SINTA Jurnal Sistem Informasi dan Teknologi Komputasi
Volume 3 Nomor 1 Januari 2026 [page: 36-44]
DOI: 10.61124/sinta.v3i1.145

EFEKTIVITAS XGBOOST LIGHTGBM DAN CATBOOST PADA DATASET IMBALANCED PREDICTIVE MAINTENANCE

Moeng Sakmar*[1], Nurul Tiara Kadir[2], Puteri Awaliatush Shofo[2], Agus Darmawan[2]
[1]Teknik Komputer, Universitas Jenderal Soedirman, Indonesia
[2]Informatika, Universitas Jenderal Soedirman, Indonesia
*e-mail: moeng.sakmar@unsoed.ac.id

Abstrak
Dalam era Industri 4.0, kegagalan mesin yang tidak terduga menjadi tantangan kritis yang memicu *unplanned downtime* serta kerugian finansial signifikan bagi sektor manufaktur. Kendala fundamental dalam pengembangan sistem *Predictive Maintenance* berbasis *Machine Learning* adalah ketidakseimbangan data *(class imbalance)*, di mana insiden kerusakan jauh lebih jarang terjadi dibandingkan kondisi normal, menyebabkan model cenderung bias dan gagal mengenali anomali vital. Penelitian ini bertujuan mengevaluasi efektivitas teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dalam mengoptimalkan kinerja deteksi kerusakan pada dataset AI4I 2020. Melalui pendekatan komparatif menggunakan tiga algoritma *Gradient Boosting* yaitu *XGBoost, LightGBM,* dan *CatBoost*. Penelitian ini menyoroti fenomena *Accuracy Paradox* pada skenario tanpa resampling yang menghailkan akurasi rata-rata 0.98, di mana tingginya akurasi semu menutupi ketidakmampuan model dalam mendeteksi kegagalan atau *Recall* yang rendah. Temuan penelitian ini menunjukkan bahwa integrasi SMOTE berhasil merekonstruksi batasan keputusan model, sehingga meningkatkan sensitivitas terhadap kelas minoritas secara signifikan dengan nilai *recall* rata-rata 0.82. Berdasarkan analisis mendalam melalui *Confusion Matrix*, algoritma *XGBoost* yang dikombinasikan dengan SMOTE teridentifikasi sebagai model paling optimal karena mampu menyeimbangkan *trade-off* kritis dengan mencatatkan *Recall* tinggi untuk menjamin keselamatan aset, sekaligus meminimalkan alarm palsu *(False Positive)* yang berdampak pada efisiensi kerja teknisi dibandingkan kompetitornya. Penelitian ini menyimpulkan bahwa penanganan ketimpangan data merupakan langkah deterministik untuk membangun sistem pemeliharaan prediktif yang tidak hanya presisi secara teknis, tetapi juga andal dan aman untuk diterapkan dalam ekosistem industri nyata.

Kata kunci: predictive maintenance; ketidakseimabangan kelas; SMOTE; machine learning.
.

Abstract
In the era of Industry 4.0, unexpected machine failures have become a critical challenge that triggers unplanned downtime and significant financial losses for the manufacturing sector. A fundamental obstacle in the development of Machine Learning-based Predictive Maintenance systems is data imbalance, where damage incidents occur much less frequently than normal conditions, causing models to become biased and fail to recognize vital anomalies. This study aims to evaluate the effectiveness of the Synthetic Minority Over-sampling Technique (SMOTE) in optimizing failure detection performance on the AI4I 2020 dataset. It uses a comparative approach with three Gradient Boosting algorithms: XGBoost, LightGBM, and CatBoost. This study highlights the Accuracy Paradox phenomenon in scenarios without resampling, where high spurious accuracy masks the model's inability to detect failures or low Recall. The findings of this study show that the integration of SMOTE successfully reconstructs the model's decision boundaries, thereby significantly increasing sensitivity to minority classes. Based on an in-depth analysis through the Confusion Matrix, the XGBoost algorithm combined with SMOTE was identified as the most optimal model because it was able to balance critical trade-offs by recording high Recall to ensure asset safety, while minimizing false alarms (False Positive) that impact technician work efficiency compared to its competitors. This study concludes that addressing data imbalance is a deterministic step in building a predictive maintenance system that is not only technically precise, but also reliable and safe for implementation in real industrial ecosystems.

Keywords: predictive maintenance; class imbalance; SMOTE; machine learning.

1. PENDAHULUAN

Transformasi digital dalam era Industri 4.0 telah menempatkan efisiensi operasional sebagai prioritas utama dalam keberlanjutan sektor manufaktur modern. Dalam ekosistem ini, strategi pemeliharaan mesin telah berevolusi dari pendekatan reaktif *(run-to-failure)* menjadi proaktif melalui penerapan *Predictive Maintenance* (PdM) [1] [2]. Studi terbaru menunjukkan bahwa kegagalan mesin yang tidak terprediksi masih menjadi penyumbang terbesar terhadap *downtime* produksi, yang secara signifikan menggerus profitabilitas perusahaan [3] [4]. Oleh karena itu, integrasi teknologi cerdas berbasis data menjadi krusial untuk memantau kondisi aset secara real-time, meminimalkan biaya perawatan, serta memperpanjang siklus hidup peralatan industri tanpa mengganggu jadwal produksi yang ketat [5].

Penerapan *Machine Learning* (ML) dan *Artificial Intelligence* (AI) kini menjadi tulang punggung dalam arsitektur PdM, menggantikan metode statistik konvensional yang memiliki keterbatasan dalam menangani data kompleks [6]. Berbagai penelitian telah mendemonstrasikan efektivitas algoritma cerdas dalam mengekstraksi pola tersembunyi *(hidden patterns)* dari data sensor seperti getaran, dan suhu. [7] [8], juga terkait torsi untuk memprediksi anomali sebelum kerusakan fatal terjadi [9]. Kemampuan *model machine learning* untuk memberikan peringatan dini memungkinkan para insinyur mengambil keputusan berbasis data, yang terbukti lebih akurat dibandingkan inspeksi manual maupun pemeliharaan berbasis jadwal tetap [10] [11].

Meskipun potensi penerapannya menjanjikan, tantangan mendasar dalam pengembangan model PdM yang reliabel adalah karakteristik data industri yang sangat tidak seimbang atau *imbalanced* [12]. Insiden kerusakan mesin merupakan peristiwa yang relatif jarang terjadi dibandingkan dengan kondisi operasional normal pada lingkungan dunia nyata [13] [14], sehingga rasio data antarkelas sering kali menunjukkan ketimpangan ekstrem [15] [16]. Kondisi ini menyebabkan algoritma klasifikasi standar cenderung bias ke arah kelas mayoritas, menghasilkan akurasi semu yang tinggi namun gagal mendeteksi kelas minoritas yang justru menjadi target utama prediksi [17]. Kegagalan deteksi ini, atau *False Negative*, membawa risiko operasional yang jauh lebih fatal dibandingkan kesalahan prediksi lainnya [18].

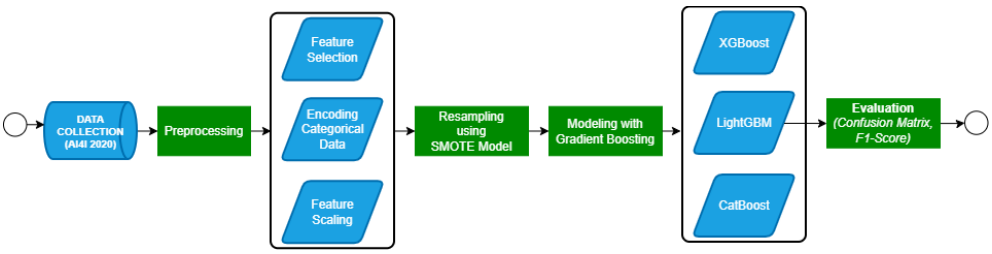
Guna mengatasi bias tersebut, diperlukan strategi komprehensif yang melibatkan teknik resampling data dan penggunaan algoritma yang *robust*. Teknik *Synthetic Minority Over-sampling Techniq*ue (SMOTE) telah divalidasi oleh berbagai literatur sebagai metode efektif untuk menyeimbangkan distribusi kelas dengan mensintesis sampel baru, sehingga batas keputusan *(decision boundary)* model menjadi lebih jelas [19] [20]. Di sisi lain, algoritma berbasis *Gradient Boosting* seperti *XGBoost, LightGBM*, dan *CatBoost* telah muncul sebagai standar industri untuk data tabular karena kemampuannya menangani hubungan non-linear yang kompleks dengan kecepatan komputasi yang superior dibandingkan model *Deep Learning* yang membutuhkan sumber daya besar [21] [22]. Kendati ketiga algoritma boosting tersebut mendominasi kompetisi data sains, penelitian yang membandingkan performanya secara *head-to-head* pada dataset *predictive maintenance* yang telah dioptimasi dengan SMOTE masih perlu diperkaya. Beberapa studi sebelumnya cenderung berfokus pada satu algoritma spesifik atau menggunakan metrik akurasi yang kurang representatif untuk kasus data tidak seimbang [23] [24].

Studi oleh Matzka [25], pada dataset AI4I berfokus membangun baseline klasifikasi menggunakan algoritma konvensional seperti *Random Forest*, namun cenderung mengabaikan dampak fatal ketidakseimbangan kelas yang menyebabkan model bias terhadap kondisi normal. Sementara penelitian lain menerapkan teknik *resampling*, mayoritas masih terpaku pada metrik akurasi global yang semu *(Accuracy Paradox)* tanpa mengeksplorasi kemampuan algoritma *ensemble modern* secara komparatif [26]. Penelitian ini mengisi kesenjangan tersebut, dengan menginvestigasi hasil integrasi SMOTE secara spesifik pada tiga arsitektur *Gradient Boosting* seperti, *XGBoost*, *LightGBM*, dan *CatBoost*, serta menawarkan perspektif evaluasi yang berorientasi industri, yaitu memprioritaskan ekuilibrium antara jaminan keselamatan aset *(High Recall)* dan efisiensi operasional *(Low False Positive)*, lalu menentukan model terbaik yang menawarkan keseimbangan optimal antara presisi deteksi kerusakan dan efisiensi waktu pelatihan.

2. METODE

2.1 Alur Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan sistematis untuk memastikan validitas hasil eksperimen. Kerangka kerja penelitian dimulai dari pengumpulan data hingga evaluasi performa model. Alur kerja disajikan pada Gambar 1. Seluruh proses pada penellitian dimulai dengan mengumpulkan dataset publik AI4I 2020 *Predictive Maintenance* yang diperoleh dari *UCI Machine Learning Repository*, lalu dilanjutkan dengan tahapan *preprocessing data*, *resampling* dengan SMOTE, pemodelan dengan *Gradient Boosting*, serta evaluasi menggunakan *Confusion Matrix,* dan *F1-Score*.



Gambar 1. Alur penelitian

DOI: 10.61124/sinta.v3i1.145

2.2 Dataset

Data yang digunakan pada penelitian ini adalah dataset publik AI4I 2020 *Predictive Maintenance* yang diperoleh dari *UCI Machine Learning Repository*. Dataset ini merupakan data sintetis yang mencerminkan peristiwa nyata pada industri manufaktur. Dataset terdiri dari 10.000 baris data dengan 14 fitur. Variabel target pada penelitian ini adalah kolom *Machine Failure* (label biner), di mana nilai '0' merepresentasikan kondisi normal dan nilai '1' merepresentasikan kegagalan mesin. Berdasarkan Tabel 1, menyajikan enam fitur utama yang diekstraksi sebagai variabel independen *(predictors)* yaitu *air temperature, process temperature, rotational speed, torque, tool wear, type* merepresentasikan parameter fisik krusial dalam operasional mesin CNC *(Computer Numerical Control)*. Pemilihan fitur ini didasarkan pada prinsip bahwa kegagalan mesin umumnya didahului oleh penyimpangan anomali pada parameter suhu, gaya, atau durasi penggunaan.

Tabel 1. Deskripsi fitur utama yang digunakan sebagai variable independent *(input)*

Index	Nama fitur (input)	Deskripsi fisik	Satuan	Tipe data
0	*Air temperature (K)*	Suhu udara ruangan sekitar mesin	*Kelvin (K)*	Numerik *(Float)*
1	*Process temperature (K)*	Suhu proses saat mesin beroperasi	*Kelvin (K)*	Numerik *(Float)*
2	*Rotational speed (rpm)*	Kecepatan putaran spindel	RPM	Numerik *(Integer)*
3	*Torque (Nm)*	Torsi atau gaya putar spindel	*Newton-Meter (Nm)*	Numerik *(Float)*
4	*Tool wear (min)*	Durasi waktu alat potong digunakan (keausan)	Menit	Numerik *(Integer)*
5	*Type*	Kualitas varian produk (L/M/H)	Kategorikal	*String/Object*

2.3 Preprocessing Data

Serangkaian teknik transformasi data diterapkan untuk meningkatkan integritas informasi, yang diawali dengan reduksi dimensi melalui penghapusan variabel identitas non-prediktif, seperti UDI dan *Product ID*, karena variabel tersebut tidak berkontribusi terhadap proses pemodelan dan berpotensi menimbulkan *noise*. Proses dilanjutkan dengan konversi variabel kategorikal menjadi representasi numerik menggunakan teknik *Label Encoding* agar dapat diproses secara matematis, serta penerapan normalisasi *Min-Max Scaling* pada seluruh fitur sensor fisik guna memetakan nilai ke dalam rentang seragam [0, 1]. Standarisasi skala ini merupakan langkah krusial untuk mencegah dominasi fitur dengan satuan *magnitude* besar terhadap fitur lainnya, sekaligus mempercepat laju konvergensi gradien *(gradient descent convergence)* pada algoritma berbasis *Boosting* selama fase pelatihan model.

2.4 Penangan *Imbalanced Data* dengan SMOTE

Pada dataset ditemukan rasio ketidakseimbangan yang ekstrem dimana kelas minoritas <5%, penelitian ini menerapkan *Synthetic Minority Over-sampling Technique* (SMOTE). Berbeda dengan *Random Oversampling* yang hanya menduplikasi data, SMOTE bekerja dengan mensintesis sampel buatan baru berdasarkan tetangga terdekat *(k-nearest neighbors)* dari data kelas minoritas. Secara matematis, sampel baru diporeleh melalui persamaan 1. Dimana x_i adalah sampel minoritas, x_{zi} adalah tetangga terdekat yang dipilih secara acak, dan λ adalah bilangan acak antara 0 dan 1. Proses ini dilakukan hanya pada *training set* untuk mencegah kebocoran data *(data leakage)* pada saat pengujian.

$$x_{new} = x_i + \lambda \text{ x } (x_{zi} - x_i) \quad (1)$$

2.5 Skenario Eksperimen

Pada tahap ini, disusun secara sistematis untuk menjamin validitas komparasi dan reproduktibilitas hasil penelitian melalui pendekatan *controlled environment*. Dengan menerapkan skema pemisahan data *stratified hold-out validation* (80:20), penelitian ini memastikan bahwa proporsi kelas minoritas pada data uji tetap representatif terhadap populasi aslinya, sehingga mencegah bias evaluasi yang sering muncul pada *random sampling* konvensional. Skenario pengujian dirancang secara bertingkat—mulai dari evaluasi baseline tanpa intervensi hingga penerapan teknik *Synthetic Minority Over-sampling Technique* (SMOTE)—guna mengisolasi dan mengukur dampak spesifik penyeimbangan data terhadap performa algoritma XGBoost, LightGBM, dan CatBoost. Pendekatan ini memungkinkan analisis mendalam tidak hanya pada kemampuan prediksi

global, tetapi juga pada ketahanan (robustness) model dalam mengatasi dilema trade-off antara presisi dan sensitivitas (Recall) yang krusial bagi sistem pemeliharaan prediktif.

2.6 Evaluasi Kinerja

Evaluasi difokuskan pada *Confusion Matrix*, dengan matrix utama yang digunakan adalah *precission* untuk mengukur ketepatan prediksi positif pada persamaan 2. Juga digunakan *Recall* untuk mengukur kemampuan model menemukan seluruh data kerusakan pada dengan persamaan 3. Matrix ketiga adalah *F1-score*, untuk menghitung rata-rata harmonik antara *Precision* dan *Recall*, memberikan gambaran performa menyeluruh pada data timpang menggunakan persamaan 3.

$$Precission = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2x\frac{Precission\ x\ Recall}{Precission+Recall} \quad (4)$$

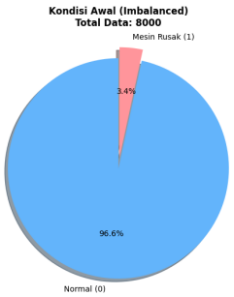
3. HASIL PENELITIAN

A. Lingkungan Eksperimen

Seluruh rangkaian eksperimen komputasi pada penelitian ini dieksekusi menggunakan lingkungan pengembangan berbasis *cloud*, *Google Colaboratory* (Colab). Pemilihan *platform* ini didasarkan pada kebutuhan akan sumber daya komputasi yang stabil dan terstandarisasi, guna mengeliminasi *bias* yang mungkin timbul akibat keterbatasan perangkat keras lokal. Ekosistem utama dibangun menggunakan bahasa pemrograman Python 3.10, yang didukung oleh pustaka manipulasi data *Pandas* dan *NumPy* untuk *reprocessing* yang efisien. Selain itu, spesifikasi lingkungan komputasi pendukung meliputi RAM 12GB, CPU Intel Core i7, dan VGA NVIDIA GeForce RTX 5060 8 GB. Secara spesifik, implementasi algoritma pembelajaran mesin didukung oleh *Scikit-learn* untuk evaluasi metrik, serta pustaka khusus *Imbalanced-learn* untuk eksekusi teknik *resampling SMOTE* yang menjadi fokus intervensi data. Adapun untuk pemodelan prediktif tingkat lanjut, penelitian ini mengintegrasikan kerangka kerja terdedikasi dari *XGBoost*, *LightGBM,* dan *CatBoost*. Seluruh visualisasi hasil analisis, termasuk *Confusion Matrix* dan distribusi data, direpresentasikan menggunakan *Matplotlib* dan *Seaborn* untuk menjamin interpretasi grafis yang akurat dan komunikatif. Konfigurasi lingkungan ini didokumentasikan secara rinci untuk menjamin reproduktifitas penuh dari temuan penelitian.

B. Preprocessing

Kualitas performa model *machine learning* memiliki ketergantungan linear terhadap kualitas data input. Oleh karena itu, tahapan pra-pemrosesan penelitian ini dijalankan bukan sekadar sebagai prosedur pembersihan teknis, melainkan sebagai intervensi strategis untuk mengeliminasi *noise*, menerjemahkan informasi semantik menjadi numerik, dan menstandarisasi variabilitas fitur. Sebelum masuk ke tahap pelatihan model, analisis awal dilakukan terhadap distribusi kelas pada dataset seperti disajikan pada Gambar 2.

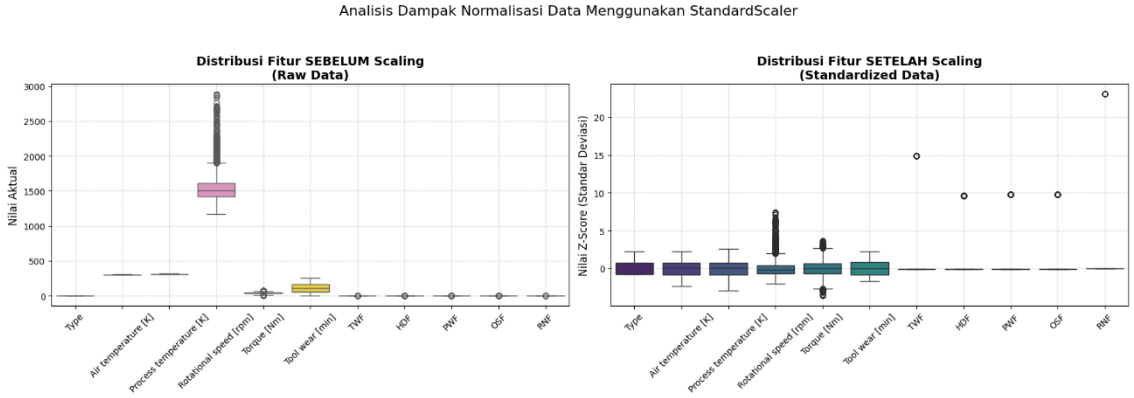


Gambar 2. Distribusi kelas sebelum SMOTE

dataset AI4I 2020 memiliki ketimpangan yang ekstrem, di mana kelas 'Normal' mendominasi sebesar 96,6% dan kelas 'Rusak' hanya 3,4%. Jika model dilatih menggunakan data mentah ini, model akan mengalami bias mayoritas. Penerapan teknik SMOTE *(Synthetic Minority Over-sampling Technique)* pada data latih dapat

mengubah struktur distribusi data. SMOTE bertugas mensintesis sampel baru pada kelas minoritas sehingga rasio antara kelas normal dan rusak diharapkan menjadi seimbang.

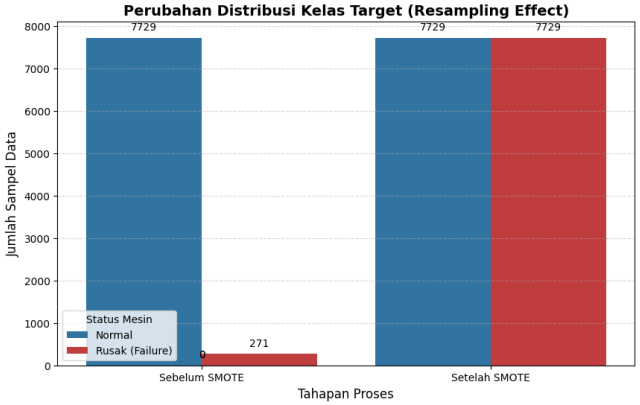
Gambar 3 menyajikan tantangan fundamental dalam dataset, yaitu disparitas magnitudo yang ekstrem antar-fitur. Terlihat bahwa fitur *Rotational Speed* memiliki rentang nilai ribuan (rpm), yang secara visual menekan distribusi fitur lain seperti *Torque* dan *Temperature* hingga tampak mendatar. Jika kondisi ini dibiarkan, algoritma optimasi model akan mengalami bias bobot, di mana perubahan kecil pada fitur berskala besar akan dianggap jauh lebih signifikan dibandingkan perubahan pada fitur berskala kecil. Penerapan *StandardScaler* terbukti efektif mengatasi masalah ini, sebagaimana ditampilkan pada Gambar 3. Seluruh fitur telah ditransformasi ke dalam skala unit standar $\mu=0$ dan deviasi standar $\alpha=1$. Homogenisasi skala ini menjamin bahwa setiap variabel sensor memiliki kontribusi yang setara dalam pembentukan ruang fitur, sekaligus mencegah dominasi numerik yang dapat mendistorsi perhitungan jarak *Euclidean* pada algoritma berbasis gradien dan tetangga terdekat seperti SMOTE.



Gambar 3. *Preprocessing* data latih

C. Resampling dengan SMOTE

Tahap ini menyoroti distribusi kelas target yang sangat timpang *(highly imbalanced)*, di mana insiden kerusakan mesin merupakan kejadian minoritas, penelitian ini menerapkan teknik augmentasi data sintetis menggunakan algoritma *Synthetic Minority Over-sampling Technique* (SMOTE). Pendekatan ini dipilih untuk mengatasi keterbatasan metode *Random Oversampling* konvensional yang cenderung memicu *overfitting* akibat duplikasi data yang berulang.

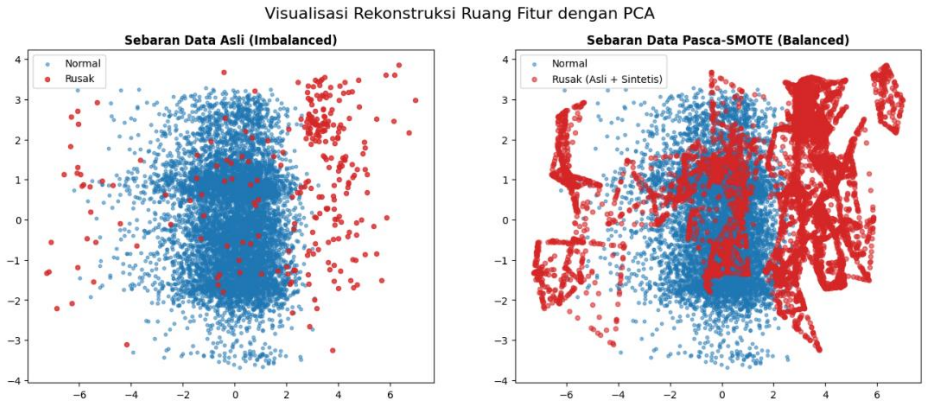


Gambar 4. *Resampling effect*

Gambar 4 menyajikan efektivitas teknik SMOTE dalam menyeimbangkan proporsi kelas. Pada kondisi awal sebelum diterapkan SMOTE, terlihat ketimpangan ekstrem di mana kelas kerusakan warna merah, hanya berjumlah sedikit dibandingkan kelas normal warna biru. Setelah intervensi algoritma dengan SMOTE, jumlah sampel kerusakan meningkat secara sintetis hingga menyamai jumlah sampel normal. Ekuilibrium jumlah data ini setara dengan rasio 1:1, dimana merupakan prasyarat krusial agar model tidak mengalami bias mayoritas saat masuk pada proses pelatihan. Gambar 5 menyajikan ilustrasi dampak geometris dari penerapan SMOTE. Pada *plot* sebelah kiri, data kerusakan pada titik merah tampak jarang

DOI: 10.61124/sinta.v3i1.145

dan terisolasi. Pada *plot* sebelah kanan, terlihat peningkatan densitas titik merah yang mengisi celah antar sampel kerusakan asli. Hal ini mengonfirmasi bahwa SMOTE berhasil membentuk *decision boundary* yang lebih tegas tanpa merusak struktur topologi data asli, yang diharapkan dapat meningkatkan kemampuan model dalam mengenali pola kerusakan yang kompleks.



Gambar 5. Visualisasi rekonstruksi ruang fitur dengan PCA

D. Analisis Performa Algoritma

Evaluasi kinerja algoritma dilakukan berdasarkan skenario *baseline* yaitu tanpa *resampling* dan skenario usulan dengan model *SMOTE*. Hasil komparasi lengkap disajikan pada Tabel 2 berikut.

Tabel 2. Komparasi Model

Sekanrio	Algoritma	Akurasi	Precision	Recall	F1-Score
Tanpa Resampling	XGBoost	0.989	0.9107142857142857	0.75	0.8225806451612904
	LightGBM	0.989	0.9107142857142857	0.75	0.8225806451612904
	CatBoost	0.987	0.9038461538461539	0.6911764705882353	0.7833333333333333
Random Oversampling	LightGBM	0.983	0.7361111111111112	0.7794117647058824	0.7571428571428571
	XGBoost	0.9815	0.7066666666666667	0.7794117647058824	0.7412587412587412
	CatBoost	0.9805	0.6835443037974683	0.7941176470588235	0.7346938775510204
SMOTE *(Proposed)*	XGBoost	0.976	0.6086956521739131	0.8235294117647058	0.7
	LightGBM	0.9655	0.49557522123893805	0.8235294117647058	0.6187845303867403
	CatBoost	0.9655	0.4954954954954955	0.8088235294117647	0.6145251396648045

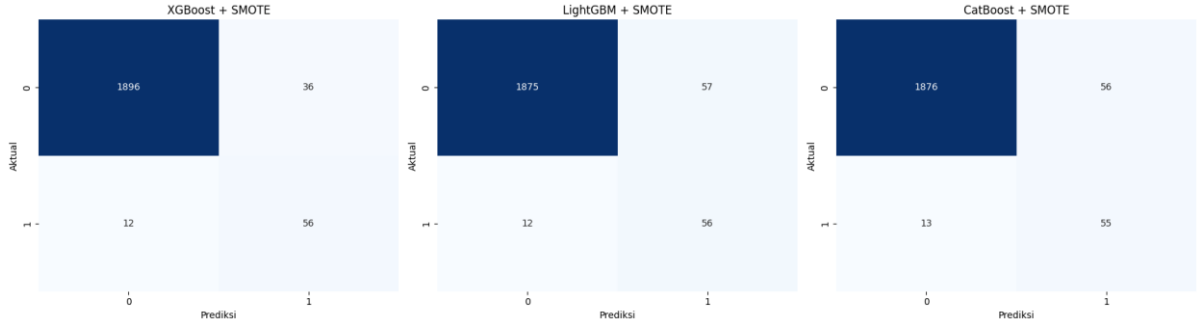
Tabel 2 menyajikan fenomena *accuracy paradox,* terlihat jelas pada skenario tanpa *resampling*. Algoritma *XGBoost* dan *LightGBM* mampu mencapai akurasi 98%, namun nilai *Recall* rendah, yaitu 0.75. Terlihat juga bahwa skenario tanpa SMOTE menghasilkan *F1-Score* yang lebih tinggi dibandingkan skenario SMOTE. Namun, jika ditinjau lebih dalam, tingginya *F1-Score* tersebut didominasi oleh nilai *precision*, sementara nilai *Recall* yaitu kemampuan mendeteksi kerusakan sangat rendah. Ini menunjukkan bahwa model hanya menebak sebagai kategori "Normal" untuk semua data dan gagal mendeteksi kerusakan sama sekali. Akurasi tinggi tersebut bersifat semu dan tidak valid untuk kasus *Predictive Maintenance.*

Sebaliknya, setelah penerapan *SMOTE*, terjadi peningkatan drastis pada *recall.* Model *XGBoost* dan *LighGBM* menunjukkan kinerja paling optimal dengan lonjakan *recall* mencapai 0.82. Dalam konteks *predictive maintenance*, peningkatan *recall* jauh lebih bernilai dibandingkan penurunan pada *precision*. Hal ini dikarenakan biaya risiko akibat kegagalan mendeteksi kerusakan mesin *(False Negative)* jauh lebih fatal dan mahal dibandingkan biaya pengecekan akibat alarm palsu *(False Positive)*. Oleh karena itu, skenario *SMOTE* dinilai

lebih efektif untuk diterapkan pada sistem keamanan industry. Peningkatan ini membuktikan hipotesis penelitian bahwa penyeimbangan distribusi kelas secara signifikan memperbaiki kemampuan model dalam mengidentifikasi anomali mesin.

E. Analisis Confusion Matrix

Untuk memahami detail prediksi model terbaik, analisis dilakukan menggunakan *Confusion Matrix,* visualisasi ini memetakan distribusi prediksi dari model terbaik, yaitu *XGBoost* dengan integrasi *SMOTE,* ke dalam empat kuadran hasil yang merepresentasikan konsekuensi teknisnya, hasil analisis disajikan melalui Gambar 6.



Gambar 6. Hasil Confusion Matrix

Fokus utama penelitian ini adalah keselamatan aset produksi. *Model XGBoost* terbukti mampu mengidentifikasi sebanyak 56 kejadian kerusakan mesin secara tepat *(True Positive).* Angka ini menunjukkan bahwa sistem memiliki sensitivitas yang memadai untuk menangkap mayoritas sinyal anomali yang muncul pada data sensor, sehingga tindakan perawatan preventif dapat segera dijadwalkan sebelum kerusakan fatal terjadi. Meskipun performa model sudah ditingkatkan menggunakan *SMOTE*, masih terdapat 12 kasus kerusakan yang terlewat atau diprediksi normal oleh sistem *(False Negative).* Namun, jika dibandingkan dengan total populasi kelas positif, tingkat kesalahan ini masih berada dalam batas toleransi yang wajar untuk model machine learning pada data yang sangat tidak seimbang. Angka ini menjadi catatan untuk pengembangan selanjutnya, namun risiko fatalitas telah ditekan seminimal mungkin dibandingkan model tanpa penanganan *imbalance*.

Salah satu keunggulan utama *XGBoost* dibandingkan model pembanding lainnya sepeti *LightGBM* dan *CatBoost*, terletak pada kemampuannya menekan angka alarm palsu. Terlihat bahwa *XGBoost* hanya menghasilkan 36 prediksi salah pada kelas normal *(False Positive)*. Sebagai perbandingan, model *LightGBM* menghasilkan 57 alarm palsu, meskipun memiliki kemampuan deteksi kerusakan yang setara 56 *True Positive*. Hal ini menunjukkan bahwa *XGBoost* lebih efisien secara operasional. Model menunjukkan stabilitas yang sangat tinggi dalam mengenali kondisi mesin normal, dengan ketepatan prediksi sebanyak 1896 sampel *(True Negative)*. Dominasi angka ini pada kuadran kiri atas menegaskan bahwa model tidak mengalami *overfitting* berlebihan terhadap kelas minoritas dan tetap andal dalam memantau operasi harian yang berjalan normal.

4. DISKUSI

Temuan pada penelitian ini memberikan gambaran bahwa penanganan *class imbalance* merupakan prasyarat mutlak dalam pengembangan sistem pemeliharaan prediktif yang andal. Pada skenario awal tanpa *resampling*, model menunjukkan bias mayoritas yang ekstrem, ditemukan akurasi yang tinggi dan hanya sebagai refleksi kemampuan model mengenali pola normal, namun gagal total dalam mengidentifikasi anomali kritis. Fenomena ini, yang dikenal sebagai *Accuracy Paradox*, berhasil dimitigasi melalui penerapan teknik *SMOTE*. Peningkatan signifikan pada metrik *Recall* setelah tahap implementasi *SMOTE* mengindikasikan bahwa sintesis data minoritas efektif memperluas *decision boundary model*, memaksanya untuk belajar mengenali fitur-fitur laten penyebab kerusakan mesin yang sebelumnya tertutup oleh dominasi data normal. *XGBoost* menunjukkan superioritasnya sebagai model yang paling *robust* dibandingkan *LightGBM* dan *CatBoost*. Keunggulan ini terlihat dari kemampuan *XGBoost* dalam menyeimbangkan sensitivitas dan presisi. Sebagaimana termuat dalam analisis *Confusion Matrix*, meskipun *LightGBM* mampu mencapai tingkat deteksi kerusakan *(True Positive)* yang setara dengan *XGBoost*, algoritma tersebut cenderung menghasilkan tingkat alarm palsu *(False Positive)* yang jauh lebih tinggi. Hal ini mengindikasikan bahwa mekanisme

regularisasi pada *XGBoost* bekerja lebih efektif dalam mengontrol kompleksitas model, mencegah *overfitting* terhadap *noise* yang mungkin timbul dari data sintetis buatan *SMOTE*, sehingga menghasilkan prediksi yang lebih efisien secara operasional. Berdasarkan perspektif keberlanjutan operasional industri, hasil eksperimen ini menawarkan implikasi manajerial yang krusial. Dalam ekosistem *Predictive Maintenance*, biaya kesalahan prediksi bersifat asimetris, biaya akibat kegagalan mendeteksi kerusakan *(False Negative)* jauh melampaui biaya inspeksi akibat alarm palsu *(False Positive)*. Oleh karena itu, dominasi *XGBoost* dengan integrasi *SMOTE* yang mampu menekan angka *False Negative* hingga titik minimal, sambil menjaga *False Positive* pada tingkat yang wajar, merupakan konfigurasi yang paling optimal. Model ini tidak hanya berfungsi sebagai alat deteksi teknis, tetapi juga sebagai instrumen mitigasi risiko yang valid untuk meminimalkan *unplanned downtime* dan kerugian finansial akibat kerusakan mesin yang tak terduga.

5. KESIMPULAN

Penelitian ini menyimpulkan bahwa integrasi teknik *Synthetic Minority Over-sampling Technique* (SMOTE) merupakan strategi vital dalam mengatasi bias mayoritas pada data perawatan mesin yang ekstrem. Dari rangkaian eksperimen komparatif, algoritma *XGBoost* yang dikombinasikan dengan SMOTE terbukti sebagai model paling optimal, tidak hanya karena kemampuannya mendeteksi potensi kerusakan dengan sensitivitas melalui nilai *recall* yang tinggi, tetapi juga efisiensinya dalam menekan laju alarm palsu *(False Positive)* dibandingkan kompetitornya seperti *LightGBM*. Keseimbangan performa ini menegaskan bahwa model yang diusulkan mampu memberikan jaminan keselamatan operasional yang lebih baik sekaligus meminimalkan pemborosan sumber daya akibat inspeksi yang tidak perlu, menjadikannya solusi cerdas yang layak diterapkan untuk mitigasi risiko *downtime*. Untuk penelitian selanjutnya disarankan memperdalam aspek teknis melalui penerapan optimasi hiperparameter otomatis seperti Bayesian Optimization, dan eksplorasi metode *Deep Learning* seperti LSTM untuk menangkap pola data yang lebih kompleks. Di samping itu, model *Explainable AI* menggunakan SHAP atau LIME menjadi langkah krusial guna menjamin transparansi keputusan model, sehingga prediksi kerusakan tidak hanya presisi tetapi juga dapat dipahami secara logis oleh operator di lapangan.

DAFTAR PUSTAKA

[1] T. Zonta, C. A. Da Costa, R. da Rosa Righi, M. J. de Lima, E. S. Da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, p. 106889, 2020.

[2] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, p. 8211, 2020.

[3] P. Mallioris, E. Aivazidou, and D. Bechtsis, "Predictive maintenance in Industry 4.0: A systematic multi-sector mapping," *CIRP J. Manuf. Sci. Technol.*, vol. 50, pp. 80–103, 2024, doi: https://doi.org/10.1016/j.cirpj.2024.02.003.

[4] M. Achouch *et al.*, "On predictive maintenance in industry 4.0: Overview, models, and challenges," *Appl. Sci.*, vol. 12, no. 16, p. 8081, 2022.

[5] M. T. Y. Taimun, S. M. I. Sharan, M. A. Azad, and M. M. I. Joarder, "Smart maintenance and reliability engineering in manufacturing," *Saudi J. Eng. Technol.*, vol. 10, no. 4, pp. 189–199, 2025.

[6] J. Dalzochio *et al.*, "Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges," *Comput. Ind.*, vol. 123, p. 103298, 2020.

[7] T. P. Carvalho, F. A. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, p. 106024, 2019.

[8] M. Pech, J. Vrchota, and J. Bednář, "Predictive maintenance and intelligent sensors in smart factory," *Sensors*, vol. 21, no. 4, p. 1470, 2021.

[9] A. Kane, A. Kore, A. Khandale, S. Nigade, and P. Joshi, *Predictive Maintenance using Machine Learning*. 2022. doi: 10.48550/arXiv.2205.09402.

[10] O. Merkt, "Predictive Models for Maintenance Optimization: an Analytical Literature Survey of Industrial Maintenance Strategies *," Jan. 2020.

[11] B. Lu, Z. Chen, and X. Zhao, "Data-driven dynamic predictive maintenance for a manufacturing system with quality deterioration and online sensors," *Reliab. Eng. [?] Syst. Saf.*, vol. 212, Mar. 2021, doi: 10.1016/j.ress.2021.107628.

[12] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, 2019.

[13] S. Cicak and U. Avci, *Handling Imbalanced Data in Predictive Maintenance: A Resampling-Based Approach*. 2023. doi: 10.1109/HORA58378.2023.10156799.

[14] D. B. Arianto and S. Nurrahmasita, "A Comparative Study For Imbalanced Data Techniques Of Classification Algorithms," *Citiz. J. Ilm. Multidisiplin Indones.*, vol. 5, no. 4, pp. 1064–1073, 2025.

[15] Y. Zhang, X. Li, L. Gao, L. Wang, and L. Wen, "Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning," *J. Manuf. Syst.*, vol. 48, pp. 34–50, Sep. 2018, doi: 10.1016/j.jmsy.2018.04.005.

[16] S. Liu, H. Jiang, Z. Wu, Z. Yi, and R. Wang, "Intelligent fault diagnosis of rotating machinery using a multi-source domain adaptation network with adversarial discrepancy matching," *Reliab. Eng. Syst. Saf.*, vol. 231, p. 109036, 2023, doi: https://doi.org/10.1016/j.ress.2022.109036.

[17] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf. Sci. (Ny).*, vol. 513, pp. 429–441, 2020.

[18] B. Pes and G. Lai, "Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study," *PeerJ Comput. Sci.*, vol. 7, p. e832, Dec. 2021, doi: 10.7717/peerj-cs.832.

[19] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Inf. Sci. (Ny).*, vol. 505, pp. 32–64, 2019.

[20] D. Arifah, T. H. Saragih, D. Kartini, and M. I. Mazdadi, "Application of SMOTE to Handle Imbalance Class in Deposit Classification Using the Extreme Gradient Boosting Algorithm," *J. Ilm. Tek. Elektro Komput. dan Inf.*, vol. 9, no. 2, pp. 396–410, 2023.

[21] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "Anomaly detection model based on gradient boosting and decision tree for IoT environments security," *J. Reliab. Intell. Environ.*, vol. 9, Jul. 2022, doi: 10.1007/s40860-022-00184-3.

[22] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. big data*, vol. 7, no. 1, p. 94, 2020.

[23] A. Odeh, Q. Abu Al-Haija, A. Aref, and A. abu taleb, "Comparative Study of CatBoost, XGBoost, and LightGBM for Enhanced URL Phishing Detection: A Performance Assessment," *J. Internet Serv. Inf. Secur.*, vol. 13, pp. 1–11, Dec. 2023, doi: 10.58346/JISIS.2023.I4.001.

[24] D. Guidotti, L. Pandolfo, and L. Pulina, "A Systematic Literature Review of Supervised Machine Learning Techniques for Predictive Maintenance in Industry 4.0," *IEEE Access*, vol. PP, p. 1, Jan. 2025, doi: 10.1109/ACCESS.2025.3578686.

[25] S. Matzka, "Explainable artificial intelligence for predictive maintenance applications," in *2020 third international conference on artificial intelligence for industries (ai4i)*, IEEE, 2020, pp. 69–74.

[26] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.