

Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki

Muhamad Amhar Rayadin¹, Mustarum Musaruddin², Rizal Adi Saputra³, Isnawaty⁴

¹Teknik Informatika, Universitas Halu Oleo, amharrayadin@gmail.com

²Teknik Elektronika, Universitas Halu Oleo, mustarum@uho.ac.id

³Teknik Informatika, Universitas Halu Oleo, rizaladisaputra@uho.ac.id

⁴Teknik Informatika, Universitas Halu Oleo, isnawaty@uho.ac.id

Keywords :

Ensemble Learning,
XGBoost,
Random Forest,
Prediction,
Battery

ABSTRACT

In motor vehicles, including cars, the battery plays an important role, namely as a place to store electrical energy and as an electric voltage stabilizer when the engine is turned on. In general, motorized vehicle users do not know the condition of the battery in their vehicle. Even though the use of battery batteries that are already in poor condition can interfere with vehicle performance. In battery replacement services such as after-sales service, the process of checking and replacing battery batteries takes a relatively long time. This can be caused by high service volume, lack of worker reliability, lack of responsiveness to the complexity of the inspection. This research aims to build a prediction model for battery battery replacement time quickly. To meet these needs, a Machine Learning approach can be used. Machine Learning uses historical replacement data to make predictions of replacement time. Machine Learning algorithms that can be used for prediction are XGBoost and Random Forest. This research uses ensemble learning techniques to combine the two models. Based on the evaluation results, it can be concluded that the model built with ensemble learning has better prediction results than a single model. Evaluation results with MSE on the ensemble bagging model have the lowest error values of 145,448. The MAPE, MAE, and RMSE evaluations on the ensemble boosting model have the lowest error values of 11.56 %, 43.80 and 38,760.

Kata Kunci :

Ensemble Learning,
XGBoost,
Random Forest,
Prediksi,
Baterai Aki,

ABSTRAK

Dalam kendaraan bermotor termasuk mobil, baterai aki memegang peranan penting, yakni menjadi tempat menyimpan energi listrik dan sebagai stabilisator tegangan listrik saat mesin dinyalakan. Pada umumnya, pengguna kendaraan bermotor tidak mengetahui kondisi baterai aki pada kendaraannya. Padahal penggunaan baterai aki yang sudah dalam keadaan kurang baik dapat mengganggu performa kendaraan. Pada layanan penggantian baterai aki seperti layanan purna jual, proses pemeriksaan dan penggantian baterai aki memerlukan waktu yang relatif lama. Hal ini dapat disebabkan oleh tingginya volume layanan, kurangnya keandalan pekerja, kurangnya responsivitas hingga kompleksitas pemeriksaan. Penelitian ini bertujuan untuk membangun model prediksi waktu penggantian baterai aki secara cepat. Untuk memenuhi kebutuhan tersebut, dapat menggunakan pendekatan *Machine Learning*. *Machine Learning* menggunakan data historis penggantian untuk membuat prediksi waktu penggantian. Algoritma *Machine Learning* yang dapat digunakan untuk prediksi adalah *XGBoost* dan *Random Forest*. Penelitian ini menggunakan teknik *ensemble learning* untuk menggabungkan kedua model tersebut. Berdasarkan hasil evaluasi, dapat disimpulkan bahwa model yang dibangun dengan *ensemble learning* memiliki hasil prediksi yang lebih baik daripada model tunggal. Hasil Evaluasi dengan MSE menunjukkan bahwa model bagging memiliki nilai error terendah yaitu 145,448. Adapun evaluasi berdasarkan MAPE, MAE dan RMSE menunjukkan bahwa model boosting memiliki nilai error terendah yaitu 11.56 %, 43.80 dan 38,760.

Korespondensi Penulis:

Muhamad Amhar Rayadin,
Universitas Halu Oleo, Kota Kendari, Indonesia
Telepon : +6282292249581
Email: amharrayadin@gmail.com

Submitted : 12-07-2024; Accepted : 22-08-2024;
Published : 27-08-2024

Copyright (c) 2024 The Author (s) This article is distributed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0)

1. PENDAHULUAN

Dalam kendaraan mobil, baterai aki (*accu*) sebagai salah satu suku cadang memegang peranan penting, yakni menjadi tempat menyimpan energi listrik. Baterai aki berfungsi sebagai sumber kelistrikan utama yang digunakan sebelum mesin mobil menyala. Saat mesin mobil menyala, alternator menggantikan baterai aki menjadi sumber energi listrik. Akan tetapi, baterai aki tetap memiliki fungsi sebagai stabilisator tegangan listrik [1]. Pada umumnya, pengguna kendaraan mobil tidak mengetahui kondisi baterai aki pada kendaraannya [2]. Padahal penggunaan baterai aki yang sudah dalam keadaan kurang baik atau rusak dapat mengganggu performa kendaraan yang digunakan.

Pada layanan penggantian baterai aki seperti layanan purna jual, proses pemeriksaan dan penggantian baterai aki memerlukan waktu yang relatif lama. Hal ini dapat disebabkan oleh beberapa faktor, seperti tingginya volume layanan, kurangnya keandalan pekerja, kurangnya responsivitas hingga kompleksitas pemeriksaan [3]. Pada umumnya proses pemeriksaan baterai aki pada mobil melibatkan beberapa langkah manual yang cukup rumit dan memerlukan ketelitian. Langkah – langkah pemeriksaan yang dilakukan yaitu pemeriksaan tegangan, berat jenis, hingga kondisi fisik baterai aki. Setelah itu barulah dilakukan perbaikan atau penggantian baterai aki sesuai dengan hasil pemeriksaan sebelumnya. Setelah dilakukan penggantian, dilakukan kembali pemeriksaan dan pengujian untuk memastikan kondisi aki yang telah diganti [4].

Melihat peran kunci baterai aki sebagai sumber kelistrikan pada kendaraan mobil dan tantangan mengenai efisiensi waktu pemeriksaan baterai aki, perlu dilakukan prediksi waktu penggantian suku cadang khususnya baterai aki. Prediksi waktu penggantian baterai aki pada mobil dapat menentukan kelayakan baterai aki untuk diperiksa secara cepat. Untuk memenuhi kebutuhan tersebut, sistem prediksi dapat dibangun menggunakan pendekatan *Machine Learning*. *Machine Learning* menggunakan data historis penggantian untuk membuat prediksi waktu penggantian.

XGBoost (Extreme Gradient Boosting) merupakan salah satu Algoritma *Machine Learning* yang biasa digunakan untuk prediksi [5], [6]. Dalam penelitian [7], Algoritma Regresi *XGBoost* digunakan untuk memprediksi harga rumah. *Dataset* dibagi menjadi tiga subset untuk pelatihan, validasi, dan pengujian. Setelah melatih setiap model, model tersebut disesuaikan dengan parameter regularisasi terbaik untuk menghindari *overfitting*. Hasil percobaan menunjukkan bahwa pembagian data pelatihan, pengujian, dan validasi sebesar 80%, 10%, 10% menunjukkan tingkat kesalahan rendah yakni 4.4%.

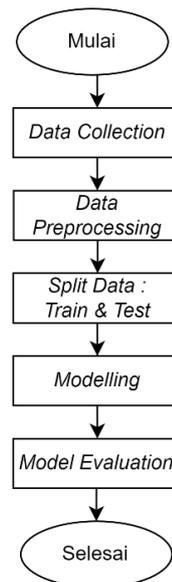
Selain *XGBoost*, Algoritma *Machine Learning* lainnya yang digunakan untuk prediksi adalah *Random Forest*. *Random Forest* adalah model yang populer digunakan untuk analisis data dengan dimensi yang besar [8]. Dalam penelitian [9], *Random Forest* digunakan untuk prediksi kasus positif Covid-19. Dari hasil penelitian yang dilakukan, *Random Forest* menghasilkan nilai RMSE sebesar 1886,555 dan MAPE 14,85. Ini lebih baik dari algoritma lainnya yakni Regresi Linear yang menghasilkan nilai RMSE sebesar 3031,127 dan MAPE 47,66.

Ensemble Learning merupakan teknik kombinasi algoritma *Machine Learning*. Beberapa model *Machine Learning* digabungkan untuk memperoleh hasil prediksi yang lebih baik dibanding menggunakan model tunggal [10], [11]. Berdasarkan penelitian [12], penggunaan teknik *majority vote* dari *Ensemble Learning* yang menerapkan beberapa algoritma ke dalam sebuah pemodelan dapat meningkatkan akurasi prediksi kinerja akademik mahasiswa dengan nilai RMSE sebesar 0,440.

Penelitian ini akan menggunakan pendekatan *Ensemble Learning* yang menggabungkan *XGBoost* dan *Random Forest*. Dengan menggunakan teknik *Ensemble Learning* dari 2 model kuat ini, diharapkan dapat menghasilkan prediksi waktu penggantian baterai aki pada mobil yang akurat. Penelitian ini bertujuan untuk mengukur performa model *ensemble* yang akan dibangun dalam melakukan prediksi waktu penggantian baterai aki.

2. METODE PENELITIAN

Tahapan metode penelitian dimulai dengan *data collecting*, *data preprocessing*, *split data*, *modelling*, dan *model evaluation*.



Gambar 1. Tahapan penelitian

2.1 Data Collection

Dalam penelitian ini, data yang digunakan berupa data historis penggantian baterai aki mobil Toyota di salah satu perusahaan otomotif di Kota Makassar. Data berjumlah 7703 baris data selama rentang waktu 2017 sampai 2023. Atribut data berupa *vehicle model* (model kendaraan), *mileage* (kilometer kendaraan), *battery type* (jenis baterai), *age* (usia kendaraan), *area* (karakteristik area). Adapun atribut target yaitu *time difference* (selisih waktu penggantian).

2.2 Data Preprocessing

Data preprocessing adalah langkah penting dalam proses analisis data yang melibatkan pembersihan dan transformasi data mentah menjadi format yang sesuai untuk analisis lebih lanjut. Tujuannya adalah untuk meningkatkan kualitas data dengan menangani masalah seperti nilai yang hilang, pencilan, dan ketidaksesuaian [13].

Beberapa teknik umum yang digunakan dalam *data preprocessing* meliputi:

1. Integrasi Data: Menggabungkan data dari berbagai sumber menjadi satu set data untuk memberikan pandangan yang lebih komprehensif.
2. Transformasi Data: Mengubah data ke dalam format yang sesuai untuk analisis, seperti normalisasi (penyekalaan data ke rentang standar) atau encoding variabel kategorikal menjadi nilai numerik [14].
3. Pembersihan Data: Menghapus atau memperbaiki kesalahan dalam dataset, menangani nilai yang hilang, dan mengatasi pencilan yang dapat memengaruhi hasil analisis [15].
4. Reduksi Data: Mengurangi dimensi dataset dengan memilih fitur yang relevan atau menerapkan teknik seperti *Principal Component Analysis* (PCA) untuk menyederhanakan data tanpa kehilangan informasi penting [16].

2.3 Split Data

Tahap *split data* ke dalam set pelatihan dan pengujian merupakan langkah krusial dalam pengembangan model Machine Learning. Dalam konteks ini, set pelatihan digunakan untuk melatih model agar dapat memahami pola-pola yang terdapat dalam data. Proporsi data yang dialokasikan ke dalam set pelatihan biasanya lebih besar, sekitar 70%, untuk memastikan model memiliki cukup informasi untuk belajar. Selama proses pelatihan, model akan menyesuaikan parameter internalnya berdasarkan data pelatihan sehingga dapat mempelajari hubungan antara fitur yang ada dalam data dan target output yang diinginkan [17].

Di sisi lain, set pengujian digunakan untuk menguji seberapa baik model yang telah dilatih dapat melakukan prediksi pada data yang belum pernah dilihat sebelumnya. Proporsi data yang dialokasikan ke dalam set pengujian biasanya lebih kecil, sekitar 30%, untuk memastikan bahwa model diuji pada data yang independen. Evaluasi kinerja model dilakukan dengan menggunakan set pengujian ini, yang membantu dalam menilai seberapa baik model dapat melakukan prediksi pada data baru dan mengidentifikasi apakah model cenderung *overfitting* atau tidak [17].

2.4 Ensemble Learning

Ensemble Learning adalah teknik dalam pembelajaran mesin (*machine learning*) yang menggunakan beberapa model pembelajaran untuk pencapaian solusi prediksi yang lebih baik daripada algoritma yang bisa diperoleh dari salah satu model konstituen saja. Istilah *ensemble* merujuk pada konsep menggabungkan banyak elemen menjadi satu kesatuan. Konsep penggabungan berbagai pendekatan pembelajaran mesin diharapkan dapat mengatasi kelemahan individu dari masing-masing model dan menciptakan prediksi yang lebih akurat dan stabil. Penggunaan beberapa model pembelajaran dan bukannya satu model tunggal untuk mengungkap masalah sulit yang diberikan telah selalu menjadi solusi yang diterima. *Ensemble Learning method* secara teoritis dan empiris telah diperoleh lebih baik kinerja yang lebih baik daripada pembelajar tunggal yang lemah, terutama saat menangani komputasi dengan dimensi tinggi, kompleks masalah regresi dan klasifikasi [18], [19].

Dalam kasus regresi, metode *ensemble* yang dapat digunakan adalah sebagai berikut :

1. Bagging

Metode *Bagging* (*Bootstrap Aggregating*) adalah salah satu pendekatan dalam *Ensemble Learning* di mana beberapa model pembelajaran digunakan secara paralel untuk meningkatkan stabilitas dan akurasi model pembelajaran mesin. Hasil prediksi dikombinasikan untuk menghasilkan prediksi akhir. Setiap model pembelajaran dilatih pada subset acak dari data pelatihan yang dihasilkan melalui proses *bootstrap* [20].

2. Stacking

Stacking adalah teknik *Ensemble Learning* yang menggabungkan beberapa model untuk meningkatkan akurasi prediksi. Ide dasar dari *stacking* adalah menggunakan *base learner* untuk menghasilkan *metadatas* untuk input kemudian menggunakan *meta-learner* yang umumnya disebut level 1 *learner* untuk memproses *metadatas* tersebut. *Base learner* biasanya disebut level 0 *learner*, dan *meta learner* yang ditumpuk pada *base learner* disebut *stacking* [21].

3. Boosting

Boosting adalah model *ensemble* yang dibentuk dari beberapa model dasar yang secara berurutan model dasar tersebut dilatih dan digabungkan dalam prediksi. Algoritma *boosting* membangun model secara bertahap dengan mengoptimalkan suatu fungsi kerugian. Setiap model dilatih pada sebuah set pelatihan yang diberi bobot. Bobot ini ditentukan berdasarkan kesalahan model sebelumnya dalam urutan tersebut. Ini merupakan metode yang dapat mengonversi pembelajar lemah menjadi pembelajar kuat [20], [22].

2.5 XGBoost

XGBoost (*Extreme Gradient Boosting*) merupakan *tree boosting* yang dapat diskalakan yang telah terbukti sebagai metode *machine learning* yang sangat efektif dan luas digunakan untuk melukan prediksi nilai dari suatu data. *XGBoost* menggunakan pohon keputusan sebagai model dasarnya dan menerapkan teknik penguatan untuk meningkatkan kinerja model. Sistem ini mencakup fitur-fitur regularisasi, *parallel processing*, dan *handling missing values*. Selain itu, *XGBoost* memperhatikan pola akses *cache*, kompresi data, dan *sharding* untuk membangun sistem penguatan pohon yang dapat diskalakan. Dengan menggabungkan berbagai teknik ini, *XGBoost* mampu menangani data besar menggunakan sumber daya yang lebih sedikit daripada sistem yang sudah ada [6], [23].

Model pohon *boosting* tradisional hanya menggunakan informasi turunan pertama. Ketika melatih pohon ke- n , sulit untuk mengimplementasikan pelatihan terdistribusi karena sisa dari pohon sebelumnya ke- $n-1$ digunakan. *XGBoost* melakukan ekspansi Taylor orde kedua pada fungsi kerugian dan secara otomatis dapat menggunakan multithreading dari CPU untuk komputasi paralel. Selain itu, *XGBoost* menggunakan berbagai metode untuk menghindari *overfitting* [24].

Algoritma *XGBoost* secara sederhana dituliskan pada Persamaan 1 sampai Persamaan 4 sebagai berikut. Integrasi model pohon dengan metode penjumlahan, diasumsikan jumlah dari K pohon, dan F menggambarkan model pohon dasar, maka :

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (1)$$

Objective function dituliskan :

$$L = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2)$$

Sebagai keterangan, l adalah *loss function* yang menggambarkan *error* antara nilai prediksi dengan nilai sebenarnya. Kemudian Ω adalah fungsi yang digunakan untuk *regularization* untuk mencegah *overfitting*, sebagai berikut :

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (3)$$

Nilai T mewakili jumlah daun per pohon, dan w mewakili bobot daun dari setiap pohon. Setelah ekspansi Taylor orde dua dari *objective function* dan perhitungan rinci lainnya, didapatkan *information gain* dari *objective function* setelah setiap pemisahan adalah :

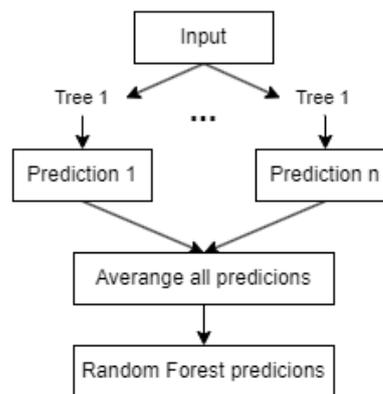
$$gain = \frac{1}{2} \left[\frac{(\sum_{i=I_L} g_i)^2}{\sum_{i=I_L} h_{i+\lambda}} + \frac{(\sum_{i=I_R} g_i)^2}{\sum_{i=I_R} h_{i+\lambda}} + \frac{(\sum_{i=I} g_i)^2}{\sum_{i=I} h_{i+\lambda}} \right] - \gamma \quad (4)$$

Keterangan :

- $gain$: *Information gain* yang dihitung setelah setiap pemisahan dalam proses pembangunan pohon. *Information gain* menggambarkan seberapa banyak informasi yang diperoleh dari suatu pemisahan.
- $\sum_{i=I_L} g_i$: Total *gradien* dari data yang masuk ke *node* kiri setelah pemisahan.
- $\sum_{i=I_L} h_i$: Total *hessian* dari data yang masuk ke *node* kiri setelah pemisahan. *Hessian* digunakan untuk menyesuaikan gradien dan memberikan bobot lebih besar pada pengamatan yang lebih signifikan.
- $\sum_{i=I_R} g_i$: Total *gradien* dari data yang masuk ke *node* kanan setelah pemisahan.
- $\sum_{i=I_R} h_i$: Total *hessian* dari data yang masuk ke *node* kanan setelah pemisahan.
- $\sum_{i=I} g_i$: Total *gradien* dari data pada *node* sebelum pemisahan.
- $\sum_{i=I} h_i$: Total *hessian* dari data pada *node* sebelum pemisahan.
- λ : Parameter regularisasi yang digunakan untuk mengontrol kompleksitas model dan mencegah *overfitting*.
- γ : *Threshold* untuk mengontrol pertumbuhan pohon. Pemisahan *node* hanya akan terjadi jika *information gain* lebih besar dari nilai *threshold* γ

2.6 Random Forest

Random Forest adalah salah satu algoritma *Machine Learning* yang paling sering digunakan. Algoritma ini pertama kali dipublikasikan dalam [25]. Metode ini menggabungkan beberapa pohon keputusan (*Decision Tree*) yang dibangun secara acak dan independen satu sama lain. Setiap pohon keputusan dalam *Random Forest* dibangun dengan menggunakan subset acak dari fitur-fitur yang tersedia dalam data. Prediksi dalam regresi dilakukan dengan merata-ratakan prediksi dari setiap pohon.



Gambar 2. Alur *Random Forest*

2.7 Model Evaluation

Pengukuran kinerja model yang dilatih dapat menggunakan metrik evaluasi seperti *Mean Absolute Percentage Error* (MAPE), *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE) dan *Root Mean Squared Error* (RMSE).

1. Mean Absolute Percentage Error (MAPE)

MAPE adalah metrik evaluasi yang mengukur rata-rata dari persentase kesalahan absolut antara nilai aktual dan nilai ramalan. Semakin rendah nilai MAPE, semakin tinggi tingkat akurasi model peramalan tersebut [26]. Secara matematis, MAPE dituliskan dalam Persamaan 5.

$$MAPE = \frac{1}{m} \sum_{i=1}^m \left| \frac{Y_i - X_i}{Y_i} \right| \times 100 \quad (5)$$

Keterangan :

- m : Jumlah data dalam dataset.
- i : Data ke- i .
- X_i : Prediksi nilai ke- i .

Y_i : Nilai aktual data ke- i

2. Mean Absolute Error (MAE)

MAE adalah metrik evaluasi yang digunakan dalam analisis regresi untuk mengukur kesalahan prediksi dari model regresi dalam skala yang sama dengan variabel dependen. MAE menghitung rata-rata dari selisih absolut antara nilai prediksi yang dihasilkan oleh model regresi dan nilai aktual dari variabel dependen untuk setiap observasi dalam *dataset* [27]. Secara matematis, MSE dituliskan dalam Persamaan 6.

$$MAE = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i| \quad (6)$$

Keterangan :

m : Jumlah data dalam *dataset*.

i : Data ke- i .

X_i : Prediksi nilai ke- i .

Y_i : Nilai aktual data ke- i .

3. Mean Squared Error (MSE)

MSE adalah digunakan untuk mengukur seberapa baik model regresi memprediksi nilai aktual dari variabel dependen. MSE menghitung rata-rata dari kuadrat selisih antara nilai prediksi yang dihasilkan oleh model regresi dan nilai aktual dari variabel dependen untuk setiap observasi dalam *dataset* [27]. Secara matematis, MSE dituliskan dalam Persamaan 7.

$$MSE = \frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2 \quad (7)$$

Keterangan :

m : Jumlah data dalam *dataset*.

i : Data ke- i .

X_i : Prediksi nilai ke- i .

Y_i : Nilai aktual data ke- i .

4. Root Mean Squared Error (RMSE)

RMSE adalah metrik evaluasi digunakan untuk mengukur tingkat kesalahan prediksi dari model regresi dalam unit yang sama dengan variabel dependen. RMSE menghitung akar kuadrat dari rata-rata dari kuadrat selisih antara nilai prediksi yang dihasilkan oleh model regresi dan nilai aktual dari variabel dependen untuk setiap observasi dalam *dataset* [27]. Secara matematis, MSE dituliskan dalam Persamaan 8.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (X_i - Y_i)^2} \quad (8)$$

Keterangan :

m : Jumlah data dalam *dataset*.

i : Data ke- i .

X_i : Prediksi nilai ke- i .

Y_i : Nilai aktual data ke- i .

3. HASIL DAN ANALISIS

3.1 Data Preprocessing

Tahap data preprocessing dilakukan dengan langkah – langkah berikut :

1. Menghilangkan *outlier* data pada fitur kategorikal.
2. Melakukan pemilihan fitur untuk *modelling*. Fitur terdiri dari 5 atribut yaitu jenis kendaraan, selisih kilometer, jenis baterai, usia kendaraan, dan karakteristik area. Adapun target model adalah selisih waktu pengantian dalam satuan hari.
3. Melakukan transformasi data. Untuk fitur kategorikal dilakukan *one hot encoding* sedangkan fitur numerikal dilakukan *scaling* dengan *standar scaling*.
4. Melakukan *split* data menjadi data *train* dan data *test*. Data dibagi dengan rasio 70:30.

3.2 Modelling

Tahap *modelling* dilakukan dengan melatih model tunggal *XGBoost* dan *Random Forest* terlebih dahulu, kemudian melatih model *ensemble* dari model tunggal yang telah dibuat. Pelatihan model dilakukan dengan kode program berbahasa pemrograman Python. Adapun detail dari setiap model adalah sebagai berikut :

1. *XGBoost*

Proses pemodelan *XGBoost* diawali dengan *hyperparameter tuning* untuk mendapatkan parameter terbaik. Skema *hyperparameter tuning* dituliskan pada Tabel 1.

Tabel 1. *Hyperparameter tuning* model *XGBoost*

Parameter	Nilai
n_estimators	[20, 40, 60, 80, ... , 180, 200]
learning_rate	[0,001, 0,01, 0,1]
max_depth	[None, 3, 4, 5, 6]
subsample	[0,6, 0,7, 0,8]
reg_lambda	[0,5, 1, 1,5]
gamma	[0, 0,1, 0,2]
min_child_weight	[1, 10, 20, 25]

Hasil *hyperparameter tuning* model *XGBoost* yang dituliskan pada Tabel 2.

Tabel 2. Parameter model *XGBoost*

Parameter	Nilai
n_estimators	60
learning_rate	0,1
max_depth	6
subsample	0,8
reg_lambda	1
gamma	0
min_child_weight	1

2. *Random Forest*

Proses pemodelan *Random Forest* diawali dengan *hyperparameter tuning* untuk mendapatkan parameter terbaik. Skema *hyperparameter tuning* dituliskan pada Tabel 3.

Tabel 3. *Hyperparameter tuning* model *Random Forest*

Parameter	Nilai
max_depth	[5, 10, 15, 20, ... , 45, 50]
min_sample_split	[None, 2, 3, 4, 5, ... , 10, 11]
n_estimator	[10, 20, 30, 40, ... , 190, 200]

Hasil *hyperparameter tuning* model *Random Forest* dituliskan pada Tabel 4.

Tabel 4. Parameter model *Random Forest*

Parameter	Nilai
max_depth	20
min_sample_split	3
n_estimator	160

3. *Bagging*

Model *bagging* merupakan kombinasi model *XGBoost* dan *Random Forest* yang telah dilatih sebelumnya. *Output* model *bagging* adalah rata – rata dari prediksi *XGBoost* dan *Random Forest*.

4. *Stacking*

Model *stacking* menggunakan prediksi dari model *XGBoost* dan *Random Forest* untuk membuat *meta-model Linear Regression*. Proses pemodelan *meta-model* diawali dengan *hyperparameter tuning* untuk mendapatkan parameter terbaik. Skema *hyperparameter tuning* dituliskan pada Tabel 5.

Tabel 5. Parameter model *Linear Regression*

Parameter	Nilai
learning_rate	[0,001, 0,01, 0,1]
n_iterations	[100, 200, 300, 400, ... 1000]
scaler	['standard', 'minmax', 'maxabs', 'robust']

Hasil *hyperparameter tuning meta-model Linear Regression* dituliskan pada Tabel 6.

Tabel 6. Parameter model *Linear Regression*

Parameter	Nilai
learning_rate	0,1
n_iterations	600
scaler	maxabs

5. *Boosting*

Model *boosting* dibangun dari hasil prediksi model *Random Forest* sebelumnya, kemudian dilakukan *boosting*. Parameter model *boosting* sama dengan parameter *XGBoost* yang dituliskan pada Tabel 2.

3.3 *Model Evaluation*

Evaluasi performa model dilakukan dengan menggunakan metrik evaluasi. Metrik evaluasi yang digunakan yaitu MAPE, MAE, MSE, dan RMSE. Adapun hasil pengujian dari model yang telah dibuat dituliskan pada Tabel 7.

Tabel 7. Hasil evaluasi

Model	MAPE	MAE	MSE	RMSE
<i>XGBoost</i>	11,01 %	41,28	147,433	41,283
<i>Random Forest</i>	11,47 %	45,47	146,633	45,470
<i>Bagging</i>	11,18 %	43,11	145,448	43,116
<i>Stacking</i>	11,86 %	45,53	146,463	45,539
<i>Boosting</i>	10,96 %	38,76	155,388	38,760

Berdasarkan hasil evaluasi, dapat dilihat bahwa evaluasi dengan MSE menunjukkan bahwa model *bagging* memiliki nilai *error* terendah yaitu 145,448. Adapun evaluasi berdasarkan MAPE, MAE dan RMSE menunjukkan bahwa model *boosting* memiliki nilai *error* terendah yaitu 11.56 %, 43.80 dan 38,760. Oleh karena itu dapat disimpulkan bahwa model yang dibangun dengan *ensemble learning* memiliki hasil prediksi yang lebih baik daripada model tunggal.

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa model yang dibangun dengan *ensemble learning* memiliki hasil prediksi yang lebih baik daripada model tunggal. Hasil Evaluasi dengan MSE menunjukkan bahwa model *bagging* memiliki nilai *error* terendah yaitu 145,448. Adapun evaluasi berdasarkan MAPE, MAE dan RMSE menunjukkan bahwa model *boosting* memiliki nilai *error* terendah yaitu 11.56 %, 43.80 dan 38,760. Dari hasil evaluasi pemodelan tersebut, dapat dilakukan prediksi waktu penggantian baterai aki pada mobil. Adapun saran peneliti untuk penelitian selanjutnya adalah untuk menambah atribut prediksi yang dapat memengaruhi performa model, melakukan penambahan *dataset*, hingga membuat implementasi sistem dalam *website* atau *mobile* agar dapat digunakan oleh *user*.

REFERENSI

[1] H. Firdaus, E. Rustendi, and A. Herdiana, "Analisis Konsumsi Arus Listrik pada Mobil Multi Purpose Vehicle," *J. Ilm. Teknol. Infomasi Terap.*, vol. 8, no. 1, pp. 150–158, 2021, doi: 10.33197/jitter.vol8.iss1.2021.736.

[2] A. F. Fahmi and I. M. Arsana, "Alat Pendeteksi Kerusakan Cell Accu," *J. Rekayasa Mesin*, vol. 06, no. 03, pp. 1–7, 2021, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/jurnal-rekayasa-mesin/article/view/41987>

[3] J. R. Balinado, Y. T. Prasetyo, M. N. Young, S. F. Persada, B. A. Miraja, and A. A. N. Perwira Redi, "The Effect of Service Quality on Customer Satisfaction in An automotive After-Aales Aervice," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 2, 2021, doi: 10.3390/joitmc7020116.

[4] I. Prasetyo and I. Saputro, "Perbaikan dan Perawatan Aki Basah," *J. Surya Tek.*, vol. 2, no. 2, pp. 16–21, 2018.

[5] S. Y. J. Prasetyo, Y. B. Christianto, and K. D. Hartomo, "Analisis Data Citra Landsat 8 OLI Sebagai Indeks Prediksi Kekeringan Menggunakan Machine Learning di Wilayah Kabupaten Boyolali dan Purworejo," *Indones. J. Model. Comput.*, vol. 2, no. 2, pp. 25–36, 2019, [Online]. Available: <https://ejournal.uksw.edu/icm/article/view/2954>

[6] S. Usman, "Predictive Sparepart Maintenance Menggunakan Algoritma Machine Learning Extreme Gradient Boosting Regressor," *J. Syst. Comput. Eng.*, vol. 5, no. 2, pp. 249–258, 2024.

[7] J. Avanija, G. Sunitha, K. R. Madhavi, P. Kora, and R. H. SaiVittal, "Prediction of House Price Using XGBoost Regression Algorithm," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 2151–2155, 2021, doi: 10.17762/turcomat.v12i2.1870.

[8] D. Borup, B. J. Christensen, N. S. Mühlbach, and M. S. Nielsen, "Targeting Predictors in Random Forest Regression," *Int. J. Forecast.*, vol. 39, no. 2, pp. 841–868, 2023, doi: <https://doi.org/10.1016/j.ijforecast.2022.02.010>.

[9] S. Fachid and A. Triayudi, "Perbandingan Algoritma Regresi Linier dan Regresi Random Forest Dalam Memprediksi Kasus Positif Covid-19," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 68, 2022, doi: 10.30865/mib.v6i1.3492.

[10] J. M. A. S. Dachi and P. Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *J. Ris. Rumpun Mat. dan Ilmu Pengetah. Alam*, vol. 2, no. 2, pp. 87–103, 2023, doi: <https://doi.org/10.55606/jurrimipa.v2i2.1336>.

[11] B. Sunarko *et al.*, "Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat," *Edu*

- Komputika J.*, vol. 10, no. 1, pp. 55–63, 2023, doi: <https://doi.org/10.15294/edukomputika.v10i1.72080>.
- [12] U. Indahyanti, N. L. Azizah, and H. Setiawan, “Pendekatan Ensemble Learning Untuk Meningkatkan Akurasi Prediksi Kinerja Akademik Mahasiswa,” *J. Sains dan Inform.*, vol. 8, no. 2, pp. 160–169, 2022, doi: 10.34128/jsi.v8i2.459.
- [13] I. M. Hamdani, A. Karman, N. F. A. H, and A. Hermina, “Edukasi dan Pelatihan Data Science dan Data Preprocessing,” *INTISARI J. Inov. Pengabd. Masy.*, vol. 2, no. 1, pp. 19–26, 2024, doi: 10.58227/intisari.v2i1.125.
- [14] F. Alghifari and D. Juardi, “Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes,” *J. Ilm. Inform.*, vol. 9, no. 02, pp. 75–81, 2021, doi: 10.33884/jif.v9i02.3755.
- [15] M. A. Wiratama and W. M. Pradnya, “Optimasi Algoritma Data Mining Menggunakan Backward Elimination untuk Klasifikasi Penyakit Diabetes,” *J. Nas. Pendidik. Tek. Inform.*, vol. 11, no. 1, p. 1, 2022, doi: 10.23887/janapati.v11i1.45282.
- [16] D. Sartika and I. Saluza, “Penerapan Metode Principal Component Analysis (PCA) Pada Klasifikasi Status Kredit Nasabah Bank Sumsel Babel Cabang KM 12 Palembang Menggunakan Metode Decision Tree,” *Generic*, vol. 14, no. 2, pp. 45–49, 2022, doi: 10.18495/generic.v14i2.130.
- [17] H. Tao *et al.*, “Training and Testing Data Division Influence on Hybrid Machine Learning Model Process: Application of River Flow Forecasting,” *Complexity*, vol. 2020, pp. 1–22, 2020, doi: 10.1155/2020/8844367.
- [18] E. K. Sahin, “Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest,” *SN Appl. Sci.*, vol. 2, no. 7, pp. 1–17, 2020, doi: 10.1007/s42452-020-3060-1.
- [19] E. Mardiani, N. Rahmansyah, S. Ningsih, and ..., “Komparasi Metode Knn, Naive Bayes, Decision Tree, Ensemble, Linear Regression Terhadap Analisis Performa Pelajar Sma,” *Innov. J. ...*, vol. 3, no. 2, pp. 13880–13892, 2023, [Online]. Available: <http://j-innovative.org/index.php/Innovative/article/view/1949%0Ahttp://j-innovative.org/index.php/Innovative/article/download/1949/1468>
- [20] S. Choi and J. Hur, “An Ensemble Learner-Based Bagging Model Using Past Output Data for Photovoltaic Forecasting,” *Energies*, vol. 13, no. 6, 2020, doi: 10.3390/en13061438.
- [21] M. Liang *et al.*, “A Stacking Ensemble Learning Framework for Genomic Prediction,” *Front. Genet.*, vol. 12, no. March, pp. 1–9, 2021, doi: 10.3389/fgene.2021.600040.
- [22] S. C. Pal *et al.*, “Ensemble of machine-learning methods for predicting gully erosion susceptibility,” *Remote Sens.*, vol. 12, no. 22, pp. 1–25, 2020, doi: 10.3390/rs12223675.
- [23] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.
- [24] W. Li, Y. Yin, X. Quan, and H. Zhang, “Gene Expression Value Prediction Based on XGBoost Algorithm,” *Front. Genet.*, vol. 10, no. November, pp. 1–7, 2019, doi: 10.3389/fgene.2019.01077.
- [25] L. Breiman, “Random Forests,” *Mach. Learn.* 45, vol. 5, no. 32, pp. 5–32, 2001, doi: <https://doi.org/10.1023/A:1010933404324>.
- [26] E. Vivas, H. Allende-Cid, and R. Salas, “A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score,” *Entropy*, vol. 22, no. 12, pp. 1–24, 2020, doi: 10.3390/e22121412.
- [27] 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.