

Prediksi Cacat Lempeng Baja Menggunakan Algoritma *Bagging*: Pendekatan Pembelajaran Mesin untuk Peningkatan Kualitas Produksi

Aji Digdoyo¹, Adhitio Satyo Bayangkari Karno², Widi Hastomo³, Elliya Sestri³, dan Reza Fitriansyah³

¹Department of Mechanical Engineering, Diponegoro University

²Universitas Gunadarma,

³Institut Teknologi dan Bisnis Ahmad Dahlan

E-mail: digdoyoaji@gmail.com, adh1t10.2@gmail.com, widie.has@gmail.com*),
ellyasestri24@gmail.com, rezafsyh@gmail.com

Abstrak

Industri baja memiliki peran krusial dalam berbagai sektor, menjadi faktor kunci dalam memastikan integritas struktural produk akhir. Penelitian ini bertujuan untuk mengatasi masalah ini dengan menerapkan algoritma *Bagging* dalam prediksi cacat lempeng baja. Hasil model training dengan kurva ROC dengan nilai AUC 99% dan *logloss* 0,14. Pengukuran *precision*, *recall*, dan *f1 score* untuk 7 jenis cacat baja memperoleh prosentase yang sangat baik (lebih dari 90%). *Confusion Matrix* menunjukkan korelasi yang kuat antara jenis cacat ke 6 dan ke 5. Sedangkan validasi, antara jenis cacat ke 4 dan ke 0 terdapat hubungan yang sangat kuat. *Classification report* menunjukkan nilai *precision*, *recall*, dan *f1 score* terbaik (lebih dari 80%) untuk jenis cacat ke 1, 2, dan 3. Nilai AUC yang cukup baik yaitu 88% dan *Logloss* yang cukup besar yaitu 3,13. Penelitian selanjutnya dapat fokus untuk meningkatkan nilai *logloss* yang masih harus diperbaiki untuk proses validasi.

Kata kunci: *prediksi cacat, pelat baja, algoritma bagging, pembelajaran mesin.*

Pendahuluan

Industri baja memegang peranan penting dalam manufaktur dan konstruksi serta berkontribusi signifikan terhadap pertumbuhan ekonomi global [1]. Baja digunakan untuk berbagai komponen struktural dan mekanikal seperti kendaraan, jembatan, mesin, dan bangunan [2]. Oleh karena itu, menjaga kualitas produksi lempeng baja sangatlah penting, karena adanya cacat pada lempeng baja dapat mempengaruhi kualitas produk dan kinerja struktur [3].

Deteksi cacat pada lempeng baja tidak hanya bertujuan untuk meningkatkan kualitas produk, tetapi juga berperan dalam mengurangi biaya produksi, mencegah risiko kegagalan struktur, dan memperpanjang umur pemakaian produk [4]. Seiring meningkatnya permintaan produk baja berkualitas tinggi, metode deteksi cacat terus berkembang dengan dukungan teknologi modern [5].

Namun, metode konvensional yang digunakan saat ini masih menghadapi berbagai tantangan [6]. Teknik seperti inspeksi visual dan pengujian non-destruktif (NDT) masih bergantung pada keahlian manusia [7], [8]. Keterbatasan ini sering kali

mengakibatkan hasil yang tidak konsisten, dengan potensi terlewatnya cacat kecil tetapi signifikan. Selain itu, metode ini memerlukan waktu lama, tidak efisien untuk volume produksi tinggi, serta memiliki kompleksitas operasional yang sulit diterapkan pada struktur baja yang mutakhir. Meskipun metode NDT yang lebih canggih seperti ultrasonik dan radiografi telah digunakan, teknik ini tetap menghadapi keterbatasan terkait biaya tinggi, proses yang rumit, serta ketidakefektifan dalam mendeteksi cacat internal secara waktu nyata [9].

Untuk mengatasi keterbatasan tersebut, teknologi berbasis pembelajaran mesin muncul sebagai solusi yang lebih canggih dan efisien. Salah satu pendekatan yang potensial adalah algoritma *Bagging*, yaitu teknik *ensemble learning* yang menggabungkan prediksi dari beberapa model dasar untuk meningkatkan akurasi dan mengurangi variabilitas hasil [9]. Dengan kemampuannya mengatasi masalah *overfitting*, menggabungkan prediksi, dan memproses volume data besar, *Bagging* dianggap sebagai metode yang tepat untuk deteksi cacat baja secara otomatis dan waktu nyata.

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Bagging* dalam deteksi cacat pada

lempeng baja untuk mencapai akurasi yang optimal dengan menggabungkan prediksi dari beberapa model dasar dan mengurangi variabilitas hasil. Tinjauan literatur menunjukkan bahwa metode berbasis *computer vision* dan pengolahan citra telah dikembangkan untuk mendeteksi cacat permukaan [10]. Namun, kemampuan dalam mendeteksi cacat internal yang kompleks masih kurang mendapat perhatian [11]. Dalam konteks ini, algoritma *Bagging* menawarkan solusi dengan hasil prediksi yang lebih stabil, andal, dan presisi tinggi. Selain itu, pendekatan ini dapat mengolah data berskala besar, sehingga sangat relevan dengan kebutuhan industri baja saat ini.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam mengatasi keterbatasan metode konvensional, serta memiliki dampak praktis dalam meningkatkan kualitas produk baja dan efisiensi produksi secara keseluruhan.

Metode Penelitian

Dataset

Penelitian ini menggunakan kumpulan data dalam format CSV yang diperoleh dari www.kaggle.com

[12], dengan nama file "train.csv". Dataset ini mencakup berbagai karakteristik cacat lempeng baja, termasuk morfologi, ukuran retak, inklusi non-logam, lubang, deformasi permukaan, dan kemungkinan dampak cacat tersebut pada kualitas produk akhir.

Dataset ini cukup besar, dengan lebih dari 19.000 baris dan 33 kolom. Ada 26 kolom untuk data input dan 7 kolom untuk data target; yang pertama mengandung informasi tentang 7 jenis cacat yang ada di lempeng baja, dan yang kedua menampilkan ukuran dan bentuk cacat di lempeng baja, seperti yang ditunjukkan pada Gambar 1. Dataset ini sangat relevan untuk penelitian ini karena mencerminkan kondisi produksi industri baja yang sebenarnya. Selain itu, karena dataset ini representatif, model yang dikembangkan memiliki validitas eksternal yang kuat dan aplikasi praktis yang lebih luas. Penelitian ini diharapkan dapat menghasilkan peningkatan signifikan dalam akurasi dan efisiensi deteksi cacat pada lempeng baja dengan menggunakan data yang realistis dan komprehensif.

	id	X_Minimum	X_Maximum	Y_Minimum	Y_Maximum	Pixels_Areas	X_Perimeter	Y_Perimeter	Sum_of_Luminosity	Minimum_of_Luminosity	Maximum
	0	0	584	590	909972	909977	16	8	5	2274	113
	1	1	808	816	728350	728372	433	20	54	44478	70
	2	2	39	192	2212076	2212144	11388	705	420	1311391	29
	3	3	781	789	3353146	3353173	210	16	29	3202	114
	4	4	1540	1560	618457	618502	521	72	67	48231	82

	19214	19214	749	757	143210	143219	17	4	4	2193	122
	19215	19215	723	735	2488529	2488541	231	17	26	27135	104
	19216	19216	6	31	1578055	1578129	780	114	98	71112	41
	19217	19217	9	18	1713172	1713184	126	13	26	14808	88
	19218	19218	1505	1525	1733458	1733471	182	24	33	22785	98

	LogOfAreas	Log_X_Index	Log_Y_Index	Orientation_Index	Luminosity_Index	SigmoidOfAreas	Pastry	Z_Scratch	K_Scratch	Stains	Dirtiness	Bumps	Other_Faults
	1.2041	0.9031	0.6990	-0.5000	-0.0104	0.1417	0	0	0	1	0	0	0
	2.6365	0.7782	1.7324	0.7419	-0.2997	0.9491	0	0	0	0	0	0	1
	4.0564	2.1790	2.2095	-0.0105	-0.0944	1.0000	0	0	1	0	0	0	0
	2.3222	0.7782	1.4314	0.6667	-0.0402	0.4025	0	0	1	0	0	0	0
	2.7694	1.4150	1.8808	0.9158	-0.2455	0.9998	0	0	0	0	0	0	1

	1.2305	0.7782	0.6021	-0.1429	0.0044	0.2901	0	0	0	1	0	0	0
	2.3636	1.0414	1.4150	0.7222	-0.0989	0.5378	0	0	0	0	0	0	1
	2.8921	1.4314	1.8692	0.7719	-0.4283	0.9997	1	0	0	0	0	0	0
	2.1004	1.0414	1.4150	0.9610	-0.1162	0.3509	0	0	0	0	0	0	1
	2.2601	1.1461	1.5051	0.5263	-0.1120	0.6619	0	0	0	0	0	1	0

19219 rows x 35 columns

Gambar 1: File dataset "train.csv"

Pengolahan Awal Dataset

Pengolahan awal dataset merupakan langkah krusial untuk memastikan kualitas data sebelum digunakan dalam proses pelatihan model prediksi. Bagian ini akan membahas secara ringkas mengenai proses-proses yang dilakukan sebelum melaksanakan tahap pelatihan menggunakan algoritma *Bagging*. Langkah ini penting untuk menghindari bias dalam model pembelajaran mesin yang dapat timbul akibat perbedaan skala antar variabel.

Pemisahan Dataset

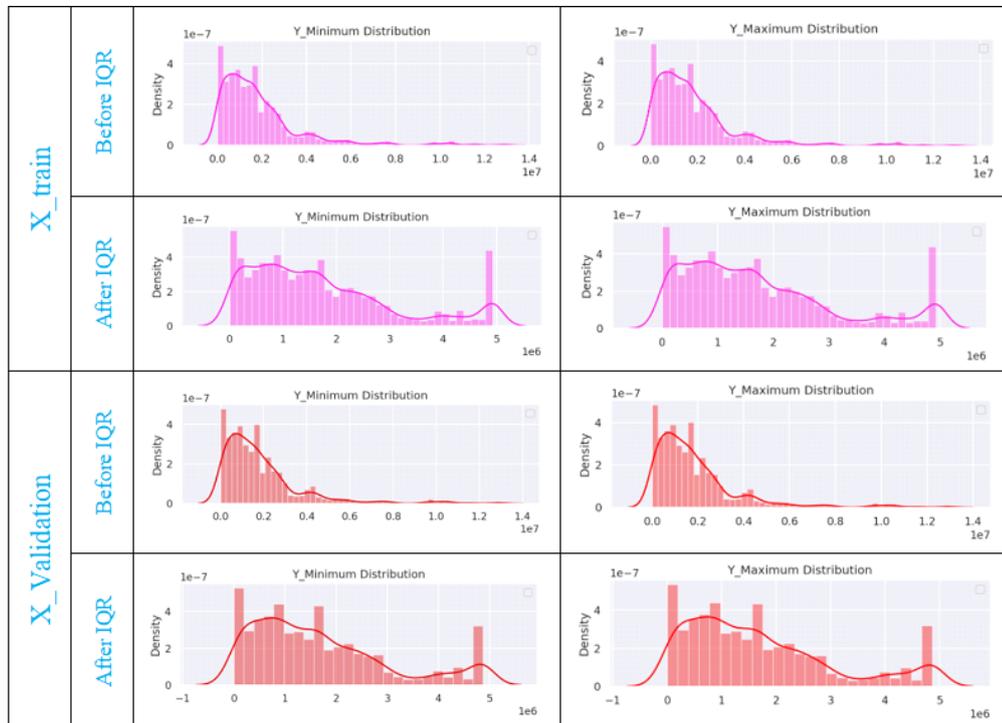
Pembagian dataset menjadi data pelatihan dan validasi dilakukan untuk memastikan model yang dikembangkan memiliki performa yang baik dan dapat digeneralisasi. Metode *k-fold cross-validation* dipilih untuk memberikan estimasi yang lebih akurat terhadap kinerja model dengan menggunakan beberapa subset dari dataset [13].

Rasio pembagian data mengikuti proporsi 75% untuk pelatihan dan 25% untuk validasi, yang dipilih berdasarkan praktik terbaik dalam pembelajaran mesin dan kebutuhan spesifik penelitian ini [14]. Justifikasi pemilihan metode ini didasarkan pada kebutuhan untuk mengurangi *overfitting* dan meningkatkan kemampuan generalisasi model pada data baru. Data target menyusut dari tujuh kolom menjadi satu, sehingga totalnya turun menjadi dua

puluh tujuh kolom. 27 kolom data tersebut dibagi menjadi 26 kolom untuk data fitur (x) dan 1 kolom untuk data target (y) guna mempersiapkan proses pelatihan. Baris data asli yang awalnya berisi 19 ribu lebih, dipisahkan menjadi 14 ribu lebih baris data pelatihan (75%) dan 4 ribu lebih baris data validasi (25%) untuk kedua kategori data tersebut (x dan y).

Data outlier

Penanganan data *outlier* dilakukan untuk memastikan integritas data. Data *outlier* yang dapat merusak performa model diidentifikasi dan ditangani dengan menggunakan metode IQR (*Interquartile Range*). Karena pendekatan non-parametrik ini cenderung tidak menghasilkan temuan ekstrim atau *outlier*, pendekatan ini sangat berguna bila diterapkan pada kumpulan data dengan distribusi yang berat atau tidak normal. IQR adalah selisih antara persentil ke-75 dan ke-25, atau kuartil ketiga dan pertama, dari sebaran data. Karena Q1 dan Q3 mewakili kuartil pertama dan ketiga, maka semua data yang berada di luar rentang antara batas $-1,5 \cdot IQR + Q1$ dan batas $1,5 \cdot IQR + Q3$ dianggap *outlier* [15]. Karena keterbatasan halaman, contoh ini hanya menampilkan plot sebaran dari sebagian fitur *data_train* dan *data_validation* yang menunjukkan keberadaan *outlier*, serta plot sebar fitur yang mengikuti tahapan IQR (gambar 2).



Gambar 2: Plot distribusi dengan penerapan IQR sebelum dan sesudah

Data Tidak Seimbang

Ketidakeimbangan kelas muncul ketika kumpulan data yang digunakan untuk mengidentifikasi cacat

pelat baja berisi lebih banyak sampel dari kelas non-cacat (kelas mayoritas) dibandingkan dari kelas cacat (kelas minoritas). Perbedaan data ini

mempunyai kemampuan untuk merusak algoritma pembelajaran mesin dan mengganggu akurasi kategorisasi kelas minoritas. Untuk mengatasi masalah dalam penelitian ini, teknik *Synthetic Minority Over-sampling* (SMOTE) diterapkan pada kumpulan data [16].

Menggunakan pendekatan *oversampling* yang disebut SMOTE, sampel minoritas buatan dibuat dengan melakukan interpolasi dari sampel minoritas yang sudah ada sebelumnya untuk kelas minoritas. Proses ini dilanjutkan sampai kelas minoritas memperoleh jumlah sampel sintetik yang diperlukan. SMOTE diterapkan pada kumpulan data pelat baja yang mengalami ketidakseimbangan kelas antara cacat dan non-cacat. Dengan menggunakan nilai $k = 5$ untuk mencari tetangga terdekat pada proses SMOTE. Jumlah sampel sintetik yang dihasilkan disesuaikan untuk mencapai rasio yang lebih seimbang antara kelas cacat dan tidak cacat.

Penskalaan

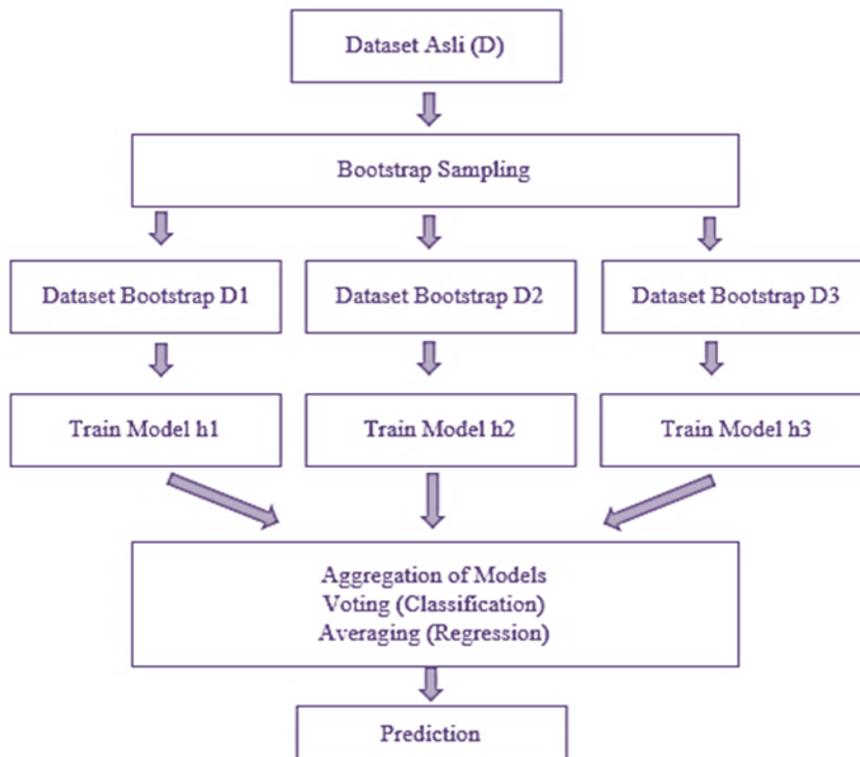
Dalam pembelajaran mesin, penskalaan fitur merupakan langkah penting dalam pra-pemrosesan data. Penskalaan fitur bertujuan untuk menempatkan semua nilai fitur pada satu skala, dalam rentang yang telah ditentukan [17]. Hal ini penting karena perbedaan ukuran fitur dapat mempengaruhi performa beberapa teknik pembelajaran mesin [18].

Di antara metode penskalaan fitur yang paling banyak digunakan adalah *StandardScaler*. Metode penskalaan yang disebut *StandardScaler* memodifikasi setiap karakteristik sehingga rata-rata (*mean*) sama dengan nol dan variansnya sama dengan satu [15]. Dengan menggunakan metode ini, rata-rata dan standar deviasi setiap nilai fitur dibagi [19].

StandardScaler sudah terintegrasi dalam paket *scikit-learn* Python dengan memanfaatkan kelas *StandardScaler* dari modul *preprocessing* [20]. Selain itu, beberapa studi teoritis dan empiris telah menunjukkan bahwa model pembelajaran mesin bekerja lebih baik ketika *StandardScaler* digunakan.

Bagging

Algoritma *Bagging* adalah teknik pembelajaran mesin yang bertujuan meningkatkan stabilitas dan akurasi prediksi dengan memadukan beberapa model dasar. Metode ini awalnya diperkenalkan oleh Leo Breiman in 1996 sebagai alat bantu untuk pemrosesan data. Ide utama dari algoritma *Bagging* adalah membuat sejumlah bagian dataset asli melalui prosedur *bootstrap sampling*, dimana setiap bagian tersebut digunakan untuk melatih model berbeda-beda. Prediksi akhir diperoleh melalui *voting* untuk klasifikasi atau rata-rata untuk regresi, merupakan salah satu yang sangat populer untuk algoritma pembelajaran mesin.



Gambar 3: Bagan proses algoritma Bagging

Bagging memberikan manfaat yang signifikan dalam konteks deteksi cacat pada lempeng baja. Dengan menggabungkan beberapa model, *Bagging* memungkinkan penangkapan karakteristik dan pola cacat yang beragam, yang terkadang tidak mungkin terdeteksi oleh model individu. Hal ini penting untuk meningkatkan akurasi dan komprehensifitas deteksi, mengingat kompleksitas dan variasi cacat yang ada pada lempeng baja. Selain itu, *Bagging* digunakan untuk mengurangi sensitifitas terhadap anomali data dan peningkatan kemampuan generalisasi model.

Dalam penelitian ini, algoritma *Bagging* diimplementasikan menggunakan perangkat lunak Python dengan pustaka *scikit-learn*, yang memungkinkan pengguna untuk mendukung alat dan fungsi yang diperlukan untuk mengimplementasikan *Bagging*. Model yang dipilih dalam algoritma *Bagging*, yaitu *decision trees* dan *random forests*, dipilih karena kemampuannya untuk menangani ancaman data yang berbeda dari berbagai tipe dan skalabilitas. Parameter penting seperti jumlah model dasar (*estimators*) dan *depth* dari *decision trees* diatur secara hati-hati untuk mengoptimalkan kinerja model.

Proses pelatihan menggunakan algoritma *Bagging* melibatkan beberapa langkah utama. Pertama, dataset awal dibagi menjadi subset awal melalui *bootstrap sampling*. Kemudian, setiap subset yang dihasilkan dilatih pada model dasar, yang hasil prediksinya digabungkan untuk menghasilkan bagian akhir dari model pembelajaran [21].

Selama pelatihan, kinerja model diskriminasi menggunakan metode *cross-validation* untuk memastikan bahwa model tidak *overfitting* dan memiliki kemampuan yang generalisasi tinggi (gambar 3). Algoritma *Bagging* membantu mengurangi perbedaan model serta meningkatkan akurasi dan keandalan dari prediksi. Ini dapat dianggap sangat berguna dalam ancaman pencacatan pada pembakaran lempeng logam.

Evaluasi Kinerja

Salah satu langkah penting dalam studi ini adalah melakukan evaluasi kinerja model. Ini dilakukan untuk memastikan bahwa algoritma *Bagging* yang dibuat dapat mendeteksi cacat pada pelat baja secara akurat dan efisien. Tujuan utama evaluasi ini adalah untuk menentukan seberapa efektif model dalam berbagai kondisi dan seberapa andal model tersebut dalam lingkungan industri dunia nyata.

Evaluasi menyeluruh sangat penting untuk membenarkan hasil penelitian dan memberikan bukti empiris tentang keunggulan model yang disarankan. Sangat penting untuk melakukan evaluasi kinerja karena tanpa evaluasi yang tepat, hasil studi tidak akan memiliki validitas yang cukup untuk diakui oleh komunitas ilmiah dan industri.

Untuk menetapkan jalan untuk perbaikan dan pengembangan, evaluasi yang sistematis dan ke-

perlu dilakukan untuk mengidentifikasi kekuatan dan kelemahan suatu model. Dalam konteks ini, evaluasi juga berfungsi sebagai jembatan antara pengembangan model laboratorium dan aplikasinya dalam dunia nyata, memastikan bahwa solusi yang dihasilkan tidak hanya teoritis tetapi juga praktis. Kriteria evaluasi yang digunakan dalam penelitian ini dimulai dengan pengukuran menggunakan *Confusion Matrix*, kemudian dapat diturunkan menjadi berbagai metrik performa penting lainnya yaitu akurasi, *precision*, *recall*, *F1-score* dan *log loss* dan dapat juga ditunjukkan dalam bentuk kurva ROC dan AUC.

Confusion Matrix

Dalam pembelajaran mesin, mengevaluasi kinerja model klasifikasi sangat diperlukan untuk memahami sejauh mana model efektif dan di mana area yang memerlukan perbaikan. *Confusion matrix* adalah alat dasar untuk menyelesaikan pertanyaan ini, memberikan sejumlah detail tentang prediksi yang dibuat oleh model. Dengan kata lain, *confusion matrix* adalah visualisasi kinerja algoritma, di mana para praktisi membandingkan klasifikasi yang diprediksi dengan kelas yang sebenarnya. Secara umum, sebuah *confusion matrix* berupa matriks persegi yang merangkum kinerja suatu algoritma klasifikasi. Berikut di Gambar 4 adalah *confusion matrix* dan *accuracy* dalam pembelajaran mesin.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Gambar 4: Confusion matrix

TP adalah singkatan dari *True Positives*, yang menunjukkan jumlah kasus positif yang benar-benar diprediksi dengan benar. FP mengacu pada *false positives*, yang menunjukkan jumlah kasus negatif yang salah diprediksi sebagai positif. TN adalah *true negatives*, mengacu pada jumlah kasus negatif yang diprediksi dengan benar. FN menunjukkan *false negatives*, yang menunjukkan berapa banyak positif aktual yang salah diperkirakan sebagai negatif. Dari confusion matrik ini, beberapa ukuran kinerja dapat diturunkan, seperti: Akurasi: Fraksi (TP + TN) mewakili semua "prediksi yang benar" di antara semuanya. Presisi: persentase hasil positif yang diprediksi dengan benar dibandingkan dengan semua prediksi perkiraan lainnya, termasuk pandangan ke depan *Recall (Sensitivity)*.

Logloss

Logarithmic Loss, yang biasa dikenal sebagai *Logloss*, *metric* dasar yang digunakan untuk mengukur *performance* model klasifikasi, terutama dalam probabilistik kerangka, memberikan indikasi seberapa baik probabilitas prediksi model sesuai dengan kelas label sebenarnya. *Logarithmic Loss* nilai yang lebih rendah adalah lebih baik, dan *Logarithmic Loss* secara nol adalah hasil yang sempurna.

Logloss signifikan digunakan dalam area yang tidak pasti dari spesialis habilidades seperti finansial, perawatan kesehatan, dan pemasaran, di mana ketidakpastian tentang prediksi penting. *Metrik* juga merupakan parameter evaluasi standar untuk banyak kompetisi pembelajaran mesin, antara lain oleh Kaggle. Pustaka seperti *Scikit-learn*

dalam Python menyediakan fungsi yang mudah digunakan untuk menghitung Kerugian Logaritmik, yang memfasilitasi penggunaannya secara luas.

Hasil dan Pembahasan

Hasil penelitian ini menunjukkan bahwa model prediksi cacat pada lempeng baja yang dikembangkan menggunakan algoritma *Bagging* memberikan kinerja yang superior dibandingkan dengan metode tradisional. Berdasarkan hasil dari metrik kinerja, mengilustrasikan hasil training dan validasi algoritma *bagging* untuk 7 jenis cacat dengan menggunakan beberapa alat ukur yaitu ROC, *Confusion Matrix*, presisi, *recall*, *f1-score*, AUC, akurasi, dan *Logloss* (gambar 5).

	ROC	Confusion Matrix	Classification Reports	SCORE		
				AUC	Acc	LogLoss
Training			<pre> precision recall f1-score 0 0.99 0.99 0.99 1 0.99 1.00 0.99 2 0.99 1.00 0.99 3 0.99 1.00 1.00 4 0.99 1.00 1.00 5 0.98 0.98 0.98 6 0.99 0.96 0.98 accuracy 0.99 </pre>	0.999854520294	0.990017564402	0.143476422656
Validation			<pre> precision recall f1-score 0 0.39 0.48 0.43 1 0.81 0.78 0.80 2 0.89 0.92 0.91 3 0.91 0.93 0.92 4 0.69 0.48 0.57 5 0.44 0.47 0.46 6 0.39 0.37 0.38 accuracy 0.64 </pre>	0.886405726174	0.635197934595	3.138371389676

Gambar 5: Hasil pengukuran training dan validasi.

Dari proses training yang telah dilakukan, secara kasar terlihat dari luas dibawah kurva grafik ROC untuk 7 jenis cacat menunjukkan sempurna (bernilai 1). Untuk *confusion matrix* dapat diketahui hubungan yang sangat kuat ada diantara jenis cacat ke 6 dan jenis cacat ke 5 dengan nilai 67. Dari *classification report* ketiga pengukuran *precision*, *recall*, dan *f1 score* semua mempunyai prosentase lebih dari 90%.

Nilai AUC merupakan nilai luas rata-rata kurva ROC untuk 7 jenis cacat yaitu 99%, nilai *Logloss* dari proses training cukup kecil yaitu 0,14. Hasil dari proses training mengindikasikan kemampuan yang sangat baik dalam mengklasifikasikan cacat dengan tepat, menunjukkan bahwa model ini mampu meminimalkan *false positives* dan mendeteksi sebagian besar cacat yang ada. Sedangkan untuk proses validasi, luas dibawah kurva grafik ROC untuk 7 jenis cacat terlihat bervariasi.

Hubungan yang sangat kuat antara jenis cacat ke 4 dan jenis cacat ke 0, dengan nilai *confusion matrix* sebesar 5.4e+02. Pengukuran yang lebih detail *classification report* menunjukkan nilai *precision*, *recall*, dan *f1 score* terbaik (lebih dari 80%) untuk jenis cacat ke 1, 2, dan 3. Nilai AUC yang cukup baik yaitu 88% dan *Logloss* yang cukup besar yaitu 3,13 [22].

Agar lebih jelas nilai numerik skor AUC, Acc, dan *Logloss* untuk training dan validasi dapat diilustrasikan secara grafik (gambar 6), menunjukkan perbedaan yang cukup berarti untuk pengukuran *Logloss*. *Logloss* untuk proses validasi terlihat lebih besar dibandingkan dengan proses training, ini berarti model untuk proses training lebih baik dibandingkan dengan proses validasi. Perbedaan nilai *logloss* ini dapat disebabkan sebaran data yang dipergunakan antara proses training dan validasi perlu untuk ditingkatkan kembali. Nilai AUC 0.886

menunjukkan bahwa model *Bagging* memiliki kemampuan yang sangat baik dalam mendeteksi dan membedakan cacat dan bukan cacat pada baja. Model memiliki performa yang stabil untuk mendeteksi kelas positif dan negatif dengan cukup baik.



Gambar 6: Hasil pengukuran AUC, Acc, dan Logloss

Penutup

Hasil diskusi mengenai temuan ini menunjukkan bahwa penggunaan algoritma *Bagging* tidak hanya meningkatkan performa deteksi cacat, tetapi juga menyediakan kerangka kerja yang lebih robust dan adaptif untuk aplikasi pengendalian kualitas di industri lainnya. Meskipun demikian, ada beberapa tantangan yang perlu diperbaiki dari hasil penelitian ini, yaitu meningkatkan sebaran data yang lebih seimbang antara data yang dipergunakan dalam proses training dengan data yang dipergunakan dalam proses validasi, sehingga harapan untuk memperoleh nilai *Logloss* dapat menjadi lebih baik terutama untuk proses validasi.

Model ini dapat diintegrasikan dengan sistem produksi yang ada tanpa gangguan signifikan terhadap proses operasional. Implementasi model di lingkungan nyata dapat meningkatkan efisiensi pengendalian kualitas, dengan deteksi cacat yang lebih cepat dan lebih akurat dibandingkan metode manual atau berbasis teknik konvensional. Pengurangan jumlah produk cacat yang tidak terdeteksi juga berkontribusi pada penurunan biaya operasional dan peningkatan kepuasan pelanggan.

Secara keseluruhan, penelitian ini memberikan kontribusi signifikan terhadap literatur ilmiah dalam bidang pembelajaran mesin dan pengendalian kualitas. Temuan ini membuka jalan bagi inovasi lebih lanjut dalam aplikasi industri, menawarkan solusi praktis yang dapat meningkatkan kualitas produk dan efisiensi operasional secara substansial.

Daftar Pustaka

- [1] H. Wandebori and Murtyastanto, "The Implication of Steel-Intensity-of-Use on Economic Development," *Sustainability*, vol. 15, no. 16, doi: 10.3390/su151612297, 2023.
- [2] A. Bucur, G. Dobrotă, C. Oprean-Stan, and C. Tănăsescu, "Economic and Qualitative Determinants of the World Steel Production," *Metals (Basel)*, vol. 7, no. 5, doi: 10.3390/met7050163, 2017.
- [3] Z. Jie, H. Zhang, K. Li, X. Xie, and A. Shi, "Image Enhancement of Steel Plate Defects Based on Generative Adversarial Networks," *Electronics*, vol. 13, no. 11, doi: 10.3390/electronics13112013, 2024.
- [4] A. A. P. Chazhooor, E. S. L. Ho, B. Gao, and W. L. Woo, "A Review and Benchmark on State-of-the-Art Steel Defects Detection," *SN Comput. Sci.*, vol. 5, no. 1, p. 114, doi: 10.1007/s42979-023-02436-2, 2023.
- [5] S. Papaefthymiou, T. Tzeveleku, A. Antonopoulos, and A. Gypakis, "Typical defects in plate and long steel products," *Int. J. Struct. Integr.*, vol. 7, no. 5, pp. 645–655, doi: 10.1108/IJSI-09-2015-0039, Jan. 2016.
- [6] I. Setiawati, P., Karno, A. S. B., Hastomo, dan W. Setiawan, "Menggunakan Xception, Transfer Learning, dan Permutasi untuk Meningkatkan Klasifikasi Ketidaktepatan Permukaan Baja," *Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. January, pp. 350–358, 2024.
- [7] P. Cawley, "Non-destructive testing—current capabilities and future directions," *Proc. Inst. Mech. Eng. Part L J. Mater. Des. Appl.*, vol. 215, no. 4, pp. 213–223, doi: 10.1177/146442070121500403, 2001.
- [8] M. Inês Silva, E. Malitckii, T. G. Santos, and P. Vilaça, "Review of conventional and advanced non-destructive testing techniques for detection and characterization of small-scale defects," *Prog. Mater. Sci.*, vol. 138, p. 101155, 2023, doi: <https://doi.org/10.1016/j.pmatsci.2023.101155>.
- [9] A. T. G. Tapeh and M. Z. Naser, "Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices," *Arch. Comput. Methods Eng.*, vol. 30, no. 1, pp. 115–159, doi: 10.1007/s11831-022-09793-w, 2023.
- [10] T. Zhang, P. Pan, J. Zhang, and X. Zhang, "Steel Surface Defect Detection Algorithm Based on Improved YOLOv8n,"

Applied Sciences, vol. 14, no. 12, doi: 10.3390/app141253252024.

- [11] K. Demir, M. Ay, M. Cavas, and F. Demir, "Automated steel surface defect detection and classification using a new deep learning-based approach," *Neural Comput. Appl.*, vol. 35, no. 11, pp. 8389–8406, doi: 10.1007/s00521-022-08112-5, 2023.
- [12] A. C. Walter Reade, "Steel Plate Defect Prediction," kaggle.com, diakses daring pada <https://kaggle.com/competitions/playground-series-s4e3>, 2024.
- [13] Y. Sujatna et al., "Stacked LSTM-GRU Long-Term Forecasting Model for Indonesian Islamic Banks," *Knowl. Eng. Data Sci.*, vol. 6, no. 2, p. 215, doi: 10.17977/um018v6i22023p215-250, 2023.
- [14] N. Aini, W. Hastomo, and R. Yulika Go, "Prediction of Anthropogenic Greenhouse Gas Emissions via Manure Management in Indonesia and Alternative Policies for Indonesian Livestock Development," *J. Renew. Energy Environ.*, doi: 10.30501/jree.2022.354796.1423, 2022.
- [15] A. Géron, "Aprende machine learning con scikit-learn, keras y tensorflow. Conceptos, herramientas y técnicas para conseguir sistemas inteligentes," O'Reilly Media, Inc, 2020.
- [16] Adhitio Satyo Bayangkari Karno, Widi Hastomo, Tri Surawan, Serlia Rafflesia Laman-dasa, Sudarto Usuli, Holmes Rolandy Kapuy, Aji Digdoyo, "Classification of cervical spine fractures using 8 variants EfficientNet with transfer learning," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 6, pp. 7065–7077, doi: 10.11591/ijece.v13i6.pp7065-7077, 2023.
- [17] R. Yulianto, M. S. Rusli, A. Satyo, B. Karno, W. Hastomo, and A. R. Kardian, "Innovative UNET-Based Steel Defect Detection Using 5 Pretrained Models," *Evergreen*, vol. 10, no. 4, pp. 2365–2378, 2023.
- [18] Richard V. McCarthy, Mary M. McCarthy, and Wendy Ceccucci, "*Applying Predictive Analytics*," Springer International Publishing, 2022.
- [19] C. M. Bishop, "Pattern recognition and machine learning," New York: springer, 2006.
- [20] Mark A. Hall and Lloyd A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper," *Proc. twelfth Int. Florida Artif. Intell. Res. Soc. Conf.*, pp. 235–239, 1999.
- [21] W. Hastomo, A. S. Bayangkari Karno, N. Kalbuana, A. Meiriki, and Sutarno, "Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia," *J. Phys. Conf. Ser.*, vol. 1933, no. 1, doi: 10.1088/1742-6596/1933/1/012050, 2021.
- [22] D. Xia, H. Tang, S. Sun, C. Tang, and B. Zhang, "Landslide Susceptibility Mapping Based on the Germinal Center Optimization Algorithm and Support Vector Classification," *Remote Sens.*, vol. 14, no. 11, doi: 10.3390/rs14112707, 2022.