ISSN: 2339-2541
JURNAL GAUSSIAN, Volume 15, Nomor 1, Tahun 2026, Halaman 01 - 11
Online di: https://ejournal3.undip.ac.id/index.php/gaussian/

OPTIMASI XGBOOST DALAM PREDIKSI KECEPATAN KENDARAAN SECARA *REAL-TIME* : PERBANDINGAN METODE TUNING *HYPERPARAMETER*

Panji Lokajaya Arifa[1*], Kusman Sadik[2], Agus M. Soleh,[3] Cici Suhaeni[4]
[1,2,3,4] Program Studi Statistika dan Sains Data, Sekolah Sains Data, Matematika dan Informatika, Institut Pertanian Bogor
***e-mail**: arf4arifa@apps.ipb.ac.id

DOI: 10.14710/j.gauss.15.1.01-11

Article Info:
Received: 2025-06-25
Accepted: 2026-01-29
Available Online: 2026-01-31

Keywords:
Xgboost; Hyperparameter Tuning; Grid Search; Bayesian Optimization; Genetic Algorithms

Abstract: Real-time vehicle speed prediction plays a vital role in the development of intelligent transportation systems aimed at improving traffic flow and safety. This study investigates the performance of the XGBoost algorithm enhanced with three hyperparameter tuning techniques: Grid Search, Bayesian Optimization, and Genetic Algorithm. A simulated dataset was constructed reflect diverse urban traffic scenarios, incorporating environmental variables such as weather, road conditions, and traffic density. The models were assessed using 5 and 10-fold cross-validation based on prediction metrics (MSE, RMSE, MAE and R^2) as well as computational efficiency in terms of training and inference time. The findings reveal that Bayesian Optimization achieves the highest prediction accuracy, while Grid Search offers the fastest training time. Genetic Algorithm demonstrates a balanced trade-off between accuracy and computational efficiency, making it a competitive and practical choice. These results highlight the importance of selecting hyperparameter tuning strategies based on specific system needs in real-time traffic prediction using XGBoost.

1. PENDAHULUAN

Sistem transportasi saat ini semakin mengandalkan teknologi kecerdasan buatan guna meningkatkan efektivitas arus lalu lintas, menurunkan angka kecelakaan dan mengatasi kemacetan. Salah satu isu utama dalam manajemen lalu lintas adalah memprediksi kecepatan secara langsung guna mendukung proses pengambilan keputusan dalam manjemen lalu lintas (Putatunda & Rama, 2020). Kecepatan kendaraan dapat dipengaruhi oleh beragam faktor dan karakteristik yang rumit seperti waktu, lokasi, kondisi cuaca, keadaan jalan, dan kepadatan lalu lintas. (Tian *et al.*, 2023). Berbagai pendekatan model telah diterapkan untuk meningkatkan akurasi prediksi kecepatan kendaraan, seperti *Autoregressive Integrated Moving Average* (ARIMA) dan *Gaussian Processes*. Namun, model ini memiliki keterbatasan dalam menangani hubungan *non-linear*.

Model pembelajaran mesin seperti *Extreme Gradient Boosting (XGBoost)* adalah algortima *boosting* berbasis pohon keputusan yang telah teruji kehandalannya dan banyak diaplikasi secara luas dalam melakukan prediksi nilai dari suatu data (Rayadin *et al.*, 2024). Dalam penelitian (Chen and Guestrin, 2016) menunjukkan bahwa XGBoost unggul dalam menangani data non-linear dengan kompleksitas tinggi. XGBoost juga dikenal karena memiliki *Hyperparameter* yang beragam memungkinkan untuk memberi peneliti kendali penuh dalam penerapannya sehingga dapat mengontrol semua proses menjadi lebih fleksibel. Penelitian oleh (Martinez-Munoz, 2021) mengungkapkan bahwa perfroma XGBoost sangat bergantung pada optimalisasi hyperparameter. Optimasi Hyperparameter merupakan komponen penting dalam pembelajaran mesin yang mempunyai pengaruh besar terhadap efektivitas suatu algoritma. Dengan pemilihan *Hyperparameter* dan

mengkombinasikan berbagai optimasi, *XGBoost* dapat secara efektif mengatasi masalah *overfitting* yang sering muncul dalam algoritma *Gradient Boosting* (Mehdary *et al.*, 2024).

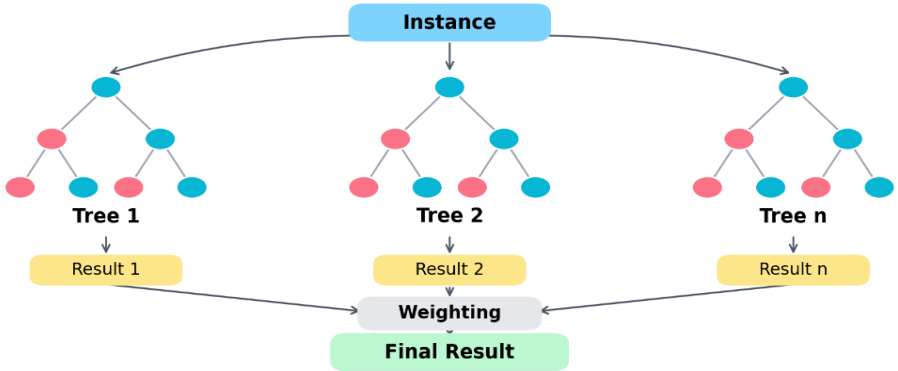
Penelitian terdahulu masih mempunyai keterbatasan yaitu hanya menggunakan satu metode tuning *hyperparameter* XGBoost dengan ruang parameter terbatas dan dalam evaluasi model hanya menekankan akurasi model berdasarkan nilai MSE dan RMSE tanpa mempertimbangkan aspek waktu komputasi yang menjadi faktor kritis dalam model prediksi. Kemudian implementasi optimasi metode tuning *hyperpamarater* hanya merepresentasikan beberapa skenario saja sehingga tidak memberikan validasi yang lebih komprehensif terhadap robustnes model dalam berbagai kondisi aktual yang ada.

Dalam mengatasi permasalahan diatas penelitian ini menganalisis dan membandingkan 3 metode optimasi *hyperparameter* dengan model *XGBoost* dalam memprediksi kecepatan kendaraan secara real-time dengan berbagai skenario dinamis untuk menguji kestabilan model Pelatihan Model ini akan menggunakan *k-fold cross validation* dengan metrik evaluasi dan akan dikembangkan dengan evaluasi yang mengintegrasikan aspek komputasi antara waktu pelatihan dan waktu pengujian sehingga menghasilkan *trade off* antar metodenya. Hasil ini diharapkan dapat memberikan wawasan tentang metode tuning *hyperparameter* yang paling efektif dalam meningkatkan performa model XGBoost terutama dalam memprediksi kecepatan kendaraan.

2. TINJAUAN PUSTAKA

XGBoost adalah algoritma dalam pembelajaran mesin berbasis pohon keputusan yang dapat digunakan dalam pemodelan prediktif. Algoritma ini bersifat iteratif dan bertujuan untuk meminimalkan kesalahan dalam prediksi dengan menggabungkan beberapa pohon keputusan secara bertahap. Ketika proses pelatihan model dilakukan melalui boosting dengan menghitung turunan pertama dan kedua dari fungsi loss, sehingga algoritma akan mampu menangani kompleksitas data dengan efisien (Wang *et al.*, 2024).

Proses pelatihan model pada pembelajaran mesin dengan dalam XGBoost berlanjut secara iteratif hingga error prediksi tidak mengalami penurunan yang signikan, sehingga mengindikasikan bahwa model telah mencapai konvergensi pada nilai optimal (Fatihah *et al.*, 2024). Dalam mengurangi kesalahan ketika membentuk model baru, XGBoost menggunakan pendekatan *gradient decline* yang secara bertahap membentuk pohon baru secara iteratif untuk memperbaiki hasil prediksi dan setiap model pohon dimulai berdasarkan kesalahan residual dari model sebelumnya sehingga hubungan non-linear antar fitur dipelajari secara bertahap sampai menghasilkan prediksi akhir dari penjumlahan kontribusi seluruh pohon (Chen and Guestrin, 2016).



Gambar 1. Proses ilustrasi pohon XGBoost

Formulasi prediksi model ini didefinisikan persamaan 1 berikut :

$$\hat{y}t = \sum_{k=1}^{n} f_k(x_i) \quad (1)$$

dengan $\hat{y}t$ prediksi akhir setelah t model dibentuk, $f_k(x_i)$ output dari pohon ke- k untuk data ke- i, dan n jumlah pohon yang digunakan.

Hyperparameter tuning merupakan proses penting untuk meningkatkan performa model *machine learning*, terutama XGBoost yang memiliki banyak parameter yang dapat dikombinasikan. Parameter tersebut memiliki pengaruh signifikan terhadap ketepatan prediksi dan tingkat kerumitan model. Pemilihan kombinasi parameter tidak hanya mencegah terjadinya overfitting maupun underfitting, tetapi juga mampu meningkatkan efisiensi proses pelatihan (Kandasamy *et al.*, 2020). Penelitian ini mengeksplorasi 3 metode optimasi hyperparameter untuk *XGBoost* dalam prediksi kecepatan kendaraan secara real-time menggunakan data simulasi, yaitu:

1. *Grid Search* adalah metode optimasi *hyperparameter* yang bekerja mengevaluasi seluruh kemungkinan kombinasi parameter dalam rentang nilai yang telah ditetapkan untuk suatu algoritma (Putatunda and Rama, 2018). Namun, Metode ini membutuhkan biaya komputasi yang tinggi, khususnya ketika menghadapi parameter memiliki dimensi tinggi (Probst, Wright, *et al.*, 2019). Formula untuk mencari kombinasi parameter $\hat{\lambda}$ yang didefinisikan dalam persamaan (2) berikut.

$$\hat{\lambda} = arg\,min\ \psi(\lambda), \qquad \psi(\lambda) = \frac{1}{k}\sum_{i=1}^{k} L\left(f_\lambda\left(D^i_{train}\right), D^i_{eval}\right) \quad (2)$$

dengan *L* menyatakan fungsi loss per observasi

$$\mathrm{L}(\hat{y}, y) = (y - \hat{y})^2, \quad (3)$$

sehingga MSE pada fold ke-i adalah

$$L\left(f_\lambda\left(D^i_{train}\right), D^i_{eval}\right) = \frac{1}{|D^i_{eval}|}\sum_{(x_j,y_j)\epsilon D^i_{eval}} (y_j - f_\lambda(x_j))^2 \quad (4)$$

$f_\lambda\left(D^i_{train}\right)$ adalah model XGBoost yang dilatih menggunakan data latih D^i_{train} pada kombinasi *hyperparameter* λ, dan D^i_{eval} sebagai subset data evaluasi pada fold ke-*i* (James Bergsrtra, 2012).

2. *Bayesian Optimization* adalah metode yang memanfaatkan pendekatan probabilistic untuk mengidentifikasi *hyperparameter* yang optimal dengan jumlah iterasi yang lebih sedikit dibanding *grid searc*h (Lambora *et al.*, 2019). Metode ini tidak hanya unggul dalam efisiensi komputasi, tetapi juga mampu menghasilkan solusi optimal dengan performa tinggi (Shahriari *et al.*, 2016). Metode ini bekerja bekerja teorema Bayes, dengan memodelkan ketidakpastian dari fungsi objektif dengan formula pada persamaan 5:

$$p(f\ |D_{1:t}) = \frac{p\ (\ D_{1:t}|f)\ p(f)}{p(D)} \quad (5)$$

p($\mathrm{D_{1:t}}$|f) *likelihood* persamaan 5 dinyatakan pada persamaan 6

$$p\ (\ D_{1:t}|f) = \prod_{i=1}^{t} N(y_i\ ; f(\lambda_i), \sigma^2) \quad (6)$$

p(f) sebagai prior dari fungsi, dengan $y_{\mathrm{i}=}\psi(\lambda_i)$ adalah fungsi objektif pada titik λ_{i}, dan σ^2 menyatakan varians noise observasi. Metode ini membangun dua komponen penting yaitu model surrogate dan fungsi akuisisi, selanjutnya model akan memperkirakan nilai minimum dari fungsi objektif dengan jumlah evaluasi yang lebih sedikit dibanding metode lainnya (J Cui, 2018).

3. *Genetic Alghorithm*s (GA) adalah metode optimasi berbasis komputasi dengan algoritma pencarian parameter yang terinsipirasi dari mekanisme seleksi alam dan pewarisan genetika. Algoritma ini mengadopsi prinsip evolusi biologis dalam lingkungan digital dimulai dengan proses seleksi, rekombinasi (crossover), dan mutasi yang dilakukan secara iteratif hingga diperoleh hasil dan solusi yang otpimal (Mehdary *et al.*, 2024). GA dirancang untuk mengeskplorasi ruang pencarian yang kompleks secara global dan probabilistik, sehingga sangat efektid untuk menyelesaikan masalah dengan banyak parameter multidimensi dan struktur yang tidak linear. Formula untuk setiap prosesnya :
    - Inisialisasi populasi diambil sebanyak *n* individu yang dihasilkan secara acak, dimana masing-masing mempresentasikan kombinasi *hyperparameter*.
    - Evaluasi fungsi *fitness*, dengan metriknya
    $$f = \frac{1}{n}\sum_{i=1}^{n}(y_k - \hat{y})^2 \tag{7}$$
    dengan y_k nilai aktual data, $\hat{y}$ nilai prediksi dari model berdasarkan persamaan (7)
    - Seleksi, *crossover,* dan mutasi, probabilitas pemilihan setiap individu dihitung dengan
    $$p_i = \frac{f_i}{\Sigma_i f_i} \tag{8}$$
    - Generasi baru dihasilkan dengan
    $$P(t+1) = mutasi(crossover\left(seleksi(P(t))\right)) \tag{9}$$
    proses ini diulang iteratif hingga diperoleh hasil optimal (Huang *et al.*, 2025).

Jika model dari tuning *hyperparameter* dilatih hanya dengan data terbaik dari satu subset, maka kemampuannya untuk diuji pada data lain bisa menurun drastis. Sehingga *cross validation* menjadi penting untuk memastikan performa algoritma pembelajaran mesin dapat dapat diandalkan (Kuhn and Johnson, 2019). Evaluasi dilakukan terhadap model *xgboost* yang diterapkan untuk memprediksi kecepatan kendaraaan secara real-time dan terhadap efisiensi komputasi model (Chicco *et al.*, 2021). Evaluasi menggunakan *K-fold Cross Validation* membagi data menjadi k bagian. Model akan dilakukan pelatihan sebanyak k kali, setiap menggunakan k-1 bagian sebagai data pelatihan dan satu bagian sebagai data pengujian hingga semua bagian bagian telah digunakan sebagai data pengujian. Hasil akhirnya adalah rata-rata dari seluruh evaluasi tiap bagian yang memberikan estimasi performa dari model (Uzir *et al.*, 2016).

Metrik evaluasi yang digunakan dalam regresi adalah *Mean Squared Error* (MSE) yang mengukur rata-rata kesalahan kuadarat dari prediksi, *Root Mean Squared Error* (RMSE) merupakan akar dari MSE yang memberikan satuan sama dengan variabel target, *Mean Absolute Error* (MAE) untuk mengukur rata-rata besar kesalahan prediksi tanap memperhatikan arah kesalahan nilai aktual, dan *R-Squared* (R^2) untuk mengukur proporsi variasi dari variabel target yang dijelaskan oleh model (Chicco *et al.*, 2021) didefinisikan pada persamaan berikut

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (11)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (12)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (13)$$

Selain akurasi, efisiensi komputasi menjadi sangat penting dalam evaluasi model dan metode tuning *hyperparameter* yang meliputi *training time* dan *inference time*.

3. METODE PENELITIAN

Penelitian ini menggunakan dataset simulasi yang akan dibuat untuk mempresentasikan skenario lalu lintas dalam berbagai kondisi lingkungan yang mencerminkan kondisi lalu lintas perkotaan. Penggunaan data simulasi ini bertujuan untuk memastikan kontrol penuh sehingga memungkinkan evaluasi performa algoritma secara objektif dan sistematis. Dataset terdiri dari 20.000 observasi dengan 12 fitur dan 10 kali pengulangan yang mencakup berbagai faktor lingkungan yang mempengaruhi kecepatan kendaraan, seperti suhu, curah hujan, kepadaan lalu lintas, dan sebagainya:

Tabel 1. Fitur Input

Fitur input	Keterangan
Suhu (C)	distribusi normal dengan rata-rata 25°C dan deviasi 5°C
Curah Hujan (mm)	distribusi *Weibull* yang dikalikan 40 dan dibatasi menggunakan np.clip antara 0 hingga 60 mm.
Kepadatan lalu lintas (skala 1-10)	distribusi uniform sebagai indikator kemacetan dari lancar hingga padat
Kondisi jalan (1-5)	diambil secara acak untuk mencerminkan variasi kualitas infrastruktur jalan dari sangat baik hingga rusak
Waktu (Jam)	disimulasikan dalam rentang 0-23 untuk menangkap efek waktu sibuk, seperti pagi (07-09) dan sore (16-18)
Jenis Kendaraan	kategori kendaraan seperti motor, mobil dan truk untuk menggambarkan tipe kendaraan terhadap kecepatan
Akhir pekan	ditentukan dari waktu, jika hari % 7 > 5 (sabtu/minggu) maka dikategorikan sebagai akhir pekan (1), dan hari kerja (0)
Visibilitas (0.3-1)	Nilai ditentukan berbasis curah hujan, jika hujan deras (curah_hujan > 30), maka visibilitas rendah (0.3), sedang (0.7), atau tinggi (1.0). Ini mencerminkan kondisi pandangan pengemudi
Jarak Simpang (km)	distribusi eksponensial (skala=2), mencerminkan jarak antar persimpangan jalan yang umumnya banyak pendek tapi bisa panjang secara acak
Usia Kendaraan (tahun)	Integer acak dari 0–20 untuk menunjukkan usia kendaraan, yang dapat memengaruhi performa mesin dan kecepatan
CC Mesin (cc)	Mobil memiliki cc sekitar 1500 (±300), motor sekitar 125 (±25), dan truk diset 5000. Ini mencerminkan kapasitas mesin sebagai indikator tenaga

Fitur Target yaitu Kecepatan kendaraan (km/jam) yaitu dihitung berdasarkan rumus simulasi yang menggabungkan sleuruh fitur input dan fungsi kombinasi non linear dari fitur input, termasuk menambahkan fitur noise gaussian untuk mencerminkan variabilitas alami dalam kondisi lalu lintas yang nyata.

Tahapan yang akan dilakukan dalam penelitian ini adalah :

1. Pembangkitan Data simulasi dengan distribusi yang telah ditentukan.
2. Preprocessing Data
3. Pembagian dataset menjadi data *train* 80% dan data *test* 20%
4. Pelatihan model baseline *XGBoost* dilakukan untuk model pertama dilatih dengan *hyperparameter default* sebagai baseline berdasarkan persamaan (1)
5. Optimasi *Hyperparameter*
    - *Grid Search*
      Akan digunakan untuk mencari kombinasi *hyperparameter* $\hat{\lambda}$ yang meminimalkan fungsi objektif pada persamaan (2) dengan loss per-observasi pada persamaan (3) melalui ruang pencarian parameter berikut

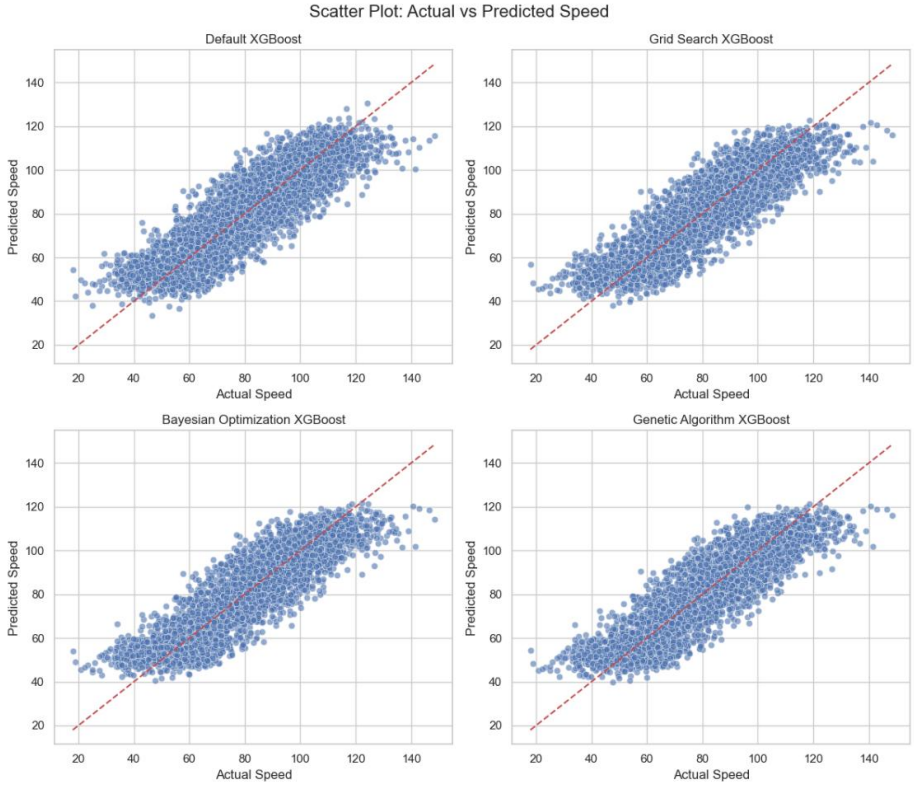
      Tabel 2. Ruang Parameter Tuning

Hyperparameter	**Ruang Parameter**
n_estimators	[100, 300, 500]
max_depth	[3, 5, 7, 11]
learning_rate	[0.01, 0.1, 0.2]
subsample	[0.5, 0.9, 1.0]
colsample_bytree	[0.01, 0.1, 1.0]
gamma	[0, 2]

    - *Bayesian Optimization*
      Menggunakan *gaussian procces regression* untuk menentukan *hyperparameter* optimal dengan persamaan (5) dan *likelihood* pada persamaan (6) ruang pencarian parameter yang digunakan seperti tabel 2.
    - *Genetic Algorithm* (GA)
      Akan digunakan untuk mencari kombinasi *hyperparameter* pada ruang pencarian tabel 2. Proses optimasi dilakukan dengan konfigurasi ukuran populasi 10 individu, crossover rate 0.5, mutation rate 0.2, dan jumlah generasi 10, di mana setiap individu merepresentasikan satu kombinasi hyperparameter dan dievaluasi menggunakan nilai *fitness* pada Persamaan (7).
6. Evaluasi Model
   Setelah diperoleh kombinasi *hypeparameter* terbaik dari masing-masing metode optimasi, evaluasi dilakukan menggunakan *Cross Validation* dengan 10-*fold* pada data *train* untunk mengestimasi performa model dan dievaluasi untuk pengujian akhir menggunakan data *test* berdasarkan aspek berikut :
    - Evaluasi model menggunakan MSE, RMSE, MAE dan *Rsquared* berdasarkan persamaan (10)(11)(12) dan (13).
    - Efisiensi komputasi dengan melihat *training time* dan *inference time*
    - *Trade-off* antara akurasi dan efisiensi dalam masing-masing metode tuning
7. Analisis hasil dan pembahasan

4. HASIL DAN PEMBAHASAN

Model Xgboost digunakan sebagai algoritma utama dalam memprediksi kecepatan kendaraan berdasarkan data simulasi. Diperoleh hasil prediksi dari setiap metode tuning hyperparameter yang telah dilakukan optimasi.



Gambar 2. *Scatter Plot* dari setiap metode

Berdasarkan gambar 2 diperoleh pada metode *default* XGBoost terlihat bahwa sebaran titik masih cukup menyebar di sekitar garis diagonal merah, tetapi masih banyak penyimpangan titik terutama pada rentang kecepatan menengah keatas dan menandakan bahwa model belum sepenuhnya mampu menangkap kompleksitas hubungan antar fitur dalam data. Pada metode *Grid Search* menunjukkan bahwa sebaran titik lebih terkonsentrasi dan terpusat di sekitar garis diagonal merah, yang menunjukkan bahwa metode ini mampu mempelajari pola secara optimal, sehingga memberikan prediksi yang mendekati nilai aktual dan kombinasi hyperparameter memberikan dampak yang signifikan.

Metode *Bayesian Optimization* menghasilkan pola yang cukup baik, dengan titik-titik banyak berada di sekitar garis diagonal. Namun, dibandingkan dengan *Grid Search*, masih terlihat sedikit penyimpangan pada kecepatan rendah dan tinggi, sehingga menunjukkan bahwa optimasi probabilistik ini belum sepenuhnya menemukan kombinasi parameter yang paling optimal. Pada metode *Genetic Algorithm* terlihat bahwa sebaran titik cukup konsisten mendekati garis diagonal merah dengan distrbusi yang lebih stabil di seluruh rentang kecepatan yang menunjukkan bahwa metode ini mampu memberikan keseimbangan yang baik antara akurasi prediksi dan generalisasi.

Tabel 3. Evaluasi *k-fold Cross-Validation*

Model	Fold	MSE	RMSE	MAE	R^2
Default	5	113.9871	10.6765	8.5042	0.7713
	10	112.6783	10.6150	8.4578	0.7680
Grid Search	5	104.9088	10.2425	8.1491	0.7895
	10	101.1157	10.0556	8.0081	0.7918
Bayesian Optimization	5	105.2732	10.2603	8.1502	0.7888
	10	101.2871	10.0641	8.0097	0.7915
Genetic Algorithm	5	105.4990	10.2713	8.1726	0.7884
	10	101.6390	10.0816	8.0181	0.7908

Pelatihan model dan tuning *hyperparameter* dilakukan menggunakan *k-fold cross-validation* untuk memperoleh estimasi performa yang lebih stabil dengan konfigurasi 5 dan 10 fold, sehingga dapat dibandingkan pengaruh jumlah *fold* terhadap akurasi dan kestabilan model. Kinerja model dievaluasi menggunakan metriks yang disajikan dalam Tabel 3.

Berdasarkan Tabel 3, peningkatan jumlah fold dari 5 menjadi 10 cenderung menghasilkan estimasi performa yang lebih stabil karena proporsi data latih pada setiap iterasi menjadi lebih besar dan variasi pembagian data lebih halus. Evaluasi utama penelitian ini menggunakan 10-*fold cross-validation*, dan nilai rata-rata metrik untuk masing-masing model dirangkum pada tabel 4 sebagai dasar perbandingan performa antar metode tuning yang disajikan dalam tabel berikut.

Tabel 4. Rata-rata evaluasi *Cross-Validation*

Model	Mean MSE	Mean RMSE	Mean MAE	Mean R^2
Default	113.6278	10.6582	8.4825	0.7677
Grid Search	105.2473	10.1341	8.0841	0.7856
Bayesian Optimization	102.5499	10.1589	8.1102	0.7911
Genetic Algorithm	102.8159	10.1585	8.1063	0.7906

Berdasarkan tabel 4. Nilai MSE, RMSE dan MAE yang lebih kecil menunjukkan kesalahan prediksi yang lebih rendah, sedangkan nilai R^2 yang lebih besar menunjukkan kemampuan model menjelaskan variasi data yang lebih baik. diperoleh bahwa setiap metode tuning memberikan pengaruh yang berbeda terhadap performa model XGBoost dalam memprediksi kecepatan kendaraan. metode *default* memperoleh nilai mean MSE sebesar 113.6278, MAE sebesar 8.4825 dan mean R^2 sebesar 0.7677 yang menunjukkan performa dasar tanpa adanya optimasi. *Grid Search* menghasilkan nilai mean MSE sebesar 105.2473 dan R^2 sebesar 0.7856. Hal ini menunjukkan bahwa *Grid Search* melalui kombinasi parameter yang ditentunkan mampu menghasilkan model lebih baik dalam menjelaskan variabilitas target dalam prediksi. Metode *Bayesian Optimazation* menghasilkan peningkatan performa dengan nilai MSE sebesar 102.5499 dan R2 sebesar 0.7911 yang menunjukkan bahwa pendekatan probabilistik dengan pencarian ruang parameter secara efisien dapat menemukan konfigurasi optimal dengan kinerja prediksi yang lebih unggul dibandingkan metode sebelumnya.

Metode *Genetic Algorithm* juga menghasilkan performa yang baik, dengan memperoleh *mean* MSE sebesar 102.8159 dan *mean* R^2 sebesar 0.7906, hampir sama dengan nilai yang diperoleh *Bayesian Optimization*. Metode ini lebih unggul dalam pencarian parameter yang fleksibilitas. Hasil dari semua metode menunjukkan bahwa penggunaan teknik tuning hyperparameter mampu meningkatkan performa model secara signifikan dibandingkan dengan metode *default.*

Tabel 5. Efisiensi Komputasi

Model	Training Time (s)	Inference Time (s)
Default	0.285	0.010
Grid Search	0.104	0.007
Bayesian Optimization	2.026	0.017
Genetic Algorithm	0.256	0.008

Berdasarkan tabel 5. diperoleh bahwa evaluasi efisiensi komputasi menunjukkan bahwa setiap metode memiliki karakteristik waktu komputasi yang berbeda untuk tahap training dan inference. Metode *default* membutuhkan waktu pelatihan sebesar 0.285 detik dan waktu prediksi 0.0010 detik dengan waktu yang singkat yang memberikan gambaran baseline efisiensi dari XGBoost tanpa tuning. Metode *Grid Search* tercatat memiliki waktu pelatihan

tercepat hanya 0.104 detik dan waktu prediksi 0.007 detik yang menunjukkan karena ruang pencarian yang terbatas dan hanya mengeksekusi pelatihan akhir dari hasil *grid search*.

Sebaliknya, *Bayesian Optimization* membutuhkan waktu pelatihan paling lama sebesar 2.026 detik dan waktu prediksi 0.017 detik yang menunjukkan bahwa proses optimasi dengan pendekatan probabilistik yang lebih kompleks dan memerlukan iterasi lebih baik untuk membangun model. Sementara itu, *Genetic Algorithm* membutuhkan waktu pelatihan yang cepat dengan 0.256 detik dan waktu prediksi 0.008 detik yang menunjukkan bahwa meskipun metode ini berbasis evolusi dan bersifat stokastik, dalam proses pencariannya tetap menghasilkan konfigurasi parameter dengan efisiensi waktu yang baik..

Tabel 6. *Trade-off* Akurasi dan Efisiensi

Model	Mean MSE	Mean R^2	Training Time (s)	Inference Time (s)
Default	113.6278	0.7677	0.285	0.010
Grid Search	105.2473	0.7856	0.104	0.007
Bayesian Optimization	102.5499	0.7911	2.026	0.017
Genetic Algorithm	102.8159	0.7906	0.256	0.008

Berdasarkan tabel 6 terlihat adanya *trade-off* yang signifikan antara akurasi prediksi dan efisiensi komputasi pada masing-masing metode. Temuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa performa XGBoost sangat sensitif terhadap pemilihan *hyperparameter* dan waktu komputasi yang cenderung meningkat ketika ruang pencarian semakin besar (Probst, Boulesteix, *et al.*, 2019). Namun, penlitian sebelumnya lebih memfokuskan pada peningkatan akurasi semata pada satu metode tuning dengan evaluasi metrik prediksi tanpa melakukan aspek waktu komputasi. Sehingga pada penelitian ini menggunakan metode tuning yang berbeda untuk dibandingkan dan menambahkan aspek waktu komputasi.

Hasil optimasi diperoleh bahwa model *Default* XGBoost tanpa optimasi menghasilkan performa prediksi yang paling rendah dengan MSE sebesar 114.0550 tetapi tetap cukup efisien dalam komputasi yang menunjukkan pentingnya penerapan tuning hyperparameter untuk meningkatkan kualitas prediksi. Diperoleh untuk *Grid Search* menghasilkan performa prediksi yang cukup baik dengan MSE sebesar 105.2473 dan memiliki waktu pelatihan tercepat sebesar 0.104 detik serta waktu pengujian sebesar 0.007 detik. Hal ini menunjukkan bahwa, meskipun metode ini menggunakan pendekatan ekshaustif dalam pencarian parameter, waktu komputasinya bisa sangat efisien tergantung pada ukuran ruang pencarian.

Sebaliknya *Bayesian Optimization* menghasilkan performa terbaik dengan nilai MSE terendah sebesar 102.5499 dan R^2 tertinggi sebesar 0.7911, yang menunjukkan bahwa pendekatan ini sangat efektif dalam menemukan kombinasi hyperparameter yang optimal. Namun, metode ini memiliki waktu pelatihan paling lama (2.026 detik) dan waktu inferensi tertinggi (0.017 detik) yang menandakan bahwa biaya komputasi yang diperlukan cukup tinggi. Metode *Genetic Algorithm* menunjukkan performa yang hampir setara dengan *Bayesian Optimization* dari segi akurasi MSE sebesar 102.8159 tetapi jauh lebih efisien dalam waktu prediksi 0.008 detik dan juga memiliki waktu pelatihan sebesar 0.256 detik yang menunjukkan bahwa menjadikan metode ini pilihan yang seimbang antara performa dan efisiensi.

Dengan demikian, penelitian ini memperjelas dalam perbandingan yang lebih komprehensif dalam model XGBoost dengan metode tuning *hyperparameter* yang berbeda untuk memprediksi kecepatan kendaraan. Pada penelitian selanjutnya diharapkan dapat memperluas kajian dengan menguji kestabilan model dengan metode optimasi lainnya dan menambahkan ruang pencarian agar pemilihan metode semakin luas dan kuat.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, bahwa prediksi kecepatan kendaraan menggunakan model XGBoost sangat dipengaruhi oleh metode optimasi *hyperparameter* yang digunakan. Hasil evaluasi performa dan akurasi model menunjukkan bahwa *Bayesian Optimization* terbukti mampu menghasilkan performa prediksi terbaik memberikan akurasi nilai MSE paling rendah dan nilai R^2 tertinggi dari metode lainnya, namun metode ini membutuhkan waktu komputasi yang jauh lebih lama. Metode *Genetic Algorithm* menunjukkan performa yang hampir sama dengan *Bayesian Optimization* dari segi akurasi, tetapi lebih efisien dalam waktu komputasi sehingga memberikan keseimbangan yang cukup baik antara akurasi dan efisiensi dengan waktu inferensi tercepat dan nilai prediksi yang cukup akurat.

Hasil dari penelitian memberikan gambaran mengenai *trade-off* antara akurasi dan efisiensi dari masing-masing metode tuning yang diharapkan dapat menjadi dasar bagi pengembangan sistem prediksi yang lebih optimal dan adaptif terutama dalam skenario lalu lintas real-time dengan penggunaan XGBoost yang telah dioptimalkan. Jika fokus utama adalah akurasi, maka *Bayesian Optimization* adalah pilihan terbaik. Jika mengutamakan efisiensi, *Grid Search* menjadi alternatif yang efektif. Namun, untuk mencari keseimbangan antara keduanya, *Genetic Algorithm* dapat dipertimbangkan sebagai pendekatan yang optimal. Dengan demikian, pemilihan metode tuning *hyperparameter* perlu disesuaikan dengan kebutuhan penelitian atau sistem untuk mencapai akurasi tinggi, efisiensi komputasi maupun keseimbangan keduanya dalam penerapan prediksi kecepatan kendaraan ini.

DAFTAR PUSTAKA

Chen, T. and Guestrin, C. (2016), "XGBoost: A scalable tree boosting system", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 13-17-Augu, pp. 785–794, doi: 10.1145/2939672.2939785.

Chicco, D., Warrens, M.J. and Jurman, G. (2021), "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation", *PeerJ Computer Science*, Vol. 7, pp. 1–24, doi: 10.7717/PEERJ-CS.623.

Fatihah, A.M., Dharmawan, K. and Swastika, P.V. (2024), *Implementation of X-Gradient Boosting in Banking Stock Price Predictions*, Atlantis Press International BV, doi: 10.2991/978-94-6463-413-6_17.

Huang, C., Zhu, X., Lu, M., Zhang, Y. and Yang, S. (2025), "XGBoost algorithm optimized by simulated annealing genetic algrithm for permeability prediction modeling of carbonate reservoirs", *Scientific Reports*, Vol. 15 No. 1, pp. 1–9, doi: 10.1038/s41598-025-99627-z.

J Cui, B.Y. (2018), "Survey on Bayesian optimization methodology and applications", *Journal of Software*, Vol. 29 No. 10, pp. 3068–3090, doi: http://dx.doi.org/10.13328/j.cnki.jos.005607.

James Bergsrtra, Y.B. (2012), "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, Vol. 13, pp. 281–305.

Kandasamy, K., Vysyaraju, K.R., Neiswanger, W., Paria, B., Collins, C.R., Schneider, J., Póczos, B., *et al.* (2020), "Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly", *Journal of Machine Learning Research*, Vol. 21, pp. 1–25.

Kuhn, M. and Johnson, K. (2019), *Feature Engineering and Selection*, *Feature Engineering and Selection*, CRC Press, doi: 10.1201/9781315108230.

Lambora, A., Gupta, K. and Chopra, K. (2019), "Genetic Algorithm- A Literature Review", *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, IEEE, No. 1998, pp. 380–384, doi: 10.1109/COMITCon.2019.8862255.

Martinez-Munoz, C.B.A.C. (2021), "A comparative analysis of gradient boosting algorithms", *Artifical Intelligence Review*, Vol. 54, pp. 1937–1967.

Mehdary, A., Chehri, A., Jakimi, A. and Saadane, R. (2024), "Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection", *Sensors*, Vol. 24 No. 4, doi: 10.3390/s24041230.

Probst, P., Boulesteix, A.-L. and Bischl, B. (2019), "Tunability : Importance of Hyperparameters of Machine Learning Algorithms", *Journal of Machine Learning Research 20*, Vol. 20 No. 53, pp. 1–32.

Probst, P., Wright, M.N. and Boulesteix, A.L. (2019), "Hyperparameters and tuning strategies for random forest", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 9 No. 3, pp. 1–15, doi: 10.1002/widm.1301.

Putatunda, S. and Rama, K. (2018), "A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost", *ACM International Conference Proceeding Series*, pp. 6–10, doi: 10.1145/3297067.3297080.

Rayadin, M.A., Musaruddin, M., Saputra, R.A. and Isnawaty, I. (2024), "Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki", *BIOS: Jurnal Teknologi Informasi Dan Rekayasa Komputer*, Vol. 5 No. 2, pp. 111–119.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P. and De Freitas, N. (2016), "Taking the human out of the loop: A review of Bayesian optimization", *Proceedings of the IEEE*, Vol. 104 No. 1, pp. 148–175, doi: 10.1109/JPROC.2015.2494218.

Tian, X., Zheng, Q., Yu, Z., Yang, M., Ding, Y., Elhanashi, A., Saponara, S., *et al.* (2023), "A Real-Time Vehicle Speed Prediction Method Based on a Lightweight Informer Driven by Big Temporal Data", *Big Data and Cognitive Computing*, Vol. 7 No. 3, doi: 10.3390/bdcc7030131.

Uzir, N., Raman, S., Banerjee, S. and Nishant Uzir Sunil R, R.S. (2016), "Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets Experimenting XGBoost Algorithm for Prediction and Classifi cation of Different Datasets", *International Journal of Control Theory and Applications*, Vol. 9 No. July.

Wang, D., Guo, H., Sun, Y., Liang, H., Li, A. and Guo, Y. (2024), "Prediction of Oil–Water Two-Phase Flow Patterns Based on Bayesian Optimisation of the XGBoost Algorithm", *Processes*, Vol. 12 No. 8, pp. 1–19, doi: 10.3390/pr12081660.