Jurnal Informatika: Jurnal pengembangan IT

Vol. 11, No. 3, 2026
ISSN : 2477-5126, DOI:10.30591/jpit.v11i3.8

Toward Reliable Machine Learning Model Selection: A Standardized Multi-Metric Evaluation Framework For Loan Approval

Tri Agus Setiawan [1], Agus Ilyas[2], Hari Agung Budijanto[3], Mosses Aidjili[4], Tjoa Han Hwa[5]
[1]Program Studi Komputerisasi Akuntansi, Institut Widya Pratama, Indonesia
[2,3,4] Institut Widya Pratama, Indonesia
[5] Universitas Bina Nusantara, Indonesia
triaguss@iwima.ac.id, ilyasagus@iwima.ac.id, hariab40@gmail.com, mossesaidjili@gmail.com, tjoa.hwa@binus.ac.id

Info Artikel

Riwayat Artikel:

Received 04-08-2026
Revised 27-08-2026
Accepted 28-08-2026

Abstract – *The growing need for accurate, consistent, and reliable loan approval systems The use of machine learning in credit decision-making is increasingly important for financial institutions, but comparative research still often focuses on Accuracy or a limited number of classification metrics, so the trade-off between predictive performance and computational efficiency is not fully described. This study aims to develop a Standardized Multi-Metric Evaluation Framework (MMEF) to support the selection of more objective and reproducible machine learning models in the case of loan approval. The research method uses a standardized experimental pipeline with consistent preprocessing, class balancing using the Synthetic Minority Over-sampling Technique (SMOTE), identical data sharing, model optimization, and multi-metric evaluation. Five algorithms, namely Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, and CatBoost, are compared using a loan approval dataset consisting of 45,000 records and 13 predictor features. The evaluation includes Accuracy, Precision, Recall, F1-score, ROC-AUC, training time, and Overall Score. The results show that XGBoost provides the best overall performance with Accuracy 87.86%, Precision 66.77%, Recall 90.30%, F1-score 76.77%, ROC-AUC 96.27%, and Overall Score 0.827503. CatBoost has the fastest training time of 1.01 seconds, while SVM obtained the highest Recall of 92.40% with a much longer training time. These results indicate that model selection is not sufficient based on a single metric. MMEF provides a more systematic evaluation basis to identify models that have a balance of performance and efficiency in loan approval experiments.*

Keywords: *Classification; Comparative Analysis; Loan Approval; Machine Learning; Multi-Metric Evaluation*

Corresponding Author:
Tri Agus Setiawan
Email: triaguss@iwima.ac.id

Abstrak – *Meningkatnya kebutuhan terhadap sistem loan approval yang akurat, konsisten, dan reliabel mendorong pemanfaatan machine learning dalam pengambilan keputusan kredit. Namun, penelitian komparatif masih sering berfokus pada Accuracy atau sejumlah terbatas metrik klasifikasi, sehingga trade-off antara performa prediktif dan efisiensi komputasi belum tergambarkan secara menyeluruh. Penelitian ini bertujuan mengembangkan Standardized Multi-Metric Evaluation Framework (MMEF) untuk mendukung pemilihan model machine learning yang lebih objektif dan dapat direproduksi pada kasus loan approval. Metode penelitian menerapkan experimental pipeline terstandar melalui preprocessing yang konsisten, penyeimbangan kelas menggunakan Synthetic Minority Over-sampling Technique (SMOTE), pembagian data yang identik, optimasi model, dan evaluasi multi-metrik. Lima algoritma, yaitu Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, dan CatBoost, dibandingkan menggunakan dataset loan approval yang terdiri atas 45.000 record dan 13 fitur prediktor. Evaluasi mencakup Accuracy, Precision, Recall, F1-score, ROC-AUC, training time, dan Overall Score. Hasil menunjukkan bahwa XGBoost menghasilkan kinerja keseluruhan terbaik dengan Accuracy 87,86%, Precision 66,77%, Recall 90,30%, F1-score 76,77%, ROC-AUC 96,27%, dan Overall Score 0,827503. CatBoost memiliki training time tercepat sebesar 1,01 detik, sedangkan SVM memperoleh Recall tertinggi sebesar 92,40% dengan waktu pelatihan yang jauh lebih lama. Temuan tersebut menegaskan bahwa pemilihan model tidak memadai apabila hanya didasarkan pada satu metrik. MMEF memberikan dasar evaluasi yang lebih sistematis untuk mengidentifikasi model dengan keseimbangan performa dan efisiensi pada eksperimen loan approval.*

Kata Kunci*: Analisis Komparatif; Evaluasi Multi-Metrik; Klasifikasi; Pembelajaran Mesin; Persetujuan Pinjaman*

I. PENDAHULUAN

Loan approval merupakan proses strategis dalam lembaga keuangan karena menentukan kelayakan pengajuan pinjaman berdasarkan karakteristik pemohon, kemampuan pembayaran, dan tingkat risiko kredit. Kualitas keputusan tersebut berimplikasi pada kualitas portofolio, potensi kerugian, profitabilitas, dan keberlanjutan aktivitas pembiayaan. Perkembangan *machine learning* (ML) mendorong pergeseran penilaian kredit dari pendekatan statistik dan berbasis aturan menuju pendekatan berbasis data yang mampu mempelajari pola serta hubungan kompleks dari

data historis [1]–[4]. Penerapan ML berpotensi meningkatkan konsistensi, kecepatan, dan objektivitas pengambilan keputusan, terutama seiring meningkatnya jumlah dan kompleksitas data pengajuan.

Ketidaktepatan keputusan loan approval dapat menimbulkan dua konsekuensi utama. Persetujuan terhadap pemohon berisiko tinggi meningkatkan kemungkinan gagal bayar dan kerugian pembiayaan, sedangkan penolakan terhadap pemohon yang sebenarnya layak dapat mengurangi peluang penyaluran kredit dan potensi pendapatan. Kondisi tersebut menuntut sistem pendukung keputusan kredit yang mampu membedakan kelas Approved dan Rejected secara akurat sekaligus mempertahankan konsistensi dan efisiensi proses pengambilan keputusan [1],[2],[5]. Sejumlah algoritma supervised learning telah digunakan untuk tujuan tersebut, antara lain Logistic Regression, Support Vector Machine (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), dan CatBoost. Logistic Regression umum digunakan sebagai model pembanding karena sederhana dan relatif mudah diinterpretasikan, SVM mampu membentuk batas keputusan pada ruang fitur berdimensi tinggi, Random Forest menangkap hubungan nonlinier melalui ensemble learning, sedangkan XGBoost dan CatBoost menerapkan gradient boosting yang banyak digunakan pada data finansial terstruktur [2],[6]–[9],[20],[25],[26].

Karakteristik data pengajuan pinjaman turut menentukan kinerja model. Data loan approval dapat mencakup atribut demografis, pekerjaan, kondisi finansial, riwayat kredit, dan karakteristik pengajuan dalam bentuk numerik maupun kategorikal. Data tersebut juga berpotensi mengandung missing value, duplikasi, outlier, dan ketidakseimbangan kelas. Class imbalance dapat menyebabkan model lebih dominan mempelajari kelas mayoritas sehingga kemampuan mengenali kelas tertentu menjadi kurang optimal. Karena itu, penelitian terdahulu memanfaatkan teknik resampling, termasuk Synthetic Minority Over-sampling Technique (SMOTE), untuk meningkatkan representasi kelas minoritas dalam proses pelatihan [4],[5],[12]. Selain itu, hubungan nonlinier antara pendapatan, jumlah pinjaman, pekerjaan, riwayat kredit, dan karakteristik finansial lainnya memperkuat kebutuhan untuk membandingkan beberapa algoritma agar diperoleh model yang sesuai dengan karakteristik data [6],[23].

Penelitian komparatif telah membandingkan Logistic Regression, SVM, Random Forest, XGBoost, LightGBM, CatBoost, jaringan saraf, dan model ensemble [6]–[8],[10],[19],[24],[27]. Hasil penelitian menunjukkan bahwa model ensemble dan gradient boosting sering memberikan performa prediktif yang kompetitif, tetapi tidak ada algoritma yang secara konsisten unggul pada seluruh dataset. Karakteristik data, preprocessing, feature engineering, distribusi kelas, konfigurasi hyperparameter, dan strategi validasi dapat memengaruhi hasil komparasi [5]–[8]. Moscato et al. menekankan pentingnya benchmarking beberapa pendekatan ML menggunakan metrik yang saling melengkapi [6], sedangkan Gunnarsson et al. menunjukkan bahwa penerapan deep learning dalam credit scoring tetap memerlukan pembuktian empiris [7],[25]. Penelitian lain juga menyoroti penanganan class imbalance [4],[5], interpretabilitas [11],[12],[16],[20], serta pengembangan online dan adaptive credit scoring [10],[16]. Penelitian yang dilakukan menyatukan enam metrik evaluasi terstandarisasi termasuk efisiensi waktu komputasi kedalam pipeline pengujian yang seragam dengan pembobotan seimbang. Pendekatan yang dilakukan membedakan riset ini secara teknis dari studi Moscato et al. dan benchmarking sejenis yang hanya mengevaluasi metrik prediktif tunggal tanpa memperhitungkan *trade-off* operasional pada dataset *credit scoring*.

Persoalan evaluasi menjadi aspek penting dalam membandingkan performa model. Accuracy menunjukkan proporsi prediksi yang benar, tetapi belum cukup menggambarkan perilaku model secara menyeluruh, khususnya pada data dengan distribusi kelas yang tidak seimbang. Precision, Recall, F1-score, dan ROC-AUC memberikan perspektif tambahan mengenai ketepatan prediksi positif, kemampuan mengenali kelas positif, keseimbangan precision dan recall, serta kemampuan diskriminasi model [4],[6],[10]–[13]. Evaluasi pada data kredit menunjukkan bahwa Accuracy yang tinggi belum tentu mencerminkan kemampuan model yang baik dalam mengenali kelas yang menjadi perhatian [4],[11],[13]. Di sisi lain, performa prediktif yang tinggi juga belum menjadikan model sebagai pilihan operasional terbaik apabila membutuhkan waktu atau sumber daya komputasi yang lebih besar. Oleh karena itu, penelitian terkini mulai mempertimbangkan interpretabilitas, kebutuhan komputasi, fairness, dan konsekuensi ekonomi dalam evaluasi model [14]–[18]. Pendekatan Explainable Artificial Intelligence (XAI), termasuk SHAP, digunakan untuk menjelaskan kontribusi fitur terhadap prediksi [11],[12],[14],[16],[19],[20], sedangkan penelitian fairness menunjukkan bahwa performa prediktif yang tinggi tidak secara otomatis menjamin keadilan keputusan bagi kelompok pemohon yang berbeda [17].

Temuan tersebut menempatkan pemilihan model ML untuk loan approval sebagai persoalan multidimensional. Fokus pemilihan model tidak cukup diarahkan pada algoritma dengan Accuracy tertinggi, tetapi juga pada bagaimana berbagai algoritma dibandingkan secara objektif dalam kondisi eksperimen yang setara. Meskipun penelitian benchmarking telah membandingkan beberapa algoritma menggunakan sejumlah metrik [6], penelitian lain juga telah memperluas evaluasi pada aspek imbalance, interpretabilitas, fairness, dan nilai ekonomi [11],[12],[16]–[24]. Namun, berbagai dimensi tersebut masih sering dianalisis secara terpisah dengan prosedur preprocessing, feature engineering, optimasi, dan validasi yang berbeda. Perbedaan prosedur tersebut menyulitkan interpretasi hasil komparasi karena variasi performa tidak sepenuhnya merepresentasikan karakteristik algoritma.

Kondisi tersebut menunjukkan kebutuhan terhadap prosedur pemilihan model yang terstandarisasi dan dapat direproduksi, dengan menempatkan berbagai algoritma pada kondisi eksperimen yang sama dan mengevaluasinya menggunakan indikator yang saling melengkapi. Research gap penelitian ini terletak pada belum terintegrasinya

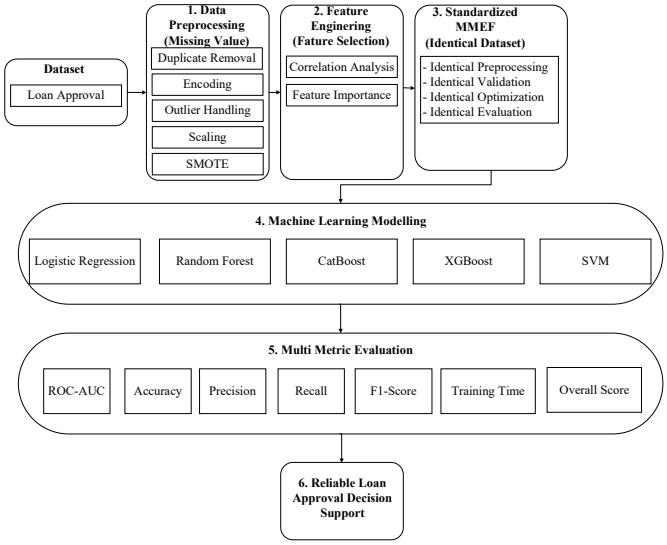
standardisasi experimental pipeline dan evaluasi multi-metrik secara konsisten dalam perbandingan beberapa algoritma loan approval. Penelitian ini menawarkan Standardized Multi-Metric Evaluation Framework (MMEF) sebagai pendekatan metodologis untuk mendukung pemilihan model ML secara lebih objektif. Kebaruan penelitian tidak terletak pada pengembangan algoritma klasifikasi baru, tetapi pada penyusunan experimental pipeline yang menstandarkan preprocessing, penyeimbangan kelas, pelatihan, optimasi model, validasi, evaluasi performa klasifikasi, dan pengukuran training time. Implementasi empiris MMEF menggunakan Accuracy, Precision, Recall, F1-score, ROC-AUC, training time, dan Overall Score. Adapun robustitas, XAI, fairness, signifikansi statistik, penggunaan memori, prediction latency, dan evaluasi berbasis biaya belum menjadi bagian dari hasil empiris penelitian ini dan diposisikan sebagai pengembangan MMEF pada penelitian selanjutnya.

Penelitian ini bertujuan mengembangkan dan mengevaluasi MMEF untuk mendukung pemilihan model ML pada sistem loan approval. Lima algoritma, yaitu Logistic Regression, SVM, Random Forest, XGBoost, dan CatBoost, dibandingkan menggunakan experimental pipeline yang terstandarisasi dan dievaluasi berdasarkan Accuracy, Precision, Recall, F1-score, ROC-AUC, training time, serta Overall Score. Hasil penelitian diharapkan memberikan dasar metodologis yang lebih objektif, sistematis, dan dapat direproduksi untuk mengidentifikasi model dengan keseimbangan performa prediktif dan efisiensi komputasi, sekaligus mendukung pengembangan sistem pendukung keputusan loan approval berbasis ML

II. METODE

Penelitian ini menggunakan pendekatan eksperimen kuantitatif untuk membandingkan performa algoritma machine learning pada permasalahan loan approval. Eksperimen menerapkan Standardized Experimental Pipeline, yaitu prosedur yang memberikan perlakuan identik kepada seluruh algoritma, mulai dari pemahaman data, preprocessing, persiapan fitur, pembagian data, penanganan class imbalance, pemodelan, optimasi hyperparameter, hingga evaluasi. Standardisasi ini diterapkan untuk meminimalkan pengaruh perbedaan prosedur eksperimen terhadap hasil komparasi antarmodel.

Tahapan penelitian mencakup pengumpulan dan pemahaman dataset, preprocessing, pembagian data, penanganan class imbalance, feature engineering, pemodelan menggunakan lima algoritma, optimasi hyperparameter, evaluasi melalui Standardized Multi-Metric Evaluation Framework (MMEF), serta pemeringkatan model. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian Standardized Multi-Metric Evaluation Framework untuk Loan Approval

A. Lingkungan Komputasi

Eksperimen diimplementasikan menggunakan Python 3.10.0 pada lingkungan Jupyter Notebook. Source code eksperimen menggunakan pustaka NumPy dan Pandas untuk pengolahan data, scikit-learn untuk preprocessing, pembagian data, pemodelan, dan evaluasi, imbalanced-learn untuk penerapan *Synthetic Minority Over-sampling Technique* (SMOTE), serta XGBoost dan CatBoost untuk implementasi algoritma boosting. Seluruh algoritma dijalankan melalui *experimental pipeline* yang sama agar dataset, preprocessing, pembagian data, penyeimbangan kelas, dan evaluasi tetap konsisten. Source code eksperimen tersedia dalam bentuk notebook Python yang mencakup beberapa skenario pengujian, yaitu Comparison1.ipynb, ComparisonSmote.ipynb, ComparisonPCA.ipynb, ComparisonSmotePCA.ipynb, dan ComparisonNonSmotePCA.ipynb.

B. Dataset

Dataset yang digunakan adalah Loan Approval Classification Dataset yang tersedia secara publik melalui Kaggle. Dataset terdiri atas 45.000 record, 13 atribut prediktor, dan satu variabel target, yaitu loan_status. Dataset merupakan versi sintetis yang dikembangkan berdasarkan Credit Risk Dataset dan diperkaya dengan atribut yang merepresentasikan karakteristik risiko finansial dalam proses loan approval. Dokumentasi dataset menyebutkan penggunaan SMOTENC dalam pembentukan data sintetis.

Ketiga belas atribut prediktor meliputi *person_age, person_gender, person_education, person_income, person_emp_exp, person_home_ownership, loan_amnt, loan_intent, loan_int_rate, loan_percent_income, cb_person_cred_hist_length, credit_score, dan previous_loan_defaults_on_file. Atribut tersebut merepresentasikan karakteristik demografis, pekerjaan, kondisi finansial, informasi pinjaman, dan riwayat kredit pemohon*.

Variabel loan_status digunakan sebagai target klasifikasi biner, dengan nilai 1 menunjukkan Approved dan 0 menunjukkan Rejected. Setiap record merepresentasikan satu pengajuan pinjaman sehingga dataset digunakan sebagai dasar pembentukan model klasifikasi keputusan loan approval.

C. Data Preprocessing

Preprocessing dilakukan secara identik terhadap seluruh algoritma untuk mempertahankan kesetaraan kondisi eksperimen. Tahapan ini mencakup pemeriksaan kualitas data, penanganan missing value, penghapusan data duplikat, transformasi variabel kategorikal, standardisasi fitur numerik, pemeriksaan outlier, serta penanganan class imbalance.

Nilai hilang pada atribut numerik ditangani menggunakan median imputation, sedangkan atribut kategorikal menggunakan mode imputation. Median dihitung berdasarkan distribusi nilai pada data pelatihan dan digunakan untuk menggantikan nilai yang hilang pada atribut numerik. Secara matematis, median dinyatakan sebagai:

$$\bar{x} = median(X) \tag{1}$$

sedangkan nilai modus digunakan untuk menggantikan nilai kosong pada atribut kategorikal.

Data duplikat diperiksa berdasarkan kesamaan observasi. Record yang teridentifikasi sebagai duplikat dihapus untuk mencegah observasi yang sama memberikan pengaruh berulang terhadap proses pembelajaran. Variabel kategorikal ditransformasikan ke dalam bentuk numerik menggunakan teknik *encoding*. Untuk kategori nominal, representasi *one-hot encoding* digunakan sehingga suatu kategori (c_j) direpresentasikan sebagai:

$$x_j = \begin{cases} 1, jika\ observasi\ termasuk\ kategori\ c_j \\ 0, jika\ lainnya \end{cases} \tag{2}$$

Fitur numerik kemudian distandardisasi menggunakan *z-score* untuk menyamakan skala antarfitur. Standardisasi dirumuskan sebagai:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{3}$$

dimana

- z_i = nilai fitur setelah standardisasi;
- x_i = nilai asli fitur;
- μ = nilai rata-rata (*mean*) fitur pada data pelatihan;
- σ = standar deviasi fitur pada data pelatihan.

Pemeriksaan *outlier* dilakukan menggunakan metode *Interquartile Range* (IQR), yaitu:

$$IQR = Q_3 - Q_1 \tag{4}$$

dimana

- IQR = Interquartile Range;
- Q1 = kuartil pertama (25th percentile);
- Q3 = kuartil ketiga (75th percentile).

Batas bawah dan batas atas ditentukan dengan:
Batas bawah outliner

$$LB = Q_1 - 1.5(IQR) \quad (5)$$

Batas atas outlier

$$UB = Q_3 + 1.5(IQR) \quad (6)$$

dimana
- LB = *Lower Bound*, yaitu batas bawah untuk identifikasi *outlier*;
- UB = *Upper Bound*, yaitu batas atas untuk identifikasi *outlier*;
- Q_1 = kuartil pertama;
- Q_3 = kuartil ketiga;
- IQR = *Interquartile Range*.

Nilai di luar batas tersebut diperiksa sebagai kandidat *outlier* dan ditangani menggunakan *capping* atau *Winsorization* apabila diperlukan.

D. Pembagian Data dan Class Balancing

Dataset dibagi menjadi 80% data pelatihan dan 20% data pengujian. Pembagian dilakukan sebelum proses SMOTE. Data pelatihan digunakan untuk pembelajaran dan optimasi model, sedangkan data pengujian dipertahankan sebagai data yang tidak mengalami proses *oversampling*.
Penanganan ketidakseimbangan kelas dilakukan menggunakan *Synthetic Minority Over-sampling Technique* (SMOTE) pada data pelatihan. Untuk observasi minoritas (x_i), data sintetis dibentuk berdasarkan observasi tersebut dan salah satu tetangga terdekatnya

$$x_{new} = x_i + \lambda(x_{nn} - x_i) \quad (7)$$

dimana
- x_{new} = data sintetis baru yang dihasilkan oleh SMOTE;
- x_i = observasi dari kelas minoritas;
- x_{nn} = salah satu *nearest neighbor* dari xix_ixi;
- λ = bilangan acak pada interval $0 \leq \lambda \leq 1$

dengan $0 \leq \lambda \leq 1$. SMOTE hanya diterapkan pada data pelatihan untuk menghindari *data leakage*, sedangkan data pengujian tetap menggunakan distribusi data yang tidak dimodifikasi. λ merupakan parameter interpolasi pada SMOTE yang menentukan posisi data sintetis x_{new} di antara observasi minoritas x_i dan *nearest neighbor* x_{nn}. Nilainya berada pada rentang 0 hingga 1.

E. Feature Engineering

Feature engineering dilakukan untuk menyiapkan representasi fitur yang digunakan dalam klasifikasi. Tahapan tersebut mencakup transformasi atribut kategorikal, standardisasi fitur numerik, dan penyusunan matriks fitur serta variabel target.
Matriks fitur dinyatakan sebagai:

$$X = [x_1, x_2, \ldots, x_p] \quad (8)$$

dengan (p) merupakan jumlah fitur prediktor. Variabel target dinyatakan sebagai:
- X = matriks atau himpunan fitur yang digunakan dalam proses pemodelan;
- $x_1, x_2, \ldots, x_p$ = fitur prediktor;
- p = jumlah fitur yang digunakan.

dengan (y=1) menunjukkan kelas *Approved* dan (y=0) menunjukkan kelas *Rejected*. Seluruh algoritma menggunakan representasi fitur yang sama untuk menjaga konsistensi komparasi.

F. Algoritma Machine Learning

Penelitian membandingkan lima algoritma, yaitu *Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, dan CatBoost*. Pemilihan algoritma merepresentasikan pendekatan pembelajaran yang berbeda, yaitu model linier, *kernel-based learning*, *bagging ensemble*, dan *gradient boosting*. Algoritma tersebut banyak digunakan pada penelitian *credit scoring* dan penilaian kredit [2],[5],[6],[8],[20],[26].
Logistic Regression digunakan sebagai *baseline* untuk klasifikasi biner. Probabilitas kelas positif dihitung menggunakan fungsi sigmoid:

$$P(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta^T x)}} \quad (9)$$

dimana
- P(y=1|x) = probabilitas observasi x termasuk kelas positif (*Approved*);
- y = variabel target;
- x= vektor fitur pengajuan pinjaman;
- β_0 = *intercept*;
- β = vektor koefisien model;
- $\beta^T x$ = hasil perkalian transpose koefisien dengan vektor fitur;
- e = bilangan Euler.

SVM digunakan untuk membangun batas keputusan antara kelas. Fungsi keputusan secara umum dapat dituliskan sebagai:

$$f(x) = \omega^T x \ + b \quad (10)$$

dimana
- f(x) = fungsi keputusan (decision function) SVM;
- ω = vektor bobot (weight vector);
- x = vektor fitur input;
- $w^T x$ = perkalian transpose vektor bobot dengan vektor fitur;
- b = bias/*intercept*.

Random Forest menggunakan sejumlah *decision tree* dan menggabungkan hasil prediksi melalui mekanisme agregasi. Secara umum, prediksi klasifikasi dapat ditentukan melalui:

$$\hat{y} = mode\{h_1(x), h_2(x), \ldots, h_T(x)\} \quad (11)$$

dimana
- $\hat{y}$ = hasil prediksi akhir Random Forest;
- ht(x) = prediksi dari decision tree ke-ttt;
- T = jumlah decision tree dalam Random Forest;
- mod = kelas yang paling sering muncul dari seluruh prediksi pohon.

XGBoost digunakan sebagai representasi algoritma *gradient boosting*. Fungsi objektifnya terdiri atas fungsi kerugian dan regularisasi:

$$\zeta(\theta) = \sum_{i=1}^{n} \iota(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (12)$$

dimana
- $\zeta(\theta)$= fungsi objektif XGBoost;
- $\iota(y_i, \hat{y}_i))$ = fungsi *loss* yang mengukur kesalahan prediksi;
- y_i = nilai aktual observasi ke-iii;
- $\hat{y}_i$ = hasil prediksi model untuk observasi ke-iii;
- f_kf_kfk = *decision tree* ke-kkk;
- $\Omega(f_k)$ = komponen regularisasi yang mengendalikan kompleksitas pohon;
- n = jumlah observasi;
- K = jumlah *decision tree*.

CatBoost digunakan sebagai alternatif *gradient boosting* untuk data tabular yang memiliki atribut kategorikal. Kelima algoritma dilatih menggunakan dataset dan prosedur *preprocessing* yang sama.

G. Hyperparameter Optimization

Optimasi *hyperparameter* dilakukan menggunakan *Bayesian Optimization*. Tujuan optimasi adalah memperoleh konfigurasi parameter yang memberikan performa terbaik dalam ruang parameter yang telah ditentukan. Secara umum, konfigurasi optimal dinyatakan sebagai:

$$\theta^* = \frac{arg\ max\ f(\theta)}{\theta \epsilon \Theta} \tag{13}$$

dimana

- θ^* = konfigurasi *hyperparameter* optimal;
- Θ = ruang pencarian *hyperparameter*;
- $f(\theta)$ = fungsi objektif yang digunakan untuk mengevaluasi konfigurasi θ;
- argmax = operator untuk memilih konfigurasi yang menghasilkan nilai fungsi objektif tertinggi.

Optimasi hyperparameter dilakukan menggunakan Bayesian Optimization untuk memperoleh konfigurasi terbaik pada setiap algoritma. Setiap model memiliki ruang pencarian hyperparameter yang sesuai dengan karakteristiknya dan dievaluasi menggunakan 5-Fold Stratified Cross-Validation. Data pengujian tidak digunakan selama proses optimasi, tetapi hanya digunakan untuk evaluasi akhir. Dengan prosedur yang sama, setiap algoritma memperoleh kesempatan optimasi yang setara sehingga hasil perbandingan lebih objektif dan dapat direproduksi. Adapun konfigurasi hyperparameter model terbaik yaitu model XGBoost seperti pada Tabel 1.

TABEL 1
KONFIGURASI HYPERPARAMETER MODEL

Model	Hyperparameter	Nilai yang digunakan
XGBoost	n_estimators	100
	max_depth	6
	learning_rate	0.10
	Subsample	0.80
	colsample_bytree	0.80
	random_state	42
	eval_metric	logloss

H. Validasi dan Standardized Experimental Pipeline

Berdasarkan Overall Score, XGBoost memperoleh Overall Score tertinggi sebesar 0,827503, sedangkan CatBoost berada pada posisi kedua dengan skor 0,820838. Selisih keduanya sebesar 0,006665 menunjukkan bahwa performa agregat kedua model relatif berdekatan. Karena penelitian menggunakan satu kali pembagian data 80:20 tanpa cross-validation, multiple runs, dan uji signifikansi statistik, hasil tersebut menunjukkan bahwa XGBoost memiliki skor agregat tertinggi pada kondisi eksperimen.

I. Standardized Multi-Metric Evaluation Framework (MMEF)

MMEF digunakan untuk mengevaluasi model berdasarkan beberapa indikator yang mencakup performa klasifikasi dan efisiensi komputasi. Pada implementasi empiris penelitian ini digunakan *Accuracy, Precision, Recall, F1-score, ROC-AUC, Training Time, dan Overall Score***.**
Accuracy dihitung menggunakan:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

Precision dihitung menggunakan:

$$Precision = \frac{TP}{TP+FP} \tag{15}$$

Recall dihitung menggunakan:

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

F1-score merupakan rata-rata harmonik Precision dan Recall:

$$F_1 = 2\,X\,\frac{Precision\ X\ Recall}{Precision + Recall} \tag{17}$$

ROC-AUC digunakan untuk mengukur kemampuan diskriminasi model terhadap dua kelas berdasarkan area di bawah kurva ROC:

$$AUC = \int_0^1 \mathrm{TPR(FPR)d(FPR)} \tag{18}$$

dimana
- AUC = *Area Under the Curve*;
- TPR = *True Positive Rate* atau *Recall*;
- FPR = *False Positive Rate*;
- d(FPR) = perubahan pada *False Positive Rate*.

Efisiensi pelatihan diukur menggunakan *Training Time*:

$$T_{train} = t_{end} - t_{start} \tag{19}$$

Nilai *Overall Score* digunakan untuk menggabungkan beberapa metrik setelah dilakukan normalisasi. Untuk metrik yang semakin tinggi semakin baik, normalisasi dilakukan dengan:

$$X_j^i = \frac{X_{ij} - X_i^{min}}{X_i^{max} - X_i^{min}} \tag{20}$$

dimana
- X_j^i = nilai fitur yang telah dinormalisasi;
- X_{ij} = nilai asli fitur ke-i pada observasi ke-j;
- X_i^{min} = nilai minimum fitur ke-i;
- X_i^{max} = nilai maksimum fitur ke-i.

Untuk metrik *cost* seperti *Training Time*, digunakan:

$$X_{ij}^1 = \frac{X_i^{max} - X_{ij}}{X_i^{max} - X_i^{min}} \tag{21}$$

dimana
- X_{ij}^1 = nilai matrik yang telah dinormalisasi;
- X_{ij} = nilai asli fitur ke-i pada model ke-j;
- X_i^{min} = nilai minimum matrik ke-i;
- X_i^{max} = nilai maksimum matrik ke-i.

Skor agregat model (j) dihitung menggunakan:

$$OS_j = \sum_{i=1}^{k} \omega_i X_{ij}^1 \tag{22}$$

dimana
- OS_j = Overall Score untuk model ke-j;
- ω_i = bobot untuk metrik ke-i;
- X_{ij}^1 = nilai metrik ke-i yang telah dinormalisasi untuk model ke-j;
- k = jumlah metrik yang digunakan dalam agregasi.

Untuk memastikan objektivitas dan kemampuan replikasi perhitungan skor agregat, penetapan bobot (ω_i) pada penelitian ini diberikan secara merata (*equal weighting*) untuk seluruh metrik evaluasi. Mengingat terdapat enam metrik utama yang diagregasi (k = 6), yaitu Accuracy, Precision, Recall, F1-score, ROC-AUC, dan Training Time, maka bobot masing-masing metrik ditetapkan sebesar 1/6 atau 0,1667. Rincian nilai bobot yang digunakan secara eksplisit pada eksperimen ini dapat dilihat pada Tabel 2.

TABEL 2
DISTRIBUSI BOBOT (ω_i) PADA MULTI-METRIC EVALUATION FRAMEWORK

Simbol	Metrik Evaluasi	Sifat Metrik	Nilai Bobot (ω)
ω_1	*Accuracy*	*Benefit* (Semakin tinggi semakin baik)	0,1667
ω_2	*Precision*	*Benefit* (Semakin tinggi semakin baik)	0,1667
ω_3	*Recall*	*Benefit* (Semakin tinggi semakin baik)	0,1667
ω_4	*F1-Score*	*Benefit* (Semakin tinggi semakin baik)	0,1667
ω_5	*ROC-AUC*	*Benefit* (Semakin tinggi semakin baik)	0,1667
ω_6	*Training Time*	*Cost* (Semakin rendah semakin baik)	0,1667
	Total Bobot (ω)		**1**

III. HASIL DAN PEMBAHASAN

A. Hasil Komparasi Performa Model

Eksperimen membandingkan *Logistic Regression, Support Vector Machine* (SVM), *Random Forest, XGBoost*, dan *CatBoost* pada klasifikasi *loan approval* menggunakan *Accuracy, Precision, Recall*, F1-score, ROC-AUC, *Training Time*, dan *Overall Score*. Hasil komparasi disajikan pada Tabel 3.

TABEL 3
HASIL KOMPARASI PERFORMA ALGORITMA

Algoritma	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Training Time	Overall Score
XGBoost	0,878556	0,667652	0,9030	0,767694	0,962667	2,848422	0,827503
Random Forest	0,871333	0,651004	0,9075	0,758145	0,957494	3,476398	0,820479
CatBoost	0,870444	0,648082	0,9125	0,757890	0,960104	1,010950	0,820838
SVM	0,865111	0,635052	0,9240	0,752749	0,954936	271,094463	0,817387
Logistic Regression	0,851444	0,610243	0,9175	0,732974	0,950985	2,379040	0,802675

Berdasarkan Tabel 3, XGBoost memperoleh *Overall Score* tertinggi sebesar 0,827503 dengan Accuracy 0,878556, Precision 0,667652, F1-score 0,767694, dan ROC-AUC 0,962667. Hasil tersebut menunjukkan bahwa XGBoost memberikan kombinasi performa klasifikasi paling kuat berdasarkan agregasi indikator dalam MMEF. Namun, keunggulan tersebut tidak berlaku pada seluruh indikator. SVM menghasilkan Recall tertinggi sebesar 0,9240, sedangkan CatBoost memiliki *Training Time* terendah sebesar 1,010950 detik. Random Forest menghasilkan Accuracy 0,871333 dan Recall 0,9075, sementara Logistic Regression memperoleh *Overall Score* terendah sebesar 0,802675. Perbedaan tersebut menunjukkan adanya karakteristik dan trade-off performa antarmodel sehingga pemilihan algoritma tidak cukup didasarkan pada satu indikator.

B. Analisis Performa Berdasarkan Multi-Metric Evaluation

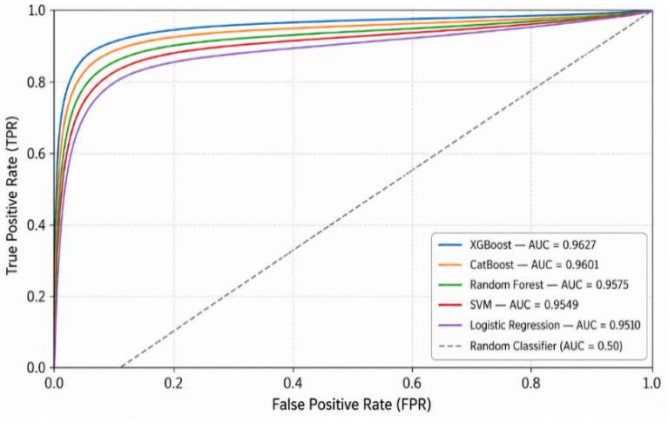
Untuk melihat pola performa kelima model secara simultan, digunakan performance heatmap yang memvisualisasikan Accuracy, Precision, Recall, F1-score, dan ROC-AUC sebagaimana ditunjukkan pada Gambar 2. Intensitas warna merepresentasikan besarnya nilai setiap metrik sehingga perbedaan performa antarmodel dapat diamati secara lebih cepat.

Gambar 2. Performance Heatmap Lima Model pada Dataset Loan Approval

Gambar 2 menunjukkan bahwa XGBoost memperoleh nilai tertinggi pada Accuracy, Precision, F1-score, dan ROC-AUC. Model ini menghasilkan Accuracy 0,8786, F1-score 0,7677, dan ROC-AUC 0,9627. SVM memiliki Recall tertinggi sebesar 0,9240, sedangkan CatBoost menunjukkan performa diskriminasi yang kompetitif dengan ROC-AUC sebesar 0,9601. Pola tersebut memperlihatkan bahwa tidak terdapat satu model yang unggul pada seluruh indikator, sehingga evaluasi multi-metrik diperlukan untuk memperoleh gambaran performa yang lebih komprehensif

C. Analisis ROC-AUC

Kemampuan model dalam membedakan kelas Approved dan Rejected dianalisis menggunakan ROC-AUC. Kurva ROC menggambarkan hubungan antara False Positive Rate (FPR) dan True Positive Rate (TPR), sedangkan AUC merepresentasikan kemampuan diskriminasi model secara keseluruhan. Hasil komparasi ROC Curve disajikan pada Gambar 3.



Gambar 3. ROC Curve Lima Model pada Dataset Loan Approval

Seluruh model menghasilkan AUC di atas 0,95. XGBoost memperoleh AUC tertinggi sebesar 0,9627, diikuti CatBoost sebesar 0,9601, Random Forest sebesar 0,9575, SVM sebesar 0,9549, dan Logistic Regression sebesar 0,9510. Hasil tersebut menunjukkan bahwa seluruh model memiliki kemampuan diskriminasi yang tinggi pada data pengujian.

Meskipun XGBoost memperoleh AUC tertinggi, perbedaan antarmodel relatif kecil. Selisih AUC antara XGBoost dan Logistic Regression hanya sebesar 0,0117. Dengan demikian, keunggulan XGBoost dalam pemeringkatan tidak semata-mata berasal dari kemampuan diskriminasi, tetapi merupakan konsekuensi dari kombinasi beberapa indikator yang digunakan dalam MMEF.

D. Analisis Efisiensi Komputasi

Training Time digunakan untuk melengkapi evaluasi performa klasifikasi dengan mempertimbangkan efisiensi komputasi. CatBoost menghasilkan waktu pelatihan tercepat sebesar 1,010950 detik, diikuti Logistic Regression sebesar 2,379040 detik, XGBoost sebesar 2,848422 detik, dan Random Forest sebesar 3,476398 detik. SVM membutuhkan waktu pelatihan paling lama, yaitu 271,094463 detik.

Hasil tersebut memperlihatkan trade-off antara performa prediktif dan efisiensi komputasi. XGBoost bukan model dengan waktu pelatihan tercepat, tetapi Training Time-nya masih jauh lebih rendah dibandingkan SVM. Sebaliknya, CatBoost menawarkan efisiensi komputasi terbaik, tetapi Overall Score sebesar 0,820838 masih berada di bawah XGBoost sebesar 0,827503. Temuan ini menunjukkan bahwa model dengan waktu pelatihan paling rendah tidak secara otomatis menghasilkan performa agregat terbaik. Kondisi tersebut memperkuat relevansi MMEF untuk mempertimbangkan beberapa dimensi performa secara simultan

E. Pemeringkatan Model

Pemeringkatan berdasarkan Overall Score disajikan pada Tabel 4. XGBoost menempati peringkat pertama dengan skor 0,827503, diikuti CatBoost sebesar 0,820838, Random Forest sebesar 0,820479, SVM sebesar 0,817387, dan Logistic Regression sebesar 0,802675.

TABEL 4
PEMERINGKATAN MODEL BERDASARKAN OVERALL SCORE

Peringkat	Algoritma	Overall Score
1	**XGBoost**	**0,827503**
2	CatBoost	0,820838
3	Random Forest	0,820479
4	SVM	0,817387
5	Logistic Regression	0,802675

Selisih Overall Score antara XGBoost dan CatBoost sebesar 0,006665 menunjukkan bahwa kedua model memiliki performa agregat yang relatif kompetitif. Oleh karena itu, hasil penelitian lebih tepat diinterpretasikan bahwa XGBoost memperoleh skor agregat tertinggi dalam kondisi eksperimen ini, bukan sebagai bukti bahwa XGBoost secara mutlak unggul pada seluruh aspek evaluasi.

F. Pembahasan Hasil Komparasi

Meskipun algoritma XGBoost menunjukkan performa agregat yang unggul, evaluasi spesifik menunjukkan nilai Precision yang relatif moderat (0,667652). Hal ini mengindikasikan bahwa dalam skenario prediksi bawaannya, model masih menghasilkan sejumlah *False Positive*. Sebagai implikasi praktis bagi pengguna sistem, kami sangat merekomendasikan penerapan *decision threshold tuning* sebagai strategi mitigasi. Secara default, fungsi klasifikasi biner memprediksi kelas positif jika probabilitas $P(\hat{Y}\}=1/X) \geq \tau$ dengan $\tau=0,5$. Untuk penelitian yang dilakukan di mana biaya akibat False Positive sangat merugikan, pengguna dapat mengkalibrasi ulang ambang batas tersebut menjadi $\tau > 0,5$\$. Peningkatan ambang batas ini akan memaksa model menjadi lebih konservatif, yang secara matematis akan menekan jumlah False Positive dan mendongkrak nilai Precision, meskipun kompensasinya dapat berupa sedikit penurunan pada sensitivitas (Recall).

SVM memperoleh Recall tertinggi sebesar 0,9240, yang menunjukkan kemampuan paling tinggi dalam mengenali kelas positif pada eksperimen ini. Keunggulan tersebut disertai Training Time sebesar 271,094463 detik, jauh lebih tinggi dibandingkan model lainnya. Temuan ini memperlihatkan bahwa peningkatan pada suatu indikator performa dapat disertai konsekuensi terhadap efisiensi komputasi.

CatBoost memiliki karakteristik berbeda dengan Training Time tercepat sebesar 1,010950 detik dan ROC-AUC 0,9601. Meskipun demikian, Overall Score CatBoost sebesar 0,820838 masih berada di bawah XGBoost. Random Forest juga menunjukkan performa kompetitif dengan Accuracy 0,871333 dan ROC-AUC 0,957494. Sementara itu, Logistic Regression memperoleh Overall Score terendah sebesar 0,802675, tetapi tetap relevan sebagai baseline karena merepresentasikan pendekatan linier untuk membandingkan performa model ensemble dan gradient boosting.

Secara keseluruhan, hasil tersebut sejalan dengan penelitian yang menunjukkan bahwa pendekatan ensemble dan boosting dapat memberikan performa kompetitif pada permasalahan credit scoring [6],[8],[16],[21],[24],[27], [28]. Namun, hasil penelitian ini tidak dapat digunakan untuk menyatakan bahwa XGBoost selalu menjadi algoritma terbaik pada seluruh permasalahan loan approval. Keunggulan XGBoost yang diperoleh terbatas pada dataset dan kondisi eksperimen yang digunakan dalam penelitian ini.

TABEL 5
PERBANDINGAN EVALUASI MULTI-METRIK (MMEF) DENGAN PENELITIAN TERDAHULU

Referensi	Pendekatan/Algoritma Terbaik yang Dilaporkan	Fokus Metrik Utama pada Literatur	Keterbatasan Evaluasi	Hasil Evaluasi pada MMEF
Moscato et al. (2021)	Machine Learning *benchmark* (*ensemble*)	*Accuracy*, metrik performa klasifikasi standar	Belum mengintegrasikan analisis biaya waktu (*training time*) ke dalam *scoring* agregat	Accuracy: 87,86%
Wu & Pan (2021)	Machine Learning Model	*Accuracy*, metrik prediksi dasar	Terbatas pada evaluasi performa prediktif	Recall: 90,30%
Hlongwane et al. (2021)	*Ensemble models* & XAI	Transparansi (*Explainability*), akurasi	Tidak mengevaluasi *trade-off* komputasi (waktu vs performa)	F1-Score: 76,77%
Yang et al. (2024)	*Hybrid Ensemble* (Soft Voting)	Performa prediktif (*Accuracy*)	Fokus pada optimasi bobot, namun belum terstandarisasi dalam *pipeline* MMEF	ROC-AUC: 96,27%
Machado & Karray (2022)	*Hybrid Machine Learning*	Akurasi untuk kelas risiko komersial	Tidak ada penyeimbangan multi-metrik secara eksplisit (*equal weighting*)	Training Time: 2,84 detik
Pratama & Ghozi (2026)	XGBoost (dengan *Threshold Tuning*)	*Accuracy*, evaluasi berbasis threshold	Konteks deteksi intrusi, bukan *loan approval* yang terstandarisasi	Overall Score (OS): 0,8275

G. Implikasi Hasil Penelitian

Hasil penelitian memiliki implikasi terhadap pemilihan model dan kebutuhan operasional. Berdasarkan kombinasi Accuracy, Precision, Recall, F1-score, ROC-AUC, Training Time, dan Overall Score, XGBoost menjadi model dengan performa agregat tertinggi pada eksperimen ini. Namun, kebutuhan operasional dapat menghasilkan pilihan model yang berbeda. SVM dapat dipertimbangkan apabila Recall menjadi prioritas, sedangkan CatBoost lebih sesuai ketika efisiensi waktu pelatihan menjadi pertimbangan utama.

Temuan tersebut mendukung penggunaan Standardized Multi-Metric Evaluation Framework karena pemilihan model tidak didasarkan pada satu ukuran performa. Standardisasi dataset, preprocessing, validasi, optimasi, dan evaluasi memberikan kondisi eksperimen yang konsisten sehingga perbedaan performa lebih dapat dikaitkan dengan karakteristik model yang dibandingkan.

Secara keseluruhan, XGBoost memperoleh *Overall Score* tertinggi sebesar 0,827503 dengan Accuracy 87,86%, F1-score 76,77%, dan ROC-AUC 96,27%. SVM menghasilkan Recall tertinggi sebesar 92,40%, sedangkan CatBoost memberikan *Training Time* tercepat sebesar 1,010950 detik. Hasil tersebut menegaskan adanya trade-off antara performa klasifikasi dan efisiensi komputasi. Dengan mempertimbangkan seluruh indikator MMEF, XGBoost menjadi model dengan performa agregat tertinggi dan direkomendasikan untuk kondisi eksperimen penelitian ini. Penetapan pembobotan merata (*equal weighting*) sebesar 1/6 (0,1667) pada keenam metrik evaluasi menjaga objektivitas pengukuran *Multi-Metric Evaluation Framework* (MMEF). Penambahan tabel perbandingan kuantitatif dengan literatur terdahulu membuktikan bahwa algoritma XGBoost menghasilkan keseimbangan optimal antara daya diskriminasi (ROC-AUC 96,27%) dan efisiensi waktu komputasi (2,84 detik). Sebagai mitigasi atas nilai Precision XGBoost (66,77%), penelitian ini merekomendasikan penyesuaian ambang batas keputusan (*decision threshold tuning* > 0,5) guna menekan tingkat False Positive. Sementara itu, ketiadaan uji signifikansi statistik atas selisih *Overall Score* yang kecil antara XGBoost (0,827503) dan CatBoost (0,820838) dicantumkan sebagai keterbatasan eksplisit pada bagian pembahasan dan simpulan.

IV. SIMPULAN

Penelitian ini menerapkan Standardized Multi-Metric Evaluation Framework (MMEF) untuk mendukung pemilihan model machine learning yang lebih objektif dan dapat direproduksi pada sistem loan approval. Menggunakan 45.000 record dan 13 fitur prediktor, lima algoritma dibandingkan melalui experimental pipeline terstandarisasi dengan pembagian data 80:20, penerapan SMOTE pada data pelatihan, serta evaluasi berdasarkan Accuracy, Precision, Recall, F1-score, ROC-AUC, training time, dan Overall Score. Hasil eksperimen menunjukkan bahwa XGBoost memperoleh Overall Score tertinggi sebesar 0,827503, dengan Accuracy 87,86%, Precision 66,77%, Recall 90,30%, F1-score 76,77%, dan ROC-AUC 96,27%. CatBoost menghasilkan training time tercepat sebesar 1,010950 detik, sedangkan SVM memperoleh Recall tertinggi sebesar 92,40% dengan waktu pelatihan yang jauh lebih besar. Temuan tersebut menegaskan bahwa pemilihan model tidak memadai jika hanya didasarkan pada Accuracy karena terdapat trade-off antarmetrik. XGBoost menunjukkan performa empiris paling kuat pada kondisi eksperimen ini, tetapi hasil tersebut tidak dimaksudkan sebagai keunggulan universal. Kontribusi utama penelitian terletak pada penyediaan prosedur evaluasi terstandarisasi yang dapat dikembangkan lebih lanjut melalui integrasi robustitas, XAI, fairness, evaluasi berbasis biaya, dan pendekatan lanjutan lainnya.

DAFTAR PUSTAKA

[1] S. Bhatore, L. Mohan, and Y. R. Reddy, "Machine learning techniques for credit risk evaluation: a systematic literature review," *Journal of Banking and Financial Technology 2020 4:1*, vol. 4, no. 1, pp. 111–138, Jun. 2020, doi: 10.1007/S42786-020-00020-3.

[2] X. Dastile, T. Celik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," *Appl. Soft Comput.*, vol. 91, p. 106263, Jun. 2020, doi: 10.1016/J.ASOC.2020.106263.

[3] P. Pławiak, M. Abdar, J. Pławiak, V. Makarenkov, and U. R. Acharya, "DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring," *Inf. Sci. (N. Y).*, vol. 516, pp. 401–418, Apr. 2020, doi: 10.1016/J.INS.2019.12.045.

[4] K. Niu, Z. Zhang, Y. Liu, and R. Li, "Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending," *Inf. Sci. (N. Y).*, vol. 536, pp. 120–134, Oct. 2020, doi: 10.1016/J.INS.2020.05.040.

[5] S. Shi, R. Tse, W. Luo, S. D'Addona, and G. Pau, "Machine learning-driven credit risk: a systemic review," *Neural Computing and Applications 2022 34:17*, vol. 34, no. 17, pp. 14327–14339, Jul. 2022, doi: 10.1007/S00521-022-07472-2.

[6] V. Moscato, A. Picariello, and G. Sperlí, "A benchmark of machine learning approaches for credit score prediction," *Expert Syst. Appl.*, vol. 165, p. 113986, Mar. 2021, doi: 10.1016/J.ESWA.2020.113986.

[7] B. R. Gunnarsson, S. vanden Broucke, B. Baesens, M. Óskarsdóttir, and W. Lemahieu, "Deep learning for credit scoring: Do or don't?," *Eur. J. Oper. Res.*, vol. 295, no. 1, pp. 292–305, Nov. 2021, doi: 10.1016/J.EJOR.2021.03.006.

[8] Y. Wu and Y. Pan, "Application Analysis of Credit Scoring of Financial Institutions Based on Machine Learning Model," *Complexity*, vol. 2021, no. 1, p. 9222617, Jan. 2021, doi: 10.1155/2021/9222617.

[9] A. Ashofteh and J. M. Bravo, "A conservative approach for online credit scoring," *Expert Syst. Appl.*, vol. 176, p. 114835, Aug. 2021, doi: 10.1016/J.ESWA.2021.114835.

[10] W. Yotsawat, P. Wattuya, and A. Srivihok, "A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble," *IEEE Access*, vol. 9, pp. 78521–78537, 2021, doi: 10.1109/ACCESS.2021.3083490.

[11] Y. Chen, R. Calabrese, and B. Martin-Barragan, "Interpretable machine learning for imbalanced credit scoring datasets," *Eur. J. Oper. Res.*, vol. 312, no. 1, pp. 357–372, Jan. 2024, doi: 10.1016/J.EJOR.2023.06.036.

[12] S. Han and H. Jung, "NATE: Non-pArameTric approach for Explainable credit scoring on imbalanced class," *PLoS One*, vol. 19, no. 12, p. e0316454, Dec. 2024, doi: 10.1371/JOURNAL.PONE.0316454.

[13] F. Shen, X. Zhao, G. Kou, and F. E. Alsaadi, "A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique," *Appl. Soft Comput.*, vol. 98, p. 106852, Jan. 2021, doi: 10.1016/J.ASOC.2020.106852.

[14] X. Dastile and T. Celik, "Making Deep Learning-Based Predictions for Credit Scoring Explainable," *IEEE Access*, vol. 9, pp. 50426–50440, 2021, doi: 10.1109/ACCESS.2021.3068854.

[15] J. G. F. Cabrera, H. A. P. Vicente, S. Maldonado, and J. Velasco, "Combination of unsupervised discretization methods for credit risk," *PLoS One*, vol. 18, no. 11, p. e0289130, Nov. 2023, doi: 10.1371/JOURNAL.PONE.0289130.

[16] R. Hlongwane, K. Ramaboa, and W. Mongwe, "Correction: A novel framework for enhancing transparency in credit scoring: Leveraging Shapley values for interpretable credit scorecards," *PLoS One*, vol. 20, no. 8, p. e0329901, Aug. 2025, doi: 10.1371/JOURNAL.PONE.0329901.

[17] N. Kozodoi, J. Jacob, and S. Lessmann, "Fairness in credit scoring: Assessment, implementation and profit implications," *Eur. J. Oper. Res.*, vol. 297, no. 3, pp. 1083–1094, Mar. 2022, doi: 10.1016/J.EJOR.2021.06.023.

[18] H. Mohammadnejad-Daryani, A. A. Taleizadeh, and D. Pamucar, "A novel profit-driven framework for model evaluation in credit scoring," *Eng. Appl. Artif. Intell.*, vol. 137, p. 109137, Nov. 2024, doi: 10.1016/J.ENGAPPAI.2024.109137.
[19] L. O. Hjelkrem and P. E. de Lange, "Explaining Deep Learning Models for Credit Scoring with SHAP: A Case Study Using Open Banking Data," *Journal of Risk and Financial Management 2023, Vol. 16, Page 221*, vol. 16, no. 4, p. 221, Apr. 2023, doi: 10.3390/JRFM16040221.
[20] Y. Liu, F. Huang, L. Ma, Q. Zeng, and J. Shi, "Credit scoring prediction leveraging interpretable ensemble learning," *J. Forecast.*, vol. 43, no. 2, pp. 286–308, Mar. 2024, doi: 10.1002/FOR.3033.
[21] R. Hlongwane, K. K. K. M. Ramaboa, and W. Mongwe, "Enhancing credit scoring accuracy with a comprehensive evaluation of alternative data," *PLoS One*, vol. 19, no. 5, p. e0303566, May 2024, doi: 10.1371/JOURNAL.PONE.0303566.
[22] J. Liu, X. Zhang, and H. Xiong, "Credit risk prediction based on causal machine learning: Bayesian network learning, default inference, and interpretation," *J. Forecast.*, vol. 43, no. 5, pp. 1625–1660, Aug. 2024, doi: 10.1002/FOR.3080;REQUESTEDJOURNAL:JOURNAL:1099131X;WGROUP:STRING:PUBLICATION.
[23] X. Tian, Z. Y. Tian, S. F. A. Khatib, and Y. Wang, "Machine learning in internet financial risk management: A systematic literature review," *PLoS One*, vol. 19, no. 4, p. e0300195, Apr. 2024, doi: 10.1371/JOURNAL.PONE.0300195.
[24] D. Yang, B. Xiao, M. Cao, and H. Shen, "A new hybrid credit scoring ensemble model with feature enhancement and soft voting weight optimization," *Expert Syst. Appl.*, vol. 238, p. 122101, Mar. 2024, doi: 10.1016/J.ESWA.2023.122101.
[25] Y. Hayashi, "Emerging Trends in Deep Learning for Credit Scoring: A Review," *Electronics 2022, Vol. 11, Page 3181*, vol. 11, no. 19, p. 3181, Oct. 2022, doi: 10.3390/ELECTRONICS11193181.
[26] Y. Jin, W. Zhang, X. Wu, Y. Liu, and Z. Hu, "A Novel Multi-Stage Ensemble Model with a Hybrid Genetic Algorithm for Credit Scoring on Imbalanced Data," *IEEE Access*, vol. 9, pp. 143593–143607, 2021, doi: 10.1109/ACCESS.2021.3120086.
[27] M. R. Machado and S. Karray, "Assessing credit risk of commercial customers using hybrid machine learning algorithms," *Expert Syst. Appl.*, vol. 200, p. 116889, Aug. 2022, doi: 10.1016/J.ESWA.2022.116889.
[28] V. N. Pratama and W. Ghozi, "Optimasi Model XGBoost Dengan Seleksi Fitur Mutual Information Dan Threshold Tuning Untuk Deteksi Intrusi Jaringan," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 11, no. 2, pp. 271–283, Apr. 2026, doi: 10.30591/JPIT.V11I2.10064.