

Perbandingan kinerja algoritma Decision Tree dan XGBoost dalam Prediksi Waktu pengisian baterai Lithium Polymer

Dio Lizwardi^{1*)}, Sabilal Rasyad², Tegar Prasetyo³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Jalan Srijaya Negara, Bukit Besar, Kota Palembang, Indonesia

*Penulis Korespondensi, e-mail: dlizwardi@gmail.com

Received:15/06/2026

Revised:20/07/2026

Accepted:20/07/2026

ABSTRAK

Baterai Lithium Polymer (Li-Po) banyak digunakan pada berbagai perangkat elektronik dan kendaraan listrik karena memiliki massa yang ringan, kepadatan energi yang tinggi, serta kemampuan pelepasan daya yang baik. Seiring meningkatnya kebutuhan sistem pengisian cepat (fast charging), informasi mengenai waktu yang dibutuhkan hingga baterai terisi penuh atau Time to Full Charge (TTF) menjadi penting untuk meningkatkan efisiensi pengisian dan kenyamanan pengguna. Penelitian ini bertujuan membandingkan kinerja algoritma Decision Tree dan XGBoost dalam memprediksi TTF baterai Li-Po menggunakan data hasil monitoring berbasis ESP32. Data dikumpulkan selama proses pengisian baterai dengan merekam parameter tegangan, arus, suhu, dan State of Charge (SoC), kemudian digunakan sebagai masukan model machine learning. Evaluasi dilakukan menggunakan Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), dan koefisien determinasi (R^2). Hasil pengujian menunjukkan bahwa Decision Tree menghasilkan MAE sebesar 44,5892 detik, RMSE sebesar 62,08 detik, MAPE sebesar 10,82%, dan R^2 sebesar 0,9905. Sementara itu, XGBoost menghasilkan MAE sebesar 47,1935 detik, RMSE sebesar 63,657 detik, MAPE sebesar 7,796%, dan R^2 sebesar 0,9906. Hasil tersebut menunjukkan bahwa kedua algoritma memiliki performa prediksi yang sangat baik dengan nilai R^2 di atas 0,99, sedangkan XGBoost memberikan tingkat kesalahan relatif yang lebih rendah berdasarkan nilai MAPE.

Kata Kunci: Decision Tree, Lithium Polymer, Machine Learning, State of Charge, Time to Full Charge (TTF), XGBoost.

ABSTRACT

Lithium Polymer (Li-Po) batteries are widely used in portable electronic devices and electric vehicles due to their lightweight characteristics, high energy density, and excellent power delivery capability. As fast-charging technology becomes increasingly important, accurate prediction of the remaining charging time or Time to Full Charge (TTF) is essential to improve charging efficiency and user convenience. This study compares the performance of Decision Tree and XGBoost algorithms in predicting the TTF of Li-Po batteries using data collected from an ESP32-based monitoring system. Voltage, current, temperature, and State of Charge (SoC) were continuously recorded during the charging process and used as input features for the machine learning models. The models were evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). Experimental results indicate that the Decision Tree model achieved an MAE of 44.5892 seconds, RMSE of 62.08 seconds, MAPE of 10.82%, and R^2 of 0.9905. Meanwhile, XGBoost obtained an MAE of 47.1935 seconds, RMSE of 63.657 seconds, MAPE of 7.796%, and R^2 of 0.9906. These findings demonstrate that both algorithms provide highly accurate TTF prediction, while XGBoost achieves lower relative prediction error according to the MAPE metric.

Keywords: Decision Tree, Lithium Polymer, Machine Learning, State of Charge, Time to Full Charge, (TTF) XGBoost.

1. PENDAHULUAN

p-ISSN: 2356-0533; e-ISSN: 2355-9195



Baterai Lithium Polymer (LiPo) telah menjadi salah satu teknologi penyimpanan energi yang banyak digunakan pada berbagai aplikasi modern seperti unmanned aerial vehicle (UAV), kendaraan listrik, robot otonom, dan perangkat elektronik portabel karena memiliki kepadatan energi yang tinggi, bobot yang ringan, serta kemampuan pelepasan daya yang baik [1]. Seiring meningkatnya penggunaan perangkat berbasis baterai, kebutuhan terhadap sistem pengisian daya yang efisien serta mampu memberikan informasi estimasi waktu pengisian secara akurat juga semakin meningkat guna meningkatkan kenyamanan pengguna dan efisiensi penggunaan energi.[2].

Pada proses pengisian baterai, estimasi waktu hingga baterai terisi penuh (Time to Full Charge atau TTF) yang menyatakan sisa waktu aktual yang diperlukan hingga baterai mencapai kondisi penuh, menjadi informasi penting yang dapat membantu pengguna dalam perencanaan operasional perangkat serta mendukung pengelolaan energi yang lebih efektif. Namun, karakteristik pengisian baterai LiPo bersifat nonlinier dan dipengaruhi oleh berbagai parameter seperti State of Charge (SoC), tegangan, arus pengisian, temperatur, dan kondisi kesehatan baterai. Kompleksitas hubungan antarparameter tersebut menyebabkan metode konvensional sering mengalami keterbatasan dalam menghasilkan estimasi yang akurat pada berbagai kondisi operasi [2], [3].

Perkembangan machine learning telah membuka peluang baru dalam pemodelan karakteristik baterai yang kompleks tanpa memerlukan formulasi matematis yang rumit. Berbagai penelitian menunjukkan bahwa algoritma machine learning mampu mempelajari pola hubungan antara parameter operasional baterai dan kondisi aktual baterai secara efektif, bahkan ketika jumlah data yang tersedia relatif terbatas [4]. Pendekatan berbasis data juga dinilai lebih adaptif terhadap perubahan karakteristik baterai dibandingkan metode berbasis model fisik konvensional karena mampu menangani hubungan nonlinier yang kompleks serta mengurangi ketergantungan terhadap proses identifikasi parameter fisik yang rumit [5].

Di antara berbagai algoritma machine learning, Decision Tree dan eXtreme Gradient Boosting (XGBoost) merupakan dua metode yang banyak digunakan pada data tabular karena memiliki kemampuan menangani hubungan nonlinier dengan baik. Decision Tree memiliki struktur yang sederhana, mudah diinterpretasikan, dan membutuhkan sumber daya komputasi yang relatif rendah sehingga memiliki struktur yang sederhana, mudah diinterpretasikan, serta membutuhkan kompleksitas komputasi yang relatif rendah sehingga berpotensi diterapkan pada perangkat dengan sumber daya terbatas [6]. Sementara itu, XGBoost merupakan pengembangan metode gradient boosting berbasis pohon keputusan yang dilengkapi mekanisme regularisasi sehingga mampu meningkatkan akurasi prediksi dan mengurangi risiko overfitting [7] [8].

Penelitian terdahulu umumnya berfokus pada estimasi SoC, SoH, maupun prediksi umur pakai baterai menggunakan berbagai pendekatan machine learning. Penelitian yang dilakukan oleh Jafari dan Byun (2024) berhasil menunjukkan bahwa algoritma berbasis pohon keputusan mampu memberikan estimasi SoC yang akurat pada baterai kendaraan Listrik [9]. Ren dkk (2024). melaporkan bahwa kombinasi XGBoost dan Random Forest menghasilkan performa yang baik dalam prediksi SoC kendaraan listrik [10]. Selain itu, berbagai kajian literatur terbaru menunjukkan bahwa algoritma berbasis pohon masih menjadi pendekatan yang kompetitif dibandingkan model deep learning, khususnya ketika jumlah data yang tersedia terbatas dan kebutuhan komputasi menjadi pertimbangan utama [3]. Meskipun demikian, sebagian besar penelitian tersebut masih berfokus pada estimasi kondisi baterai, sedangkan kajian yang secara khusus membandingkan kinerja Decision Tree dan XGBoost dalam memprediksi waktu pengisian penuh (Time to Full Charge) baterai Lithium Polymer masih terbatas. Selain itu, sebagian besar penelitian hanya berfokus pada estimasi kondisi baterai tanpa mengevaluasi kemampuan model dalam memprediksi durasi pengisian aktual menggunakan data hasil proses pengisian secara langsung.

Berdasarkan kondisi tersebut, terdapat gap penelitian yang perlu dikaji lebih lanjut, yaitu masih terbatasnya evaluasi komprehensif mengenai perbandingan performa algoritma Decision Tree dan XGBoost dalam prediksi waktu pengisian penuh baterai LiPo berdasarkan data pengisian aktual. Padahal, informasi TTF memiliki peranan penting dalam meningkatkan efisiensi sistem pengisian, mendukung implementasi smart charging, dan meningkatkan kualitas layanan pada sistem berbasis baterai. Selain itu, perbandingan kedua



algoritma diperlukan untuk mengetahui apakah peningkatan kompleksitas model pada XGBoost memberikan keuntungan yang signifikan dibandingkan Decision Tree yang lebih sederhana dari sisi komputasi.

Oleh karena itu, penelitian ini bertujuan membandingkan kinerja algoritma Decision Tree dan XGBoost dalam memprediksi waktu pengisian penuh (Time to Full Charge) baterai Lithium Polymer menggunakan data hasil proses pengisian baterai. Kinerja kedua model dievaluasi menggunakan metrik Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), dan koefisien determinasi (R^2). Hasil penelitian diharapkan dapat memberikan rekomendasi pemilihan algoritma yang paling sesuai untuk implementasi sistem prediksi waktu pengisian baterai berbasis machine learning yang akurat, efisien, dan dapat diterapkan pada sistem manajemen baterai modern.

2. METODE PENELITIAN

Penelitian ini dilakukan dalam beberapa tahapan yang meliputi studi literatur, Desain Sistem, Implementasi alat, pengujian alat, dan evaluasi sistem. Tahapan penelitian dirancang untuk memperoleh data pengisian baterai Lithium Polymer yang selanjutnya digunakan dalam proses prediksi waktu pengisian penuh menggunakan algoritma Decision Tree dan XGBoost. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Diagram Metode penelitian

2.1 Studi Literatur

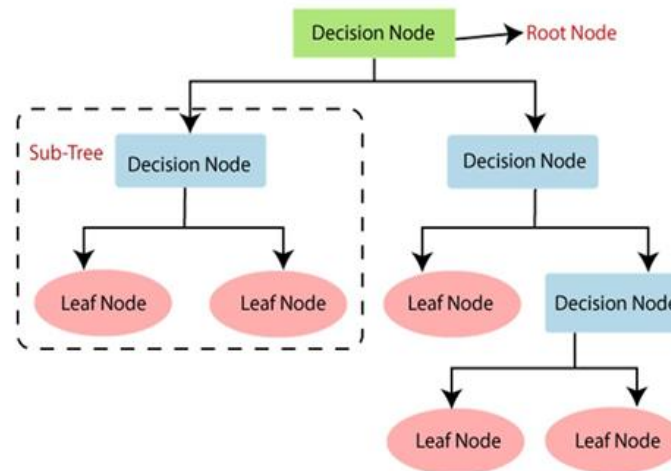
a. Karakteristik baterai Lithium Polymer dan Waktu pengisian penuh

Baterai Lithium Polymer (LiPo) adalah salah satu jenis baterai isi ulang yang paling banyak digunakan pada berbagai perangkat elektronik, kendaraan listrik, dan UAV karena memiliki kepadatan energi yang tinggi, bobot yang ringan, dan kemampuan pelepasan daya yang baik. Nilai State of Charge (SoC) baterai berubah selama proses pengisian, yang menunjukkan tingkat kapasitas energi yang tersisa. Time to Full Charge (TTF), atau perkiraan waktu yang diperlukan baterai untuk mencapai kondisi terisi penuh, adalah parameter penting dalam proses pengisian. Informasi TTF sangat penting untuk mendukung pengelolaan energi dan meningkatkan efisiensi penggunaan baterai pada berbagai aplikasi [2].

b. Algoritma Decision Tree

Algoritma pembelajaran mesin bernama Decision Tree membentuk model prediksi dalam struktur pohon yang terdiri dari root node, decision node, dan leaf node. Proses pembentukan pohon dilakukan dengan membagi data sesuai dengan atribut yang memiliki kemampuan terbaik untuk memisahkan data, sehingga menghasilkan kelompok data yang lebih homogen. [6]. Struktur umum Decision Tree ditunjukkan pada Gambar 2. Keunggulan algoritma ini adalah mudah diinterpretasikan, memiliki kompleksitas komputasi yang relatif rendah, serta mampu menangani hubungan nonlinier antarvariabel.





Gambar 2. Arsitektur Decision Tree

Nilai Gain Informasi digunakan untuk memilih atribut terbaik pada Decision Tree. Sebelum menghitung Gain Informasi, nilai entropy dihitung, yang digunakan untuk mengukur tingkat ketidakpastian atau ketidakhomogenan data. [11]. Nilai Entropy dapat dihitung menggunakan Persamaan (1)

$$Entropy(S) = - \sum_{i=1}^n p_i \log_2 (p_i) \quad (1)$$

Keterangan:

- Entropy(S): nilai entropy dari dataset S
- p_i : probabilitas kelas ke-i
- n: jumlah kelas pada dataset

Setelah nilai Entropy diperoleh, selanjutnya dihitung nilai Information Gain untuk menentukan atribut yang paling baik digunakan sebagai percabangan pada pohon keputusan. Atribut dengan nilai Information Gain terbesar akan dipilih sebagai node percabangan karena mampu menghasilkan pemisahan data yang lebih homogen. Persamaan Information Gain ditunjukkan pada Persamaan

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

Pada persamaan 2, IG merupakan nilai Information Gain dari suatu atribut, sedangkan $E(S)$ adalah nilai entropy dari dataset awal. Simbol S_v menyatakan subset data yang dihasilkan dari pembagian atribut, n_v menunjukkan jumlah data pada masing-masing subset, dan N merupakan jumlah seluruh data pada dataset[12] .

c. Algoritma Xgboost

XGBoost (Extreme Gradient Boosting) adalah algoritma ensemble learning berbasis gradient boosting decision tree yang secara bertahap membangun model prediksi dengan menggunakan pembentukan pohon keputusan untuk mengurangi kesalahan prediksi dan meningkatkan kinerja model. Tujuan XGBoost adalah untuk mengurangi fungsi objektif, yang terdiri dari fungsi kehilangan dan regularisasi, untuk menghindari overfitting:



$$obj = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (3)$$

Loss function mengukur kesalahan prediksi antara nilai aktual dan nilai prediksi, sedangkan fungsi regularisasi mengendalikan kekompleksan model [13] [14]. Fungsi regularisasi didefinisikan sebagai berikut:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (4)$$

di mana T adalah jumlah leaf node dan w_j adalah bobot pada leaf ke- j . Mekanisme boosting memperbaiki prediksi secara iteratif dengan menambahkan hasil prediksi pohon baru ke prediksi sebelumnya:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (5)$$

d. Evaluasi Matrik

Untuk menilai kinerja model prediksi yang dibangun, digunakan empat metrik evaluasi sebagai acuan pengukuran tingkat akurasi model, yaitu RMSE, MAE, MAPE, dan R^2 .

1. Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (6)$$

digunakan untuk mengukur tingkat kesalahan antara nilai aktual dan nilai prediksi, dengan memprioritaskan kesalahan yang besar.

2. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (7)$$

menghitung selisih absolut rata-rata antara nilai aktual dan nilai prediksi, seperti yang ditunjukkan di bawah ini

3. Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (8)$$

mengukur rata-rata kesalahan prediksi dalam bentuk persentase

4. Koefisien determinan R^2

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (9)$$

digunakan untuk mengevaluasi kemampuan model untuk menjelaskan variasi dalam data nyata [15] [14].

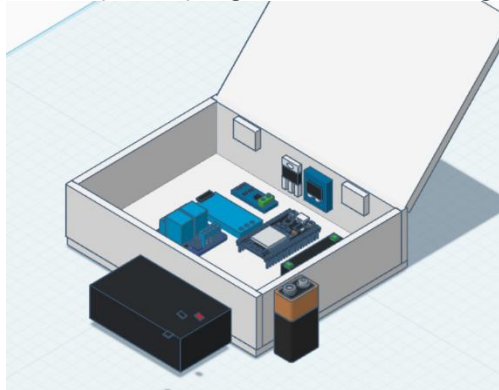
2.2 Desain sistem

Perancangan sistem dilakukan untuk mendukung proses monitoring dan prediksi waktu pengisian baterai Lithium Polymer (Li-Po). Sistem terdiri dari perangkat keras dan perangkat lunak yang terintegrasi untuk mengakuisisi data pengisian baterai serta menghasilkan estimasi Time to Full Charge (TTF) menggunakan algoritma Decision Tree dan XGBoost. Perancangan sistem meliputi perancangan mekanik dan perancangan elektronik.

a. Perancangan Mekanik



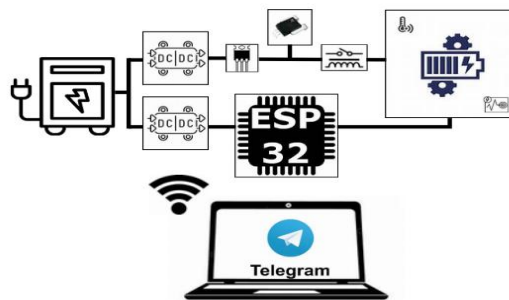
Perancangan mekanik dilakukan untuk menyusun seluruh komponen sistem dalam satu wadah yang aman dan terorganisir. Komponen utama yang digunakan meliputi baterai Li-Po, mikrokontroler ESP32, sensor, modul pendukung, dan catu daya. Desain mekanik dibuat untuk memudahkan proses instalasi, pengujian, serta pemeliharaan sistem selama proses pengambilan data berlangsung.



Gambar 3. Desain mekanik

b. Perancangan Elektronik

Perancangan elektronik dilakukan untuk menentukan hubungan antar komponen dalam sistem monitoring baterai. ESP32 berfungsi sebagai pusat pengendali untuk sistem, yang terhubung ke sensor tegangan, arus, dan suhu serta modul komunikasi. Data hasil pengukuran dikirim dan disimpan untuk digunakan dalam proses prediksi waktu pengisian baterai. Diagram rangkaian sistem ditunjukkan pada Gambar 4



Gambar 4 Desain Elektronik

2.3 Akuisisi Data

Data penelitian diperoleh melalui proses pengisian baterai Lithium Polymer (Li-Po) 3S berkapasitas 4000 mAh menggunakan sistem monitoring berbasis ESP32. Sistem dirancang untuk mengakuisisi data secara periodik dengan interval satu detik dengan membaca beberapa parameter utama selama proses pengisian, yaitu tegangan baterai, arus pengisian, suhu baterai, dan *State of Charge* (SoC). Data dari setiap sensor diproses oleh ESP32 dan direkam secara otomatis ke dalam file .csv dengan interval pengambilan data setiap 1 detik hingga baterai mencapai kondisi penuh.

Selama proses akuisisi data, baterai diisi menggunakan metode *constant current* (CC) dengan arus pengisian sekitar 2 A. Proses pengambilan data dimulai ketika tegangan baterai berada pada 10,30 V dan berakhir ketika tegangan mencapai 12,20 V atau baterai berada pada kondisi penuh. Data hasil akuisisi selanjutnya digunakan sebagai dataset dalam proses pelatihan dan pengujian model prediksi *Time to Full Charge* (TTF).

2.4 Dataset Penelitian

Dataset yang digunakan pada penelitian ini terdiri atas 4.027 sampel data yang diperoleh dari satu siklus pengisian penuh baterai Lithium Polymer. Setiap sampel merepresentasikan kondisi baterai pada waktu tertentu selama proses pengisian dengan parameter masukan berupa tegangan baterai, arus pengisian, suhu



baterai, dan State of Charge (SoC). Adapun parameter target yang digunakan dalam penelitian ini adalah Time to Full Charge (TTF).

TABEL I: Karakteristik Dataset Penelitian

Parameter	Spesifikasi
Jenis baterai	Lithium Polymer (Li-Po) 3S
Kapasitas baterai	4000 mAh
Tegangan nominal	11,4 V
Jumlah sampel data	4.027
Jumlah siklus pengisian	1 siklus
Interval sampling	1 detik
Rentang tegangan	10,30–12,20 V
Arus pengisian	± 2 A
Rentang suhu	30–32 °C
Rentang SoC	0–100 %

Berdasarkan karakteristik pada Tabel I, dataset yang diperoleh terdiri atas 4.027 sampel data hasil proses pengisian baterai Lithium Polymer. Dataset tersebut selanjutnya digunakan sebagai data masukan pada proses pelatihan dan pengujian model Decision Tree dan XGBoost untuk memprediksi Time to Full Charge (TTF).

2.5 Penentuan Target Time to Full Charge (TTF)

Variabel target pada penelitian ini adalah *Time to Full Charge* (TTF), yaitu waktu yang masih diperlukan hingga baterai mencapai kondisi penuh (100% SoC) pada setiap kondisi pengukuran. Nilai TTF diperoleh berdasarkan selisih antara waktu pengambilan data dengan waktu ketika proses pengisian selesai, sehingga setiap sampel memiliki target berupa sisa waktu aktual menuju kondisi penuh, bukan hasil estimasi menggunakan persamaan matematis maupun konversi dari nilai SoC.

Pendekatan ini dipilih agar model *machine learning* dapat mempelajari hubungan antara parameter tegangan, arus, suhu, dan SoC terhadap sisa waktu pengisian yang sebenarnya. Dengan demikian, model yang dihasilkan diharapkan mampu memberikan prediksi TTF secara lebih akurat berdasarkan kondisi baterai saat proses pengisian berlangsung.

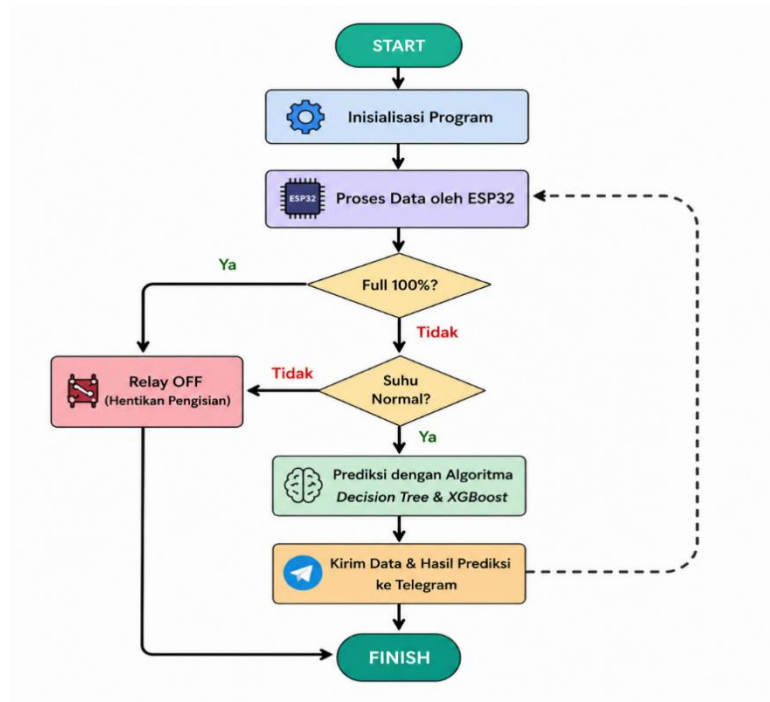
2.6 Implementasi Alat Dan Visualisasi

Mengintegrasikan perangkat lunak dan perangkat keras yang dimaksudkan untuk memantau dan memprediksi waktu pengisian baterai Lithium Polymer (Li-Po) adalah langkah pertama menuju implementasi sistem. ESP32 digunakan sebagai pengendali utama untuk membaca data sensor berupa tegangan, arus, suhu, dan State of Charge (SoC) selama proses pengisian berlangsung. Data yang diperoleh kemudian dikirim ke Python API untuk diproses menggunakan algoritma Decision Tree dan XGBoost sehingga menghasilkan estimasi Time to Full Charge (TTF). Hasil prediksi selanjutnya dikirim kembali ke sistem dan ditampilkan melalui Telegram untuk memudahkan pemantauan secara real-time. Selain itu, sistem dilengkapi mekanisme proteksi yang akan memutus proses pengisian apabila suhu baterai melebihi batas yang telah ditentukan.

a. Diagram Alir

Gambar 5 menunjukkan alur kerja sistem yang diawali dengan inisialisasi program dan pembacaan data sensor oleh ESP32. Data yang diperoleh digunakan untuk memantau kondisi pengisian baterai. Jika baterai belum mencapai kapasitas penuh dan suhu berada dalam kondisi normal, data diproses menggunakan algoritma Decision Tree dan XGBoost untuk menghasilkan prediksi Time to Full Charge (TTF). Hasil prediksi dikirim ke Telegram sebagai media pemantauan, sedangkan proses pengisian dihentikan secara otomatis ketika baterai telah mencapai kapasitas penuh.





Gambar 5. Diagram Alir

2.7 Pengujian Alat

Pengujian akuisisi data dilakukan untuk memastikan sistem mampu membaca dan menampilkan parameter pengisian baterai secara real-time. Parameter yang diamati meliputi tegangan, arus, suhu, State of Charge (SoC), dan estimasi waktu pengisian penuh (Time to Full Charge / TTF).

```

89 )
90
91 print("\n ✖ Menunggu ESP32 reboot...")
92 time.sleep(3)
93
94 print(" 📡 Membaca pesan awal ESP32...")
95 boot_start = time.time()
96
97 while time.time() - boot_start < 5:
98     line = self.read_line()
99
100     if line:
101         print(f"ESP32: {line}")
102
103         if "#READY" in line:
104             break
105
106
107 PS C:\Users\email\Downloads\Ambil Data Python> python main.py
108
109 WARNING_PZEM_DATA_TOO_SHORT
110 # 17 | ACS: 4.961 A | PZEM: 0.00 V | Obj: 33.51 °C | SoC: 0.0% | ETA Full: 0:39:54
111 # 18 | ACS: 4.963 A | PZEM: 10.22 V | Obj: 33.51 °C | SoC: 2.6% | ETA Full: 0:38:47
112 # 19 | ACS: 4.957 A | PZEM: 10.22 V | Obj: 33.47 °C | SoC: 2.6% | ETA Full: 0:38:50
113 # 20 | ACS: 4.962 A | PZEM: 10.22 V | Obj: 33.47 °C | SoC: 2.6% | ETA Full: 0:38:48
114 # 21 | ACS: 4.968 A | PZEM: 10.22 V | Obj: 33.39 °C | SoC: 2.8% | ETA Full: 0:38:45
115 # 22 | ACS: 4.963 A | PZEM: 10.22 V | Obj: 33.51 °C | SoC: 2.8% | ETA Full: 0:38:47
116 # 23 | ACS: 4.960 A | PZEM: 10.23 V | Obj: 33.43 °C | SoC: 2.9% | ETA Full: 0:38:46
  
```

Gambar 6. Pengujian Alat

Hasil pengujian menunjukkan bahwa data yang dikirim oleh ESP32 berhasil diterima dan ditampilkan dengan baik. Selanjutnya, data tersebut diproses menggunakan algoritma Decision Tree dan XGBoost pada lingkungan Python untuk menghasilkan prediksi Time to Full Charge (TTF). Hasil prediksi kemudian digunakan sebagai dasar evaluasi kinerja kedua algoritma.

2.8 Evaluasi

Dalam evaluasi ini, kinerja algoritma Decision Tree dan XGBoost diukur untuk memprediksi waktu pengisian penuh (Time to Full Charge / TTF) baterai Li-Po. Untuk melakukan evaluasi, metrik seperti Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), dan

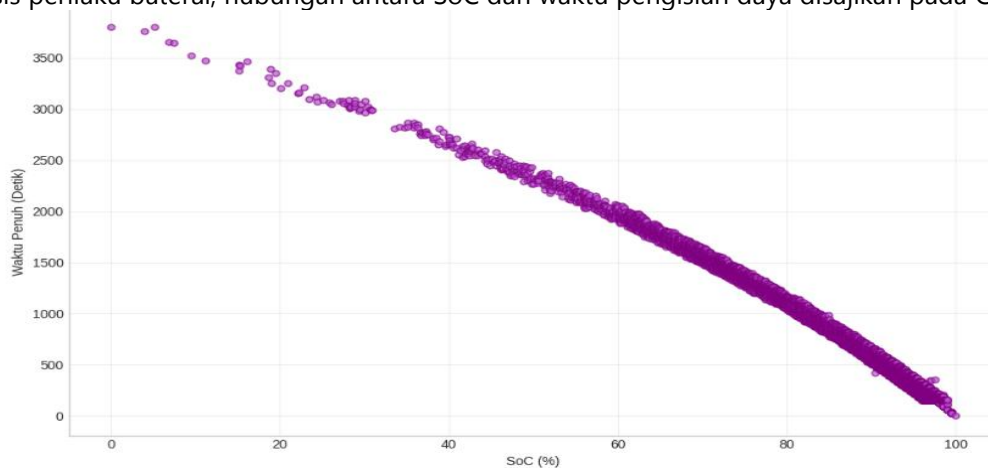
Dio Lizwardi: Perbandingan kinerja *Algoritma Decision*..... p-ISSN: 2356-0533; e-ISSN: 2355-9195



koefisien determinasi (R^2) digunakan. Metrik ini digunakan untuk mengevaluasi tingkat kesalahan prediksi model serta kemampuan model untuk menampilkan data aktual dengan benar. Hasil evaluasi digunakan sebagai dasar dalam menentukan algoritma yang memiliki performa terbaik untuk prediksi waktu pengisian baterai.

3. HASIL DAN PEMBAHASAN

Sistem berhasil mengumpulkan data terkait baterai selama proses pengisian daya. Besaran terukur yang tercatat mencakup tegangan, arus, suhu, dan Status Pengisian (SoC), yang digunakan sebagai variabel masukan untuk model prediksi. Waktu pengisian daya didefinisikan sebagai keluaran target. Untuk menganalisis perilaku baterai, hubungan antara SoC dan waktu pengisian daya disajikan pada Gambar 7.

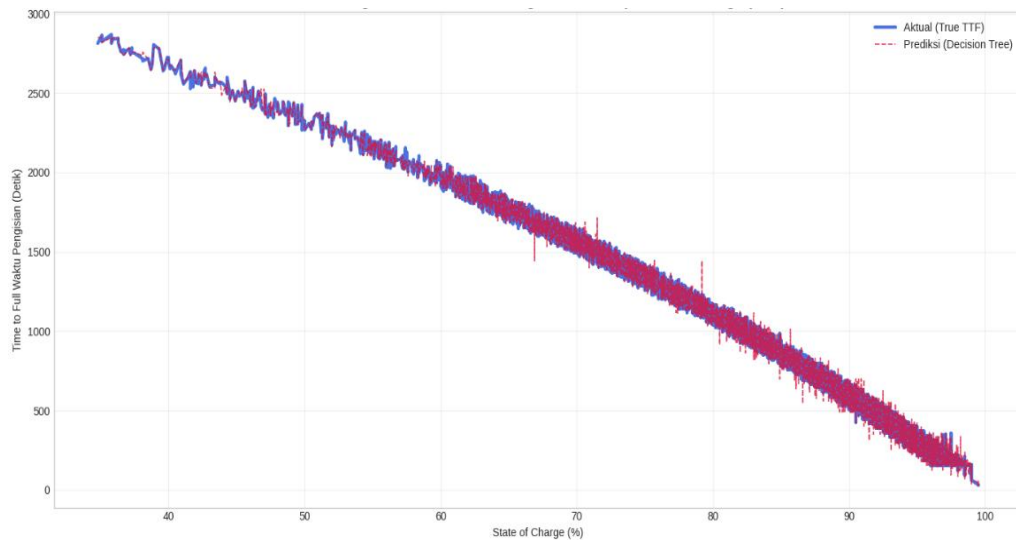


Gambar 7. Waktu Penuh Dengan SOC

Berdasarkan Gambar 7, terlihat bahwa waktu pengisian penuh menurun seiring dengan meningkatnya nilai SoC. Pola tersebut menunjukkan hubungan nonlinier antara SoC dan TTF, dimana semakin tinggi nilai SoC maka waktu yang dibutuhkan untuk mencapai kondisi penuh semakin kecil. Karakteristik ini menunjukkan bahwa SoC memiliki pengaruh yang signifikan terhadap waktu pengisian baterai dan dapat digunakan sebagai variabel prediktor dalam model Decision Tree dan XGBoost.

3.1 Hasil Algoritma Decision Tree

Gambar 8 menunjukkan perbandingan antara nilai aktual Time to Full Charge (TTF) dengan hasil prediksi menggunakan algoritma Decision Tree. Terlihat bahwa kurva prediksi mengikuti pola kurva data aktual dengan cukup baik pada seluruh rentang State of Charge (SoC).

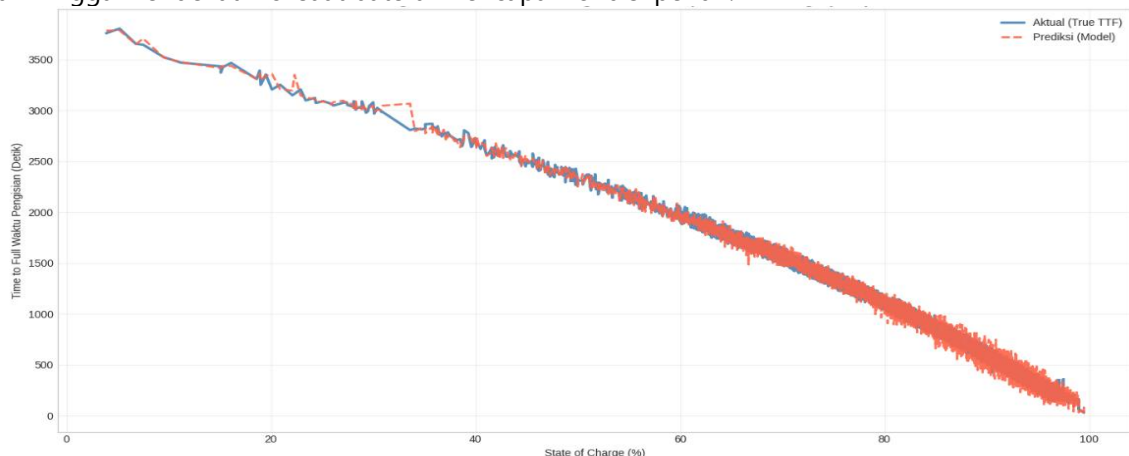


Gambar 8. aktual vs Preduksi Decision Tree

Perbedaan antara nilai aktual dan nilai prediksi relatif kecil, yang ditunjukkan oleh kedekatan posisi kedua kurva. Meskipun terdapat beberapa penyimpangan pada beberapa titik data, secara umum model Decision Tree mampu merepresentasikan hubungan antara SoC dan TTF dengan baik. Hasil ini menunjukkan bahwa algoritma Decision Tree memprediksi waktu pengisian penuh baterai Lithium Polymer dengan baik berdasarkan data pengisian yang digunakan.

3.2 Hasil Algoritma XGBoost

Gambar 9 menunjukkan perbandingan antara nilai aktual Time to Full Charge (TTF) dan hasil prediksi menggunakan algoritma XGBoost. Terlihat bahwa kurva prediksi mengikuti pola data aktual pada seluruh rentang State of Charge (SoC). Seiring meningkatnya nilai SoC, waktu pengisian penuh mengalami penurunan hingga mendekati nol saat baterai mencapai kondisi penuh.



Gambar 9. Aktual vs XGBoost

hasil prediksi XGBoost memiliki kesesuaian yang baik dengan data aktual, meskipun terdapat beberapa penyimpangan pada beberapa titik pengamatan. Kedekatan antara kurva prediksi dan kurva aktual menunjukkan bahwa algoritma XGBoost mampu mempelajari hubungan antara SoC dan waktu pengisian penuh dengan baik sehingga dapat digunakan untuk melakukan estimasi TTF pada baterai Lithium Polymer.

3.3 Analisis Prediksi Time To Full Charge menggunakan Decision Tree dan XGboost

Untuk mengetahui kemampuan algoritma Decision Tree dan XGBoost dalam memprediksi Time to Full Charge (TTF) baterai Lithium Polymer, dilakukan evaluasi kinerja model. Pengujian dilakukan dengan Dio Lizwardi: Perbandingan kinerja *Algoritma Decision*..... p-ISSN: 2356-0533; e-ISSN: 2355-9195



membandingkan hasil prediksi dengan data aktual dengan menggunakan metrik seperti Error Rata-rata Absolute (MAE), Error Rata-rata Persentase (MAPE), Error Rata-rata Kubik (RMSE), dan koefisien determinasi (R²). Hasil evaluasi kinerja kedua model ditunjukkan pada Tabel II.

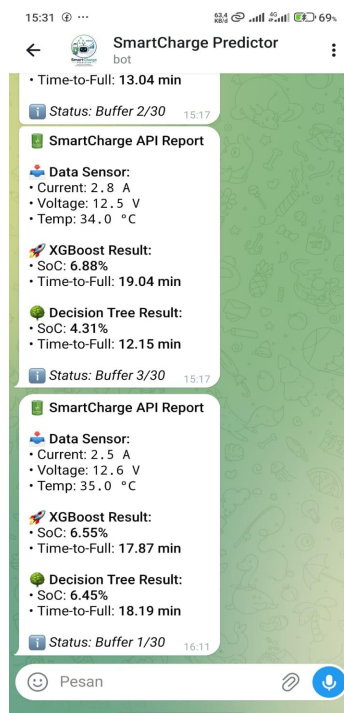
TABEL II. Perbandingan model

Model	MAE (s)	MAPE %	RMSE (s)	R ²
Decision Tree	44,5892	10,82	62,08	0,9905
XGBoost	47,1935	7,796	63,657	0,9906

Hasil menunjukkan bahwa kedua algoritma memiliki kemampuan prediksi yang luar biasa, dengan nilai R² hampir sama dengan 1. Berdasarkan nilai MAE dan RMSE, algoritma Decision Tree memberikan tingkat kesalahan yang lebih rendah dibandingkan XGBoost. Namun, berdasarkan nilai MAPE, algoritma XGBoost menghasilkan persentase kesalahan yang lebih kecil. Secara keseluruhan, kedua model mampu memprediksi waktu pengisian penuh baterai dengan tingkat akurasi yang tinggi, dengan perbedaan performa yang relatif kecil. Perbedaan nilai MAE dan RMSE menunjukkan bahwa algoritma Decision Tree menghasilkan kesalahan absolut yang sedikit lebih rendah dibandingkan XGBoost. Hal ini mengindikasikan bahwa Decision Tree telah mampu memodelkan hubungan antara parameter pengisian baterai dan *Time to Full Charge* (TTF) secara efektif pada dataset yang digunakan. Di sisi lain, XGBoost memperoleh nilai MAPE yang lebih rendah, yang menunjukkan bahwa model tersebut lebih konsisten dalam menjaga kesalahan prediksi relatif terhadap nilai aktual. Meskipun terdapat perbedaan pada masing-masing metrik evaluasi, kedua algoritma menunjukkan performa yang sangat baik dengan nilai R² di atas 0,99 sehingga keduanya layak digunakan untuk prediksi TTF baterai Lithium Polymer.

3.4 Visualisasi dan Kinerja Sistem

Hasil implementasi sistem monitoring ditampilkan melalui aplikasi Telegram sebagai media pemantauan jarak jauh. Sistem secara otomatis mengirimkan informasi hasil pembacaan sensor dan prediksi waktu pengisian baterai yang dihasilkan oleh algoritma Decision Tree dan XGBoost. Tampilan hasil monitoring ditunjukkan pada Gambar 10.



Gambar 10. Tampilan Pada Telegram



Berdasarkan Gambar 10, sistem monitoring berhasil menampilkan parameter pengisian baterai berupa tegangan, arus, suhu, *State of Charge* (SoC), serta hasil prediksi *Time to Full Charge* (TTF) secara near *real-time* melalui aplikasi Telegram. Integrasi antara ESP32, model prediksi, dan Telegram berjalan dengan baik sehingga informasi kondisi baterai dapat diterima pengguna tanpa keterlambatan yang berarti. Implementasi ini menunjukkan bahwa model Decision Tree dan XGBoost tidak hanya memberikan hasil prediksi yang akurat, tetapi juga dapat diintegrasikan ke dalam sistem monitoring berbasis IoT untuk mendukung pemantauan proses pengisian baterai secara jarak jauh.

3.5 Perbandingan Kinerja Model

Hasil penelitian menunjukkan bahwa algoritma Decision Tree dan XGBoost sama-sama mampu memprediksi waktu pengisian penuh (*Time to Full Charge* / TTF) baterai Lithium Polymer dengan tingkat akurasi yang tinggi. Hal ini ditunjukkan oleh nilai koefisien determinasi (R^2) yang mencapai lebih dari 0,99 pada kedua model, sehingga sebagian besar variasi data aktual dapat direpresentasikan dengan baik oleh model prediksi. Berdasarkan nilai MAE dan RMSE, algoritma Decision Tree menghasilkan tingkat kesalahan yang sedikit lebih rendah dibandingkan XGBoost. Hal ini menunjukkan bahwa Decision Tree memiliki kemampuan yang baik dalam memodelkan hubungan antara parameter pengisian baterai dan waktu pengisian penuh dengan kompleksitas model yang lebih sederhana. Sementara itu, XGBoost menghasilkan nilai MAPE yang lebih kecil, sehingga memiliki performa yang lebih baik dalam mengurangi kesalahan prediksi secara persentase.

Perbedaan performa kedua algoritma relatif kecil, sehingga keduanya layak digunakan untuk implementasi sistem prediksi TTF baterai Li-Po. Namun, apabila mempertimbangkan kebutuhan komputasi yang lebih ringan dan kemudahan implementasi pada sistem tertanam (*embedded system*), algoritma Decision Tree berpotensi menjadi pilihan yang lebih efisien untuk implementasi pada sistem monitoring dan prediksi waktu pengisian baterai berbasis perangkat tertanam (*embedded system*) karena kompleksitas komputasinya yang lebih rendah. Pada penelitian ini, proses prediksi dilakukan pada komputer menggunakan Python, sedangkan ESP32 berfungsi sebagai perangkat akuisisi data dan komunikasi. Hasil ini menunjukkan bahwa pemilihan algoritma sebaiknya disesuaikan dengan kebutuhan aplikasi. Decision Tree lebih unggul dari sisi efisiensi komputasi dan kesederhanaan implementasi, sedangkan XGBoost menawarkan keunggulan dalam menjaga kesalahan prediksi relatif tetap rendah.

4. KESIMPULAN

Berdasarkan hasil penelitian, sistem monitoring dan prediksi waktu pengisian baterai Lithium Polymer (Li-Po) berhasil diimplementasikan menggunakan algoritma Decision Tree dan XGBoost. Sistem mampu melakukan akuisisi data pengisian meliputi tegangan, arus, suhu, dan State of Charge (SoC) secara real-time, serta menampilkan hasil prediksi Time to Full Charge (TTF) melalui aplikasi Telegram. Hasil evaluasi menunjukkan bahwa kedua algoritma memiliki performa yang sangat baik dalam memprediksi TTF dengan nilai koefisien determinasi (R^2) di atas 0,99. Berdasarkan nilai MAE dan RMSE, algoritma Decision Tree memberikan tingkat kesalahan yang lebih rendah dibandingkan XGBoost, sedangkan XGBoost menghasilkan nilai MAPE yang lebih kecil. Secara keseluruhan, Kedua model memiliki kemampuan untuk memberikan prediksi yang akurat dengan perbedaan kinerja yang kecil. Dengan mempertimbangkan akurasi dan kompleksitas komputasi, algoritma Decision Tree berpotensi menjadi pilihan yang lebih efisien untuk implementasi pada sistem monitoring dan prediksi waktu pengisian baterai berbasis perangkat tertanam (*embedded system*), sementara XGBoost dapat digunakan ketika diperlukan tingkat akurasi persentase kesalahan yang lebih baik. Penelitian ini menggunakan dataset yang diperoleh dari satu siklus pengisian baterai sehingga evaluasi model dilakukan berdasarkan dataset yang tersedia. Oleh karena itu, penggunaan beberapa siklus pengisian dengan skema validasi berbasis siklus, seperti *leave-one-cycle-out validation*, disarankan pada penelitian selanjutnya untuk meningkatkan kemampuan generalisasi model dan meminimalkan potensi *data leakage*.



DAFTAR PUSTAKA

- [1] J. Chaoraingern and A. Numsomran, "Embedded Sensor Data Fusion and TinyML for Real-Time Remaining Useful Life Estimation of UAV Li Polymer Batteries," *Sensors*, vol. 25, no. 12, Jun. 2025, doi: 10.3390/s25123810.
- [2] R. Ranjith Kumar, C. Bharatiraja, K. Udhayakumar, S. Devakirubakaran, K. S. Sekar, and L. Mihet-Popa, "Advances in Batteries, Battery Modeling, Battery Management System, Battery Thermal Management, SOC, SOH, and Charge/Discharge Characteristics in EV Applications," 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2023.3318121.
- [3] Y. Wu, D. Bai, K. Zhang, Y. Li, and F. Yang, "Advancements in the estimation of the state of charge of lithium-ion battery: a comprehensive review of traditional and deep learning approaches," *Journal of Materials Informatics*, vol. 5, no. 2, Feb. 2025, doi: 10.20517/jmi.2024.84.
- [4] H. Hilal and P. Saha, "Forecasting Lithium-Ion Battery Longevity with Limited Data Availability: Benchmarking Different Machine Learning Algorithms," Dec. 2023, [Online]. Available: <http://arxiv.org/abs/2312.05717>
- [5] W. Guo, Z. Sun, S. B. Vilsen, J. Meng, and D. I. Stroe, "Review of 'grey box' lifetime modeling for lithium-ion battery: Combining physics and data-driven methods," Dec. 01, 2022, *Elsevier Ltd.* doi: 10.1016/j.est.2022.105992.
- [6] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," 2023, *Frontiers Media SA.* doi: 10.3389/frai.2023.1124553.
- [7] H. Aboueidah and A. Altafhan, "A Comparison of Baseline Models and a Transformer Network for SOC Prediction in Lithium-Ion Batteries," Oct. 2024, [Online]. Available: <http://arxiv.org/abs/2410.17049>
- [8] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Jun. 2016, doi: 10.1145/2939672.2939785.
- [9] S. Jafari and Y. C. Byun, "Efficient state of charge estimation in electric vehicles batteries based on the extra tree regressor: A data-driven approach," *Heliyon*, vol. 10, no. 4, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25949.
- [10] C. Lei, "New energy vehicle battery state of charge prediction based on XGBoost algorithm and RF fusion," *Energy Informatics*, vol. 7, no. 1, Dec. 2024, doi: 10.1186/s42162-024-00424-1.
- [11] M. F. Mat Baseri and A. F. Saad, "Dietary Monitoring System Using Decision Tree to Control Human Obesity," *International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 9–20, Jun. 2023, doi: 10.36079/lamintang.ijai-01001.485.
- [12] A. A. Alharbi, "Classification Performance Analysis of Decision Tree-Based Algorithms with Noisy Class Variable," *Discrete Dyn. Nat. Soc.*, vol. 2024, 2024, doi: 10.1155/2024/6671395.
- [13] M. Wiens, A. Verone-Boyle, N. Henscheid, J. T. Podichetty, and J. Burton, "A Tutorial and Use Case Example of the eXtreme Gradient Boosting (XGBoost) Artificial Intelligence Algorithm for Drug Development Applications," *Clin. Transl. Sci.*, vol. 18, no. 3, Mar. 2025, doi: 10.1111/cts.70172.
- [14] M. Noorunnahar, A. H. Chowdhury, and F. A. Mila, "A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh," *PLoS One*, vol. 18, no. 3 March, Mar. 2023, doi: 10.1371/journal.pone.0283452.
- [15] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.

