0.9769
RF	MAE	1.2281	1.229	1.229	1.2287	1.2291
	RMSE	2.2738	2.2753	2.2735	2.2751	2.2752
		0.9622	0.9623	0.9623	0.9623	0.9627
XGB	MAE	1.988	1.9894	1.9894	1.9894	1.9757
	RMSE	2.9086	2.9067	2.9067	2.9067	2.8915
		0.9456	0.9551	0.9429	0.93381	0.9073
GRU	MAE	2.525	2.2361	2.5205	2.7262	3.4379
	RMSE	3.4905	3.1703	3.5755	3.853	4.5583
		0.9357	0.9567	0.9417	0.9418	0.9244
LSTM	MAE	2.8257	2.2091	2.6015	2.5756	2.9745
	RMSE	3.7962	3.116	3.6155	3.6127	4.1174
		0.9427	0.9603	0.94	0.9238	0.9196
BiLSTM	MAE	2.6036	2.0959	2.6844	3.1481	3.1623
	RMSE	3.5826	2.9838	3.667	4.1319	4.245
		0.9132	0.9156	0.884	0.8611	0.9205
CNN	MAE	3.1575	3.1035	3.4532	3.5677	3.1132
	RMSE	4.4104	4.3498	5.099	5.581	4.2226
		0.8851	0.8851	0.8851	0.8851	0.8841
LR	MAE	3.7238	3.7238	3.7238	3.7238	3.7731
	RMSE	5.0760	5.0760	5.0760	5.0760	5.0965



Şekil 2. KNN Karşılaştırma Grafiği

Şekil 2’de, D-MinMax normalizasyonu ile normalize edilmiş veri kümesinden rastgele seçilen 100 örneğe ait gerçek SoC değerleri ile KNN algoritmasının tahmin ettiği SoC değerleri karşılaştırılmaktadır. Grafikte, tahmin edilen değerlerin gerçek değerlere büyük ölçüde yakınsadığı görülmektedir. KNN algoritması, komşuluk tabanlı yaklaşımı sayesinde veri dağılımını iyi yakalayarak düşük hata oranı ile başarılı tahminler yapabilmektedir. Özellikle veri kümesinin büyük olması ve D-MinMax normalizasyonunun etkisiyle, KNN modeli SoC tahmininde yüksek doğruluk sergilemiştir. Ancak, bazı bölgelerde küçük sapmalar olduğu gözlemlenmektedir, bu da KNN’nin belirli örneklerde yerel değişimlere karşı daha hassas olabileceğini göstermektedir.



Şekil 3. RF Karşılaştırma Grafiği

Şekil 3'te, Min-Max normalizasyonu ile işlenmiş veri kümesinden rastgele seçilen 100 örnek için gerçek SoC değerleri ile RF algoritmasının tahmin ettiği SoC değerleri karşılaştırılmaktadır. RF algoritmasının tahminleri, genel olarak gerçek değerlere oldukça yakın olup, modelin veriye iyi uyum sağladığını göstermektedir. Min-Max normalizasyonunun etkisiyle, modelin ölçeklenmiş veriler üzerinde daha dengeli bir tahmin performansı sergilediği gözlemlenmiştir. RF modeli, özellikle çok sayıda karar ağacından oluşan yapısı sayesinde gürültüye karşı daha dayanıklı olup, daha istikrarlı tahminler üretmiştir. Ancak, bazı örneklerde küçük farklılıklar gözlenmiş olup, bu farklılıkların modelin genelleme yeteneği ve kullanılan hiperparametreler ile ilişkilendirilebileceği değerlendirilmektedir.

Model performansı açısından değerlendirildiğinde, düşük SoC seviyelerinde örnek sayısının az olması, modellerin bu aralıktaki verileri öğrenme kapasitesini sınırlandırabilmektedir. Makine öğrenmesi ve derin öğrenme algoritmaları, veri yoğunluğu fazla olan bölgelerde genellikle daha iyi performans gösterirken, nadir görülen örneklerde tahmin hatalarının artması mümkündür. Bu bağlamda, düşük SoC seviyelerinde model performansında bir miktar düşüş yaşanmaktadır. Şekil 2 ve Şekil 3'te verilen KNN ve RF algoritmalarına ait tahmin grafikleri incelendiğinde, genel olarak tahmin edilen SoC değerlerinin gerçek değerlere oldukça yakın olduğu görülmektedir. Ancak, özellikle düşük SoC seviyelerinde küçük sapmalar meydana gelmektedir. Bu durum, düşük SoC seviyelerindeki örneklerin azlığı nedeniyle modelin bu bölgedeki dağılımı tam olarak öğrenememiş olabileceğini göstermektedir. Özellikle Şekil 2'de KNN algoritmasının düşük SoC seviyelerinde nispeten daha fazla sapma gösterdiği gözlemlenebilirken, Şekil 3'te RF algoritması daha istikrarlı bir performans sergilemektedir. Bu da RF algo-

ritmasının, ağaç tabanlı yapısı sayesinde düşük örneklem bölgelerinde bile daha iyi genelleme yapabildiğini göstermektedir.

5. Genel Değerlendirme ve Gelecek Çalışmalar

Yapılan bu çalışmada kullanılan farklı makine öğrenmesi ve derin öğrenme algoritmalarının performansları çeşitli normalizasyon teknikleri uygulanarak aynı veri kümesi üzerinde değerlendirilmiştir. Algoritmalar, MAE, RMSE ve hata metriklerine göre test edilmiş ve istenilen başarımlar düzeyine ulaşılmıştır. Makine öğrenimi tekniği olarak; LR, XGB, RF regresyonu ve KNN algoritmaları kullanılmıştır. Derin öğrenme teknikleri olarak; LSTM, BiLSTM, GRU ve CNN yöntemleri üzerinde çalışılmıştır. En başarılı sonuçlar KNN algoritması ve RF regresyonu yöntemlerinden elde edilmiştir. Normalizasyon tekniklerinin, regresyon modellerinin oluşturulmasında belirli bir ölçüde katkı sağlayabildiği gösterilmiştir.

Gelecek çalışmalar kapsamında, hiperparametre optimizasyonunun derin öğrenme modelleri üzerindeki etkisi daha detaylı incelenerek, model mimarilerinin iyileştirilmesi hedeflenmektedir. Özellikle, LSTM ve GRU gibi zaman serisi tahmin modelleri için daha derin ve karmaşık ağ yapıları oluşturularak model doğruluğunun artırılması planlanmaktadır. Bunun yanı sıra, SoC tahmininin doğruluğunu artırmak için ek özellik mühendisliği teknikleri uygulanarak yeni giriş değişkenleri türetilebilir. Örneğin, batarya yaşlanma etkileri, sürüş alışkanlıkları ve çevresel faktörler gibi ek parametrelerin modele dâhil edilmesi, tahmin doğruluğunu artırabilir. Ayrıca, model entegrasyonu ve hibrit yaklaşımlar ile farklı algoritmaların birlikte kullanılması hedeflenmektedir. Örneğin, makine öğrenmesi ve derin öğrenme modellerinin birleşimiyle oluşturulabilecek hibrit modellerin performansı değerlendirilebilir. Hibrit yöntemler sayesinde, geleneksel modellerin genelleme yeteneği ile derin öğrenme modellerinin güçlü temsil yetenekleri birleştirilerek daha yüksek doğruluk elde edilebilir. Bununla birlikte, farklı elektrikli araçlardan elde edilen daha geniş ve çeşitli veri kümeleri kullanılarak modellerin genelleme yetenekleri artırılabilir. Bu sayede, modelin farklı araç tipleri ve batarya teknolojileri için geçerliliği test edilebilir. Son olarak, çevrimiçi (online) SoC tahmin sistemleri üzerinde çalışarak, gerçek zamanlı tahmin yapabilen ve batarya yönetim sistemlerine entegre edilebilecek modeller geliştirmek gelecekteki araştırmalar için önemli bir yön olacaktır.

Teşekkür

BMW'nin açık kaynaklı veri kümesi bu çalışmada kullanılmıştır. Veri kümesini bilim insanlarının kullanımına açtığı için BMW'ye teşekkür ederiz.

Fon/Finansman bilgileri

Bu çalışma için herhangi bir kurum ve/veya kuruluştan destek alınmamıştır.

Etik Kurul Onayı ve İzinler

Çalışma, etik kurul izni veya herhangi bir özel izin gerektirmemektedir.

Çıkar çatışmaları/Çatışan çıkarlar

Yazarlar arasında herhangi bir çıkar çatışması yoktur.

Yazarların Katkısı

Yazarlar çalışmaya eşit oranda katkı sağlamıştır. Yazarlar makalenin son halini okumuş ve onaylamıştır.

Kaynakça

- Chemali, E., Kollmeyer, P. J., Preindl, M., & Emadi, A. (2018). State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *Journal of Power Sources*, 400, 242–255. Doi: <https://doi.org/10.1016/j.jpowsour.2018.06.104>
- Chandran, V., Patil, C. K., Karthick, A., Ganeshaperumal, D., Rahim, R., & Ghosh, A. (2021). State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electric Vehicle Journal*, 12(1), 38. Doi: <https://doi.org/10.3390/wevj12010038>
- Çavuş, E., & Sancaktar, İ. (2022). Batarya sağlığı durumunun makine öğrenmesi ile kestirimi. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 11(3), 601–610. <https://doi.org/10.28948/ngumuh.1112985>
- Diouf, B., & Pode, R. (2015). Potential of lithium-ion batteries in renewable energy. *Renewable Energy*, 76, 375–380. Doi: <https://doi.org/10.1016/j.renene.2014.11.058>
- Eissa, M. A., & Chen, P. (2023). Machine learning-based electric vehicle battery state of charge prediction and driving range estimation for rural applications. *IFAC-PapersOnLine*, 56(3), 355–360. Doi: <https://doi.org/10.1016/j.ifacol.2023.12.050>
- El Fallah, S., Kharbach, J., Hammouch, Z., Rezzouk, A., & Jamil, M. O. (2023). State of charge estimation of an electric vehicle's battery using deep neural networks: Simulation and experimental results. *Journal of Energy Storage*, 62, 106904. Doi: <https://doi.org/10.1016/j.est.2023.106904>
- Sevim, G., & Altın, N. (2023). Lityum bataryalarda şarj durumu (SoC) ve sağlık durumu (SoH) kestirimi. *Karamanoğlu Mehmetbey Üniversitesi Mühendislik ve Doğa Bilimleri Dergisi*, 5(1), 33–51. Doi: <https://doi.org/10.55213/kmu-jens.1250621>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. Doi: <https://doi.org/10.1162/neco.1997.9.8.1735>

- Keskin, B., Gunal, E. Ş., Urazel, B., & Keskin, K. (2022). Elektrikli araç bataryalarının şarj durumu tahmini için bir model. *Nicel Bilimler Dergisi*, 4(2), 161–175. Doi: <https://doi.org/10.51541/nicel.1117756>
- Kelle, A. C., & Yüce, H. (2022). MQTT trafiğinde DoS saldırılarının makine öğrenmesi ile sınıflandırılması ve modelin SHAP ile yorumlanması. *Journal of Materials and Mechatronics: A*, 3(1), 50–62. Doi: <https://doi.org/10.55546/jmm.995091>
- Kılıç, S. (2013). Doğrusal regresyon analizi. *Journal of Mood Disorders*, 3(2), 90–92.
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999–7019. Doi: <https://doi.org/10.1109/TNNLS.2021.3084827>
- Lin, X. (2018). Theoretical analysis of battery SOC estimation errors under sensor bias and variance. *IEEE Transactions on Industrial Electronics*, 65(9), 7138–7148. Doi: <https://doi.org/10.1109/TIE.2018.2795521>
- Meng, J., Ricco, M., Luo, G., Swierczynski, M., Stroe, D.-I., Stroe, A.-I., & Teodorescu, R. (2017). An overview and comparison of online implementable SOC estimation methods for lithium-ion battery. *IEEE Transactions on Industry Applications*, 54(2), 1583–1591. Doi: <https://doi.org/10.1109/TIA.2017.2775179>
- Nitta, N., Wu, F., Lee, J. T., & Yushin, G. (2015). Li-ion battery materials: Present and future. *Materials Today*, 18(5), 252–264. Doi: <https://doi.org/10.1016/j.mattod.2014.10.040>
- Oji, T., Zhou, Y., Ci, S., Kang, F., Chen, X., & Liu, X. (2021). Data-driven methods for battery SOH estimation: Survey and a critical analysis. *IEEE Access*, 9, 126903–126916. Doi: <https://doi.org/10.1109/ACCESS.2021.3111927>
- Rivera-Barrera, J. P., Muñoz-Galeano, N., & Sarmiento-Maldonado, H. O. (2017). SoC estimation for lithium-ion batteries: Review and future challenges. *Electronics*, 6(4), 102. Doi: <https://doi.org/10.3390/electronics6040102>
- Serinbaş, M. M., & Gülbahçe, M. O. (2022). Batarya şarj durumu kestirim yöntemleri üzerine inceleme. In *Güç Sistemleri Konferansı* (pp. 1–9). Cıgır Türkiye.
- Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*.

- Steinstraeter, M., Buberger, J., & Trifonov, D. (2020). Battery and heating data in real driving cycles. IEEE Dataport. Doi: <https://doi.org/10.21227/6jr9-5235>
- Şahin, E. M., Sahin, S., & Tanağardıgil, İ. (2021). Battery State of Health and Charge Estimation Using Machine Learning Methods. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 389-394. <https://doi.org/10.31590/ejosat.959630>
- Taş, G., Uysal, A., & Bal, C. (2023). A new lithium polymer battery dataset with different discharge levels: SOC estimation of lithium polymer batteries with different convolutional neural network models. *Arabian Journal for Science and Engineering*, 48(5), 6873–6888. Doi: <https://doi.org/10.1007/s13369-022-07586-8>
- Thomas, R., Despesse, G., Bacquet, S., Fernandez, E., Lopez, Y., Ramahefa-Andry, P., & Cassarino, L. (2021). A high frequency self-reconfigurable battery for arbitrary waveform generation. *World Electric Vehicle Journal*, 12(1), 8. Doi: <https://doi.org/10.3390/wevj12010008>
- Tunç, T., & Erdem, H. (2024). Lityum iyon batarya için dikkat mekanizmalı çeşitli tekrarlayan birim (GRU) sinir ağını temel alan sağlık durumu (SOH) tahmini. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 13(2), 639–647. Doi: <https://doi.org/10.28948/ngumuh.1407480>
- Vidal, C., Malysz, P., Kollmeyer, P., & Emadi, A. (2020). Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art. *IEEE Access*, 8, 52796–52814. Doi: <https://doi.org/10.1109/ACCESS.2020.2980961>
- Winter, M., & Brodd, R. J. (2004). What are batteries, fuel cells, and supercapacitors? *Chemical Reviews*, 104(10), 4245–4270. Doi: <https://doi.org/10.1021/cr020730k>
- Xie, Q., Suvarna, M., Li, J., Zhu, X., Cai, J., & Wang, X. (2021). Online prediction of mechanical properties of hot rolled steel plate using machine learning. *Materials & Design*, 197, 109201. Doi: <https://doi.org/10.1016/j.matdes.2020.109201>
- Yavuz, S., & Deveci, M. (2012). İstatiksel normalizasyon tekniklerinin yapay sinir ağının performansına etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 40, 167–187.
- Zahid, T., Xu, K., Li, W., Li, C., & Li, H. (2018). State of charge estimation for electric vehicle power battery using advanced machine learning algorithm under diversified drive cycles. *Energy*, 162, 871–882. Doi: <https://doi.org/10.1016/j.energy.2018.08.071>