Prediksi Waktu Kedatangan Gempa Bumi Berdasarkan Seismogram Dengan Pembelajaran Mesin Berbasis Pohon

Andika Zidane Faturrahman
S1 Matematika
Institut Teknologi Bandung
Bandung, Indonesia
andikazidanef15@students.itb.ac.id

Muhammad Nabil Fadhlurrahman
S1 Matematika
Institut Teknologi Bandung
Bandung, Indonesia
abilfadhlurrahman@students.itb.ac.id

Muhammad Pudja Gemilang
S1 Matematika
Institut Teknologi Bandung
Bandung, Indonesia
pudja_gemilang@students.itb.ac.id

ABSTRAK

Gempa bumi adalah fenomena alam berupa pergerakan pada lempeng bumi secara tiba-tiba. Waktu menuju gempa bumi menjadi salah satu variabel terpenting yang dapat diprediksi karena potensinya dalam menyelamatkan sebagian besar penduduk melalui proses evakuasi dini. Waktu menuju gempa sulit diprediksi jika tidak mengandalkan metode pembelajaran mesin. Harapannya dengan menggunakan pembelajaran mesin dapat dihasilkan prediksi waktu gempa yang lebih akurat dibandingkan metode lainnya. Pada penelitian ini digunakan model pembelajaran mesin tree-based pada seismogram yang dihasilkan dari data eksperimen gempa bumi yang dilakukan oleh Los Alamos National Laboratory (LANL). Hasil prediksi menunjukkan bahwa model XGB dapat memprediksi waktu menuju gempa dengan skor MAE 2.14 dan mampu memberikan estimasi waktu gempa yang lebih cepat dari kejadian aktualnya pada 10 dari 13 kejadian gempa bumi. Kode dari penelitian ini dapat dilihat di laman GitHub https://github.com/abilcode/earthquake-prediction

Kata kunci—*Gempa bumi, waktu menuju gempa, pembelajaran mesin, seismogram*

I. PENDAHULUAN

1.1. Latar Belakang

Gempa bumi adalah fenomena alam berupa pergerakan pada lempeng bumi secara tiba-tiba yang disebabkan oleh pelepasan energi dari kerak bumi. Besar kekuatan gempa bumi dapat diukur dari magnitudo yang dihasilkan dari alat seismograf. Seismograf ini menghasilkan diagram sinyal seismik dari waktu ke waktu yang disebut sebagai seismogram. Seismogram dapat digunakan untuk mengukur besarnya sinyal seismik yang sedang terjadi di waktu tertentu dan terpenting magnitudo dari gempa bumi yang terjadi di wilayah sekitarnya. Namun, seismogram tidak dapat memprediksi kapan terjadinya gempa bumi karena hanya berfungsi sebagai detektor dari pergerakan lempeng bumi yang sedang terjadi.

Prediksi waktu gempa bumi menjadi tantangan yang masih belum bisa terpecahkan oleh berbagai peneliti di bidang kebumian. Sifat gempa bumi yang terjadi secara acak dapat membuat prediksi waktu menjadi kurang akurat. Namun, perkembangan teknologi kecerdasan buatan dalam bidang pembelajaran mesin. Prediksi berbasis pembelajaran mesin [1,2] dapat menjadi solusi sementara untuk memprediksi waktu terjadinya gempa bumi. Harapannya seiring waktu pemodelan berbasis pembelajaran mesin dapat menjadi lebih akurat atau dapat berperan sebagai deteksi dini dari terjadinya gempa bumi sehingga berpotensi untuk menyelamatkan sebagian besar masyarakat dari bencana gempa bumi. Model *tree based* pembelajaran mesin seperti *Decision Tree*, dapat menjadi solusi untuk kasus regresi seperti ini karena kemampuannya dalam mengatasi fitur yang banyak dan performa yang lebih baik untuk kasus-kasus nonlinier.

Penelitian ini ditujukan untuk mendesain model pembelajaran mesin dengan pendekatan kasus regresi pada data seismogram.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, diperoleh rumusan masalah sebagai berikut,

a. Bagaimana perbandingan performa antar model pembelajaran mesin dalam memprediksi waktu menuju gempa?
b. Apa fitur yang paling berkontribusi ke performa model pembelajaran mesin?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, diperoleh tujuan penelitian sebagai berikut,

a. Untuk menentukan perbandingan performa antar model pembelajaran mesin dalam memprediksi waktu menuju gempa
b. Untuk menentukan fitur yang paling berkontribusi ke performa model pembelajaran mesin

1.4. Batasan Masalah

Dari penelitian ini, terdapat batasan masalah yang ditetapkan yakni,

a. Prediksi sisa waktu gempa bumi terbatas dalam skala waktu detik dikarenakan data yang digunakan memiliki variabel target dengan skala waktu detik,
b. Prediksi sisa waktu gempa dapat dilakukan hanya pada saat setelah terjadinya gempa bumi pertama. Yakni, setelah gempa bumi pertama dapat dilakukan perhitungan sisa waktu gempa kedua.

II. TINJAUAN PUSTAKA

2.1. Pembelajaran mesin

Pembelajaran mesin adalah salah satu algoritma komputer yang didesain sedemikian rupa sehingga dapat

meniru kecerdasan manusia dalam hal mempelajari berbagai hal disekitar kita. Algoritma pembelajaran mesin memerlukan data masukan untuk menyelesaikan permasalahan pembelajaran mesin yang dipilih tanpa harus memberikan instruksi secara eksplisit. Algoritma tersebut akan beradaptasi sesuai kebutuhan pengembang secara otomatis melalui proses pengulangan data masukan secara iteratif sehingga didapatkan hasil yang sudah sesuai kebutuhan atau yang dikehendaki oleh pengembang. Pembelajaran mesin memiliki 2 tipe yaitu,

a. *Supervised learning*

Data yang digunakan dalam pemodelan adalah data yang mengandung variabel prediktor dan juga variabel target. Model pembelajaran mesin ditugaskan untuk mempelajari data tersebut untuk memprediksi variabel target. Supervised learning terdiri dari dua permasalahan utama yaitu masalah klasifikasi, yakni memprediksi variabel target yang bersifat kategorikal, atau masalah regresi, yakni memprediksi variabel target yang bersifat kontinu.

b. *Unsupervised learning*

Data yang digunakan dalam pemodelan adalah data yang hanya mengandung variabel prediktor/fitur. Model pembelajaran mesin ditugaskan untuk membuat kesimpulan seperti *clustering* pada data tersebut.

2.2. Regresi

Regresi adalah salah satu studi ilmu statistika dengan tujuan untuk menjelaskan dua atau lebih variabel dalam bentuk persamaan matematika dengan dua jenis variabel dalam model regresi, yaitu variabel bebas dan variabel terikat yang bernilai kontinu.

Variabel bebas (fitur atau variabel prediktor) dinotasikan sebagai $(x_{11}, x_{12}, \dots, x_{1n}, x_{21}, x_{22}, \dots, x_{3n} \dots, x_{m1}, x_{m2}, \dots x_{mn})$ dengan n adalah banyaknya variabel prediktor untuk memprediksi variabel terikat dan m adalah jumlah sampel data.

Variabel terikat (variabel target) dinotasikan sebagai y. Variabel terikat disebut tidak bebas karena nilainya dipengaruhi variabel lain, dalam hal ini dipengaruhi oleh variabel bebas.

2.3. Model Pembelajaran mesin

Decision tree adalah salah satu algoritma dalam pembelajaran mesin tipe *supervised learning* yang digunakan pada masalah klasifikasi dan regresi. Algoritma ini bekerja dengan memanfaatkan sifat – sifat dari struktur data pohon seperti *root node*, *edge*, *internal node*, *leaf node*, dan lain – lain untuk membuat seperangkat aturan yang akan bekerja berdasarkan keputusan tertentu untuk menghasilkan output berupa klasifikasi atau regresi [3].

Seiring dengan perkembangan penelitian di bidang pembelajaran mesin, *decision tree* memiliki beberapa turunan algoritma yang bekerja dengan menggunakan *ensemble learning*, yaitu sebagai berikut.

2.3.1. *Adaptive Boosting* (AdaBoost)

Adaptive Boosting adalah metode *boosting* yang bekerja berdasarkan *decision stump* (model yang hanya terdiri dari satu *decision tree*). Cara bekerjanya adalah dengan melatih beberapa algoritma secara sekuensial dengan melakukan *update* pada masing – masing bobotnya [4]. Bobot yang di*update* pada model pertama nantinya akan di*transfer* untuk digunakan pada model selanjutnya. Proses ini bekerja sampai nilai galat mendekati nol atau sampai banyaknya model yang telah didefinisikan sebelumnya.

2.3.2. *Histogram Gradient Boosting* (HGB)

Histogram Gradient Boosting adalah salah satu jenis *gradient boosting* yang bekerja menggunakan *ensemble learning* dengan mengkonstruksi setiap pohon melalui pembentukan *grouping* pada tiap pohon menjadi histogram.

2.3.3. *Light Gradient Boosted Machine* (LGBM)

Light Gradient Boosting Machine adalah salah satu jenis *gradient boosting* yang bekerja dengan melakukan *splitting* pada pohon berdasarkan daun [5]. Hal ini menyebabkan LGBM merupakan salah satu model yang sangat efektif dan cukup cepat dalam meminimumkan galat.

2.3.4. *Extreme Gradient Boosting* (XGB)

Extreme Gradient Boosting adalah salah satu jenis *gradient boosting* yang bekerja dengan meminimumkan fungsi objektif yang telah diregulerisasi [6]. Pada saat melakukan pelatihan, dilakukan penambahan pohon secara sekuensial untuk memprediksi *residual of error* dari pohon sebelumnya yang kemudian akan digabungkan dengan pohon saat ini untuk membuat prediksi final.

2.4. Signal Processing

Signal Processing adalah sub cabang dari teknik elektro yang berfokus pada proses menganalisis, memodifikasi dan mengekstrak informasi data sinyal seperti seismogram. Informasi yang diperoleh dapat menjadi fitur baru untuk dipelajari oleh model pembelajaran mesin sehingga dapat meningkatkan akurasi prediksi model terhadap variabel target. Adapun jenis informasi yang diperoleh melingkupi statistika deskriptif dari kumpulan segmen sinyal seperti rerata, median, kuartil, kurtosis, kemencengan, nilai maksimum dan minimum serta transformasi data sinyal sebagai berikut,

2.4.1. *Fast Fourier Transform* (FFT)

Fast Fourier Transform adalah algoritma yang digunakan untuk menghitung transformasi diskrit Fourier dari suatu sekuens sinyal $\gamma_1, \gamma_2, \dots, \gamma_{m-1}$ sebagai berikut.

$$X_k = \sum_{n=1}^{m-1} \gamma_n e^{i2\pi kn/N}, \qquad (1)$$

dengan $k = 0,1, \dots, N-1$. Tujuan dari adanya transformasi diskrit Fourier adalah untuk melihat representasi sinyal berdasarkan domain frekuensi. Algoritma ini dapat menghitung transformasi diskrit Fourier dengan jumlah operasi $O(Nlog_2 N)$ yang jauh lebih cepat dibandingkan dengan metode manual yang membutuhkan jumlah operasi $O(N^2)$

[7]. Idenya adalah dengan membagi komponen transformasi diskrit Fourier menjadi komponen genap dan ganjil lalu menghitung komponen genap dan ganjil secara bersamaan.

2.4.2. Transformasi Hilbert

Transformasi Hilbert ditujukan untuk menentukan sinyal analitik $\gamma_a(t)$ dari suatu sinyal $\gamma(t)$, yakni

$$\gamma_a(t) = F^{-1}(F(\gamma)2U) = x + iy \quad (2)$$

dengan F adalah transformasi Fourier, U adalah fungsi tangga satuan dan y adalah hasil transformasi Hilbert [8]. Dengan kata lain, transformasi Hilbert dari suatu sinyal adalah komponen imajiner dari sinyal analitiknya.

2.4.3. Hann *Window*

Hann *window* adalah *window* yang menggunakan fungsi kosinus untuk memperhalus sinyal dengan nilai akhirnya cenderung mendekati 0 [9,10]. Untuk $0 \leq n \leq M-1$, Hann window didefinisikan sebagai berikut,

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) \quad (3)$$

dengan M adalah jumlah output *window*,

2.4.4. *Short Time Average over Long Time Average* (STA-LTA)

STA-LTA adalah rasio antara rerata amplitudo pada sinyal dengan waktu jangka pendek dan sinyal dengan waktu jangka panjang.

2.4.5. *Moving Average* (MA),

Moving Average (MA) berorde m dapat didefinisikan sebagai berikut,

$$\hat{T}_t = \frac{1}{m}\sum_{j=-k}^{k} \gamma_{t+j} \quad (4)$$

dengan $m = 2k + 1$

2.4.6. *Moving Average* eksponensial,

Variasi dari MA yakni menggunakan faktor *smoothing* α dan bobot $w_i = (1-\alpha)^i$ sehingga

$$\hat{T}_t = \frac{\gamma_t + w_1\gamma_{t-1} + \cdots w_t\gamma_0}{1 + w_1 + \cdots w_t} \quad (5)$$

2.4.7. Rolling window

Rolling window digunakan untuk menghitung statistika deskriptif pada jangka waktu (*window*) tertentu agar dihasilkan sinyal yang lebih mulus dibandingkan yang aslinya.

III. METODE PENELITIAN

3.1. *Data Pemodelan*

Data yang digunakan pada penelitian ini adalah data yang didapat dari hasil eksperimen Los Alamos National Laboratory (LANL) [11] mengenai studi gempa pada batu yang diletakkan pada *bi-axial loading* dengan *double direct shear geometry*. Dua lapisan *gouge layers* digeser secara bersamaan sambil dikenai beban normal konstan dan kecepatan geser yang telah ditentukan. Patahan laboratorium gagal dalam siklus tongkat dan slip berulang ditujukan untuk meniru siklus pemuatan dan kegagalan pada patahan tektonik. Meskipun eksperimen ini jauh lebih sederhana daripada patahan di Bumi, eksperimen ini memiliki banyak karakteristik fisik yang sama.

Eksperimen ini menunjukkan bahwa prediksi gempa bumi laboratorium dari data seismik kontinu dimungkinkan dalam kasus siklus seismik laboratorium kuasi-periodik.

Selanjutnya, dari hasil eksperimen terdapat 1 data training dengan 629 juta baris dan 2 kolom, yakni *acoustic_data* dan *time_to_failure* serta 2626 file data test masing-masing berukuran 150000 baris dan 1 kolom, yakni *acoustic data*. Penjelasan dari setiap kolom adalah sebagai berikut,

a. *acoustic_data*
berisi sinyal seismic yang dihasilkan seismograf untuk menunjukkan besar vibrasi seperti angin, erupsi, gempa, dan lain – lain.

b. *time_to_failure*
berisi waktu yang tersisa (dalam detik) sebelum terjadinya gempa berskala laboratorium.

pada penelitian ini, pemodelan akan dilakukan dengan memandang variable *acoustic_data* sebagai variabel prediktor dan *time_to_failure* sebagai variable target.

3.2. *Analisis Data*

Berikut adalah visualisasi gabungan dari data akustik dan waktu yang tersisa sebelum gempa.

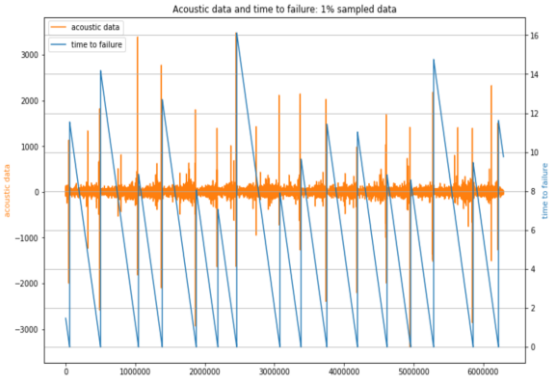


Fig. 1. Diagram visualisasi data akustik dan data waktu menuju gempa

dari visualisasi di atas dapat diinterpretasikan bahwa gempa terjadi saat nilai *time_to_failure* melompat dari nilai yang dekat dengan nol menjadi nilai yang lebih besar, akibatnya pada data ini terdapat 16 kejadian gempa. Selain itu dapat diperhatikan bahwa data akustik menunjukkan fenomena fluktuasi yang tajam sesaat sebelum terjadinya gempa. Adapun, diagram distribusi dari waktu menuju gempa bumi adalah sebagai berikut,

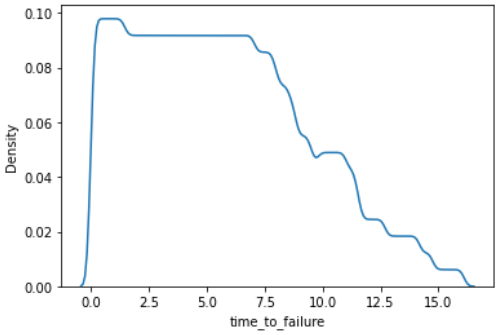


Fig. 2. Diagram distribusi dari variabel *time_to_failure*

di atas merupakan visualisasi distribusi dari waktu yang tersisa sebelum terjadinya gempa. Dapat dilihat bahwa

mayoritas dari data yaitu sekitar 10% dari data, memiliki nilai *time_to_failure* sebesar 0 sampai 2.5 detik.

3.3. *Feature Engineering*

Pada bagian ini akan dilakukan proses *feature engineering* pada data *training* dan data *test*. Prosedur yang akan dilakukan dalam proses ini menggunakan metodologi *signal processing* pada bab II dan referensi kode dari Kaggle [12]. Adapun, proses *feature engineering* ini melingkupi,

a. Membagi data *training* sinyal menjadi segmen data berukuran 150000 baris, ini ditujukan untuk mengikuti ukuran masing-masing file data test yakni 150000 baris,
b. Menentukan statistika deskriptif dari sinyal seismik pada setiap segmen data sebagai berikut,
    i. Rata-rata,
    ii. Standar deviasi,
    iii. Median,
    iv. Nilai maksimum,
    v. Nilai minimum,
    vi. Besar perubahan rata-rata antar segmen,
    vii. Rasio nilai maksimum ke nilai minimum,
    viii. Pejumlahan,
    ix. Kuantil,
    x. *Mean absolute deviation*
    xi. Kurtosis,
    xii. Kemencengan,
    xiii. *Inter Quartile Range* (IQR)
    xiv. Trend linier dari taksiran regresi
c. Melakukan *signal processing* pada data sinyal sebagai berikut,
    i. Transformasi Fourier dengan FFT untuk mengekstrak statistika deskriptif dari komponen real dan imajiner, magnitude dan sudut fasa dari hasil transformasi Fourier,
    ii. Transformasi Hilbert untuk mendapatkan rata-rata hasil transformasi tersebut,
    iii. Menggunakan Hann *window* untuk memperoleh rerata *smoothing* pada window 150,
    iv. Menentukan rerata STA-LTA dengan 8 segmen, yakni 500 – 10000, 5000 – 10000, 3333 – 6666, 10000 – 25000, 50 – 1000, 100 – 5000, 333 – 666,
    v. Menggunakan MA, *Moving Average* eskponensial dengan ukuran window 300, 400, 700, 1000, 3000, dan 30000
    vi. Menggunakan Rolling Window dengan ukuran window 10, 100 dan 1000 untuk menghitung standar deviasi dan rata-rata serta statistika deskriptif dari standar deviasi dan rata-rata tersebut.
d. Mengisi nilai kosong dari hasil transformasi dengan nilai rerata dari fitur yang bersangkutan,
e. Menstandarisasi nilai dari setiap fitur dengan menggunakan rumus berikut,

$$\tilde{\boldsymbol{x}}_{ij} = \frac{\boldsymbol{x}_{ij} - \mu_i}{s_i} \qquad (6)$$

Dengan $\boldsymbol{x}_{ij}$ adalah nilai segmen ke-j pada fitur ke-i, μ_i adalah rerata nilai pada fitur ke-i, dan s_i adalah standar deviasi nilai pada fitur ke-i,
f. Melakukan prosedur b-e pada data test.

Dari proses *feature engineering* ini dihasilkan data training berisi variabel prediktor berukuran 4194 x 164, data training berisi variabel target berukuran 4194 x 1, dan data test berisi variabel prediktor berukuran 2624 x 164 fitur. Selain itu, dihasilkan beberapa fitur prediktor yang memiliki korelasi tinggi antar fitur prediktor lainnya. Dalam kasus ini, tidak dilakukan pengurangan fitur prediktor karena model *tree* mampu menangani masalah fitur-fitur multikolinear.

3.4. Pemodelan

Dalam pembuatan model untuk mendeteksi gempa dini ini, kami menggunakan 5 model *machine learning* dan juga melakukan *hyperparameter tuning*. Berikut merupakan model pembelajaran mesin yang dipilih untuk permasalahan pendeteksian gempa dini ini

a. Decision Tree (DT) Regressor,
b. Histrogram Gradient Boosting (HGB) Regressor,
c. Adaptive Boosting (AdaBoost) Regressor,
d. Light GBM (LGBM) Regressor,
e. XGBoost (XGB) Regressor.

untuk setiap modelnya akan dilakukan *hyperparameter tuning* dengan melakukan *random search*. Pemilihan *hyperparameter tuning* tertera pada kode di laman GitHub.

Setiap model lalu divalidasi dengan menggunakan *Time Series Split* pada data *training* untuk menghindari kebocoran data. *Time Series Split* membagi data menjadi *k* fold dengan membagi data training dan data test pada interval waktu tertentu, indeks dari data test harus lebih tinggi dari indeks dari data training.

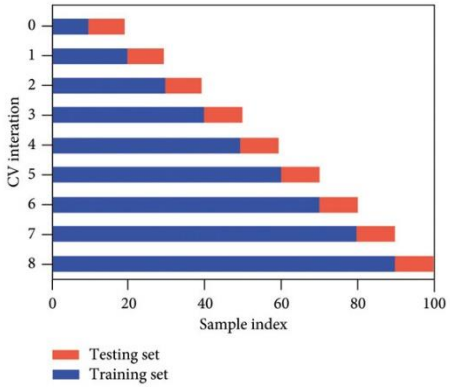
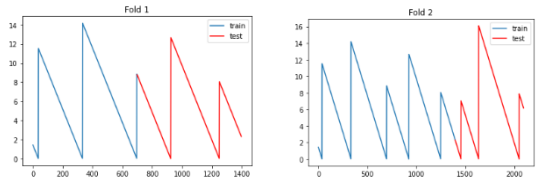


Fig. 3. Ilustrasi dari *Time Series Split*, sumber: [13]

3.5. Evaluasi Model

Masing-masing iterasi parameter dari *random search* tersebut akan dilakukan 5 split. Sebagai ilustrasi, berikut adalah 5 split pada data training y adalah sebagai berikut,



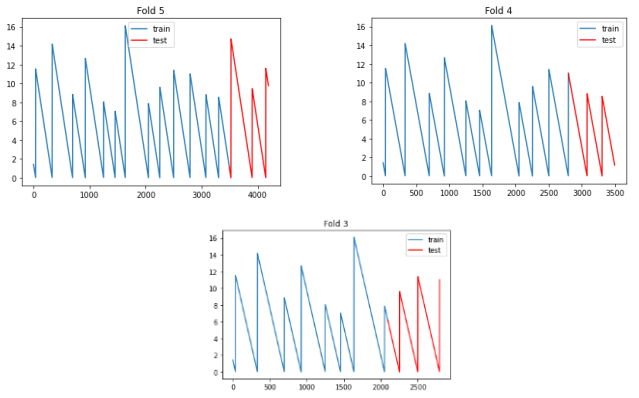


Fig. 4. Pembagian *time series split* pada data *training*

Dari model-model tersebut, akan dipilih parameter yang dapat memberikan *Mean Absolute Error* (MAE) yang minimum untuk setiap modelnya. Adapun, MAE didefinisikan sebagai berikut,

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

dengan y_i dan $\hat{y}_i$ adalah variabel target yang sebenarnya dan variabel target yang diprediksi dari model. Parameter terbaik dari setiap model kemudian dibandingkan dengan antar model yang digunakan sehingga diperoleh model yang terbaik dalam memprediksi variabel target

Selanjutnya, dibandingkan juga nilai prediksi waktu menuju gempa pada 1 indeks setelah kejadian gempa. Diketahui pada data *test* terdapat 13 kejadian gempa. Adapun, waktu prediksi gempa dikategorikan sebagai berikut

$$e = \begin{cases} \text{Awal, jika } y_t - \hat{y}_t > 0 \\ \text{Telat, jika } y_t - \hat{y}_t < 0 \end{cases} \quad (8)$$

dengan y_t dan $\hat{y}_t$ adalah nilai aktual dan prediksi waktu menuju gempa setelah 1 indeks dari kejadian gempa.

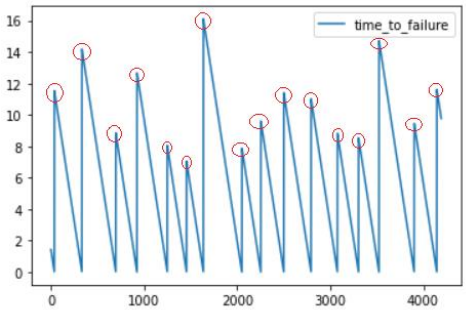


Fig. 5. Grafik variabel target pada data, lingkaran merah menandakan indeks awal setelah terjadinya gempa bumi sebelumnya dan awal waktu menuju gempa susulan.

IV. HASIL EKSPERIMEN

4.1. Parameter Model

Dari eksperimen yang telah dilakukan, diperoleh parameter terbaik dari setiap model sebagai berikut,

a. Decision Tree (DT)
   i. Criterion: Absolute error,
   ii. Splitter: Random,
   iii. Minimum weight fraction leaf: 0.1,
   iv. Minimum samples leaf: 10,
   v. Max leaf nodes: 10
   vi. Max features: Auto,
   vii. Max depth: 5
b. Adaptive Boosting (AdaBoost)
   i. Number of estimators: 100,
   ii. Learning rate: 0.05
c. Histogram Gradient Boosting (HGB)
   i. Loss: Absolute error,
   ii. Minimum samples leaf: 10
   iii. Maximum leaf nodes: 20
   iv. Maximum iteration: 100,
   v. Max depth: 1
d. Light Gradient Boosted Machine (LGBM)
   i. Bagging frequency: 69
   ii. Boosting: DART [14],
   iii. Column sample by tree: 0.75,
   iv. Drop rate: 0.1,
   v. Extra trees: True,
   vi. Lambda (L1): 0.4,
   vii. Lambda (L2): 0,
   viii. Learning Rate: 0.05,
   ix. Max Depth: 6,
   x. Number of estimators: 272,
   xi. Number of leaves: 59,
   xii. Subsample: 0.9
e. Extreme Gradient Boost (XGBoost)
   i. Subsample: 0.6,
   ii. Number of estimators: 100,
   iii. Minimum child weight: 5,
   iv. Max depth: 3,
   v. Learning rate: 0.015,
   vi. Lambda: 0.07,
   vii. Gamma: 0.075,
   viii. Column sample by tree: 0.8,
   ix. Alpha: 1

4.2. Prediksi dengan Time Series Fold

Setelah melakukan prediksi dengan 5 buah Time Series Fold, diperoleh hasil sebagai berikut,

TABLE I. TABEL SKOR MAE PADA SETIAP FOLD

Model	Waktu Eksekusi (s)	Mean Absolute Error					
		1	2	3	4	5	Total
DT	1.4	2.20	3.23	1.76	1.45	2.60	2.24
AdaBoost	15.232	2.16	3.22	1.82	1.44	2.64	2.25
HGB	**0.368**	2.20	**3.19**	1.72	1.34	**2.58**	2.21
LGBM	5.682	1.86	3.28	1.61	**1.23**	2.86	2.17
XGB	3.074	**1.76**	3.27	**1.60**	1.24	2.85	**2.14**

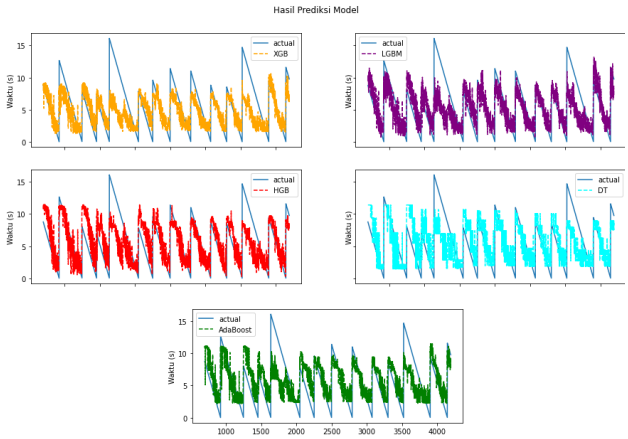


Fig. 6. Hasil prediksi dari kelima model

Dapat dilihat dari tabel di atas bahwa model XGB memiliki total MAE terkecil dibandingkan dengan model lainnya, diikuti oleh model LGBM dan HGB. Selain itu, pada fold ke-2, setiap model memiliki MAE di atas 3 dikarenakan terdapat pencilan dari waktu gempa selanjutnya yang mulai pada 15 detik.

Selanjutnya, akan dilihat dari galat waktu pemberitahuan gempa bumi.

TABLE II. TABEL PERBEDAAN WAKTU PREDIKSI GEMPA

Model	Waktu Prediksi	Frekuensi	Peluang Prediksi	Rata-rata Perbedaan Waktu (s)	Minimum Perbedaan Waktu (s)
DT	Awal	9	0.69	2.86	0.29
	Terlambat	4	0.31	-2.38	-3.66
AdaBoost	Awal	8	0.62	3.25	0.18
	Terlambat	5	0.39	-1.57	-2.88
HGB	Awal	8	0.62	3.18	0.34
	Terlambat	5	0.39	-1.55	-3.16
LGBM	Awal	9	0.69	3.57	0.63
	Terlambat	4	0.31	-1.14	-1.83
XGB	Awal	**10**	**0.77**	**4.14**	**0.71**
	Terlambat	**3**	**0.23**	**-0.58**	**-0.99**
Rerata 5 Model	Awal	9	0.69	3.27	0.30
	Terlambat	4	0.31	-1.39	-2.42

Dari tabel di atas, diperoleh bahwa model XGB dapat memprediksi terjadinya gempa lebih cepat pada 10 kasus gempa bumi dengan rerata waktu terawal yaitu 4.14 detik dan paling lambat 0.58 detik setelah gempa terjadi. Performa dari rerata 5 model juga dapat menyaingi model LGBM dalam memprediksi waktu gempa. Untuk kasus terburuk, model DT telat memprediksi waktu gempa selama 3.66 detik dari terjadinya gempa yang sebenarnya. Selanjutnya, akan ditinjau fitur-fitur terpenting dari prediksi model DT, AdaBoost, LGBM dan XGB

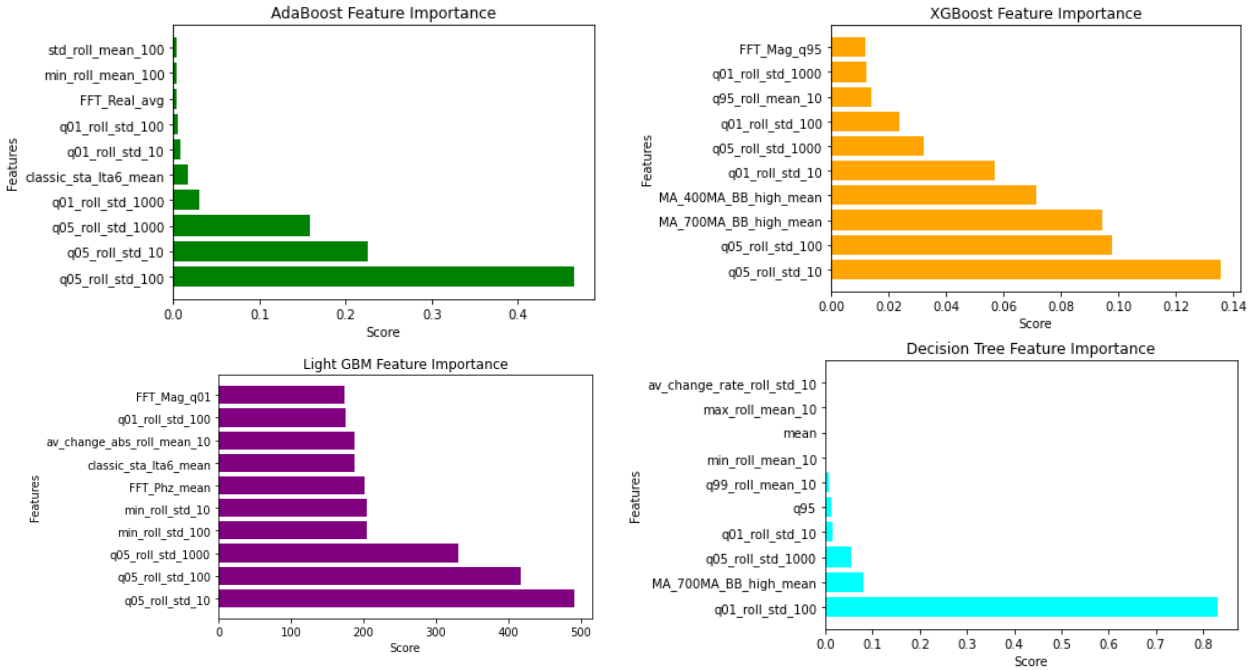


Fig. 7. 10 fitur terpenting dari prediksi model

Dari keempat diagram tersebut, fitur-fitur terpenting didominasi oleh fitur dari hasil rolling window 10, 100 dan 1000, diikuti oleh fitur MA, fitur transformasi Fourier terkait komponen real, magnitude, dan sudut fasa, fitur STA-LTA pada rentan (100 – 5000), serta statistika deskriptif seperti rata-rata dan kuantil.

4.3. Prediksi dengan laman Kaggle

Untuk menguji MAE dari prediksi 2 model terbaik, yakni XGB dan rerata 5 model, kami menggunakan kedua model tersebut untuk memprediksi data *test*. Skor private adalah skor dari prediksi 87% data *test* dan skor public adalah prediksi 13% sisanya dari data *test*. Berikut adalah skor MAE yang dihasilkan dari prediksi data *test*.

TABLE III. TABEL PREDIKSI LAMAN KAGGLE

Model	Skor Private	Skor Publik
XGB	2.98	1.68
Rerata 5 Model	**2.69**	**1.57**

Ternyata, model XGB memiliki skor private dan skor publik yang lebih tinggi dibandingkan rerata hasil prediksi dari 5 model. Lalu, skor private dan publiknya juga lebih rendah dari hasil training pada 5 fold data *train*. Hal ini berarti bahwa model XGB mengalami *overfitting* pada data training.

V. PENUTUP

5.1. Kesimpulan

Berdasarkan hasil eksperimen tersebut, diperoleh kesimpulan sebagai berikut,

a. Model XGB merupakan model terbaik dalam memprediksi waktu menuju gempa. Pada data *train*, diperoleh MAE 2.14 serta dapat memprediksi 10/13 waktu gempa yang lebih awal dibandingkan kejadian aktualnya. Namun, saat digunakan untuk memprediksi data *test* diperoleh skor MAE yang lebih tinggi dibandingkan hasil prediksi rata-rata 5 model
b. Fitur-fitur terpenting dalam memprediksi waktu gempa adalah fitur dari metode *rolling* window 10, 100, 1000, fitur MA, fitur transformasi Fourier terkait komponen real, magnitude, dan sudut fasa, fitur STA-LTA pada rentan (100 – 5000), serta statistika deskriptif seperti rata-rata dan kuantil.

5.2. Saran

Saran untuk penelitian selanjutnya adalah sebagai berikut,

a. Dapat dilakukan *feature selection* untuk mendapatkan model yang lebih efisien dari segi memori tetapi dapat mempertahankan skor MAE yang bagus,
b. Dapat digunakan model-model pembelajaran mesin lainnya seperti model linier dan model pembelajaran mendalam (*deep learning*) sebagai komparasi model

DAFTAR PUSTAKA

[1] Beroza, G. C., M. Segou, S. M. Mousavi. (2021). *Machine Learning and Earthquake Forecasting – Next Steps*, Nature Communications. 12, 4761.

[2] Mousavi, S. M., et al. (2020). *Earthquake Transformer - an attentive deep-learning model for simultaneous earthquake detection and phase picking*. Nature Communications. 11, 3952.

[3] Breiman, L., J. Friedman, R. Olshen, and C. Stone. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

[4] Freund, Y., and R. Schapire, (1997). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*.

[5] Ke, Guolin, et al. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Advances in Neural Information Processing Systems volume 30.

[6] Chen, T., Carlos, G. (2016). *XGBoost: A Scalable Tree Boosting System*. arXiv:1603.02754

[7] Brigham, E. O., R. E. Morrow. (1967). *The fast Fourier transform*, IEEE Spectrum, vol. 4, no.12, pp. 63-70,, doi: 10.1109/MSPEC.1967.5217220.

[8] Oppenheim, A.V., R. W. Schafer. (2009). *Discrete-Time Signal Processing, Third Edition*. Chapter 12. ISBN 13: 978-1292-02572-8

[9] Blackman, R.B., J.W. Tukey (1958). *The measurement of power spectra*, Dover Publications, New York.

[10] Kanasewich, E.R. (1975). *Time Sequence Analysis in Geophysics*, The University of Alberta Press, pp. 106-108.

[11] Howard, A., Bertrand RL, Laura P.N. (2019). *LANL Earthquake Prediction*. Kaggle. Retrieved from: https://kaggle.com/competitions/LANL-Earthquake-Prediction

[12] Lukayenko, A. (2019). Earthquakes FE. *More features and samples. Kaggle*. Retrieved from: https://www.kaggle.com/artgor/earthquakes-fe-more-features-and-samples

[13] Hasan, Afan, Kalipsiz, Oya, Akyokus, Selim. (2020). *Modeling Traders' Behavior with Deep Learning and Machine Learning Methods*: Evidence from BIST 100 Index. Complexity. 2020. 1-16. 10.1155/2020/8285149.

[14] Vinayak, R. K., Ran G. (2015). *DART: Dropouts meet Multiple Additive Regression Trees*. arXiv:1505.01866