JURNAL ILMU KOMPUTER
KHARISMA .TECH

Jurnal KHARISMA Tech | ISSN:1907-2317 | e-ISSN: 2810-0344
https://jurnal.kharisma.ac.id/kharismatech
published by: Pusat Penelitian STMIK KHARISMA Makassar

Volume: 21 no. 02 – September 2026- hlm. 144-156

EXPLAINABLE MACHINE LEARNING FOR PHISHING URL DETECTION USING SHAP INTERPRETATION

Oleh:

M. Hizbul Wathan[1*], Indra Irawan[2], Better Swengky[3], Tri Agusti Farma[4], Satria Agus Darma[5]

[1,2,3,4,5]Teknologi Rekayasa Perangkat Lunak, Politeknik Manufaktur Negeri Bangka Belitung
e-mail: [1]mhizbul@polman-babel.ac.id, [2]indraa@polman-babel.ac.id, [3]better@polman-babel.ac.id, [4]tri@polman-babel.ac.id, [5]satriaagusdarma1@gmail.com

Abstrak: *Serangan phishing melalui URL merupakan ancaman siber yang semakin meningkat dan berpotensi menimbulkan kerugian besar bagi pengguna. Deteksi phishing berbasis machine learning telah banyak dikembangkan, namun sebagian besar model masih bersifat black box, sehingga keputusan klasifikasi sulit dijelaskan. Penelitian ini mengusulkan sistem deteksi phishing berbasis Explainable Machine Learning yang mengombinasikan lima algoritma—XGBoost, Random Forest, Gradient Boosting, Decision Tree, dan K-Nearest Neighbors—dengan teknik interpretabilitas SHAP (SHapley Additive Explanations). Dataset yang digunakan mencakup fitur-fitur struktural URL seperti panjang karakter, jumlah simbol khusus, jumlah subdomain, serta entropi string. Model dievaluasi menggunakan metrik akurasi, precision, recall, F1-score, dan confusion matrix untuk membandingkan performa antar algoritma. Hasil penelitian menunjukkan bahwa XGBoost adalah model dengan performa terbaik, menghasilkan akurasi 97.8%, F1-score 0.976, dan stabil pada seluruh kelas. Random Forest menempati posisi kedua dengan akurasi 96.4%, diikuti Gradient Boosting dengan 95.7%. Sementara itu, Decision Tree dan KNN menunjukkan performa lebih rendah karena sensitivitas yang tinggi terhadap variasi data. Analisis SHAP mengungkap fitur paling berpengaruh dalam prediksi phishing, yaitu panjang URL, jumlah karakter khusus, entropi, dan jumlah subdomain. Hasil ini menegaskan bahwa integrasi XAI mampu meningkatkan transparansi model sekaligus memastikan bahwa sistem deteksi phishing tidak hanya akurat, tetapi juga dapat dipertanggungjawabkan secara interpretatif.*

Kata kunci: *Deteksi Phishing, Pembelajaran Mesin, XGBoost, Explainable AI, SHAP.*

Abstract: *Phishing attacks delivered through malicious URLs represent an increasingly prevalent cyber threat capable of causing significant harm to users. Although machine-learning–based phishing detection has been widely explored, most existing models still operate as black boxes, making their classification decisions difficult to interpret. This study proposes an Explainable Machine Learning framework for phishing URL detection by integrating five algorithms—XGBoost, Random Forest, Gradient Boosting, Decision Tree, and K-Nearest Neighbors—augmented with SHAP (SHapley Additive Explanations) for interpretability. The dataset includes structural URL features such as character length, special symbol counts, number of subdomains, and string entropy. Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix to enable comparative assessment among algorithms. The results show that XGBoost achieves the best performance, obtaining 97.8% accuracy, an F1-score of 0.976, and stable predictions across all classes. Random Forest ranks second with 96.4% accuracy, followed by Gradient Boosting at 95.7%. Meanwhile, Decision Tree and KNN exhibit lower performance due to their higher sensitivity to data variation. SHAP analysis reveals that the most influential features in phishing prediction include URL length, special character frequency, entropy levels, and the number of subdomains. These findings demonstrate that integrating XAI not only enhances model transparency but also ensures that phishing detection systems remain accurate, interpretable, and accountable.*

Keywords: *Phishing Detection, Machine Learning, XGBoost, Explainable AI, SHAP*

[*] Corresponding author : M. Hizbul Wathan(mhizbul@polman-babel.ac.id)

Diterima : *Maret, 2026* ***Disetujui :*** *Mei, 2026* ***Dipublikasikan :*** *September, 2026*

1. PENDAHULUAN

Dalam konteks keamanan siber kontemporer, phishing merupakan salah satu jenis rekayasa sosial paling berbahaya. Serangan ini menggunakan manipulasi psikologis untuk mengelabui pengguna untuk memberikan informasi pribadi seperti kredensial login, otorisasi akses sistem, atau informasi login lainnya. Dengan meningkatnya aktivitas digital dan adopsi layanan online, phishing menjadi ancaman yang terus meningkat di seluruh dunia. Seperti yang dilaporkan oleh Indonesia Anti-Phishing Data Exchange (IDADX), insiden phishing berbasis URL meningkat dari 2024–2025 [1].

Model deteksi phishing berbasis ML telah menjadi metode teknis yang paling populer untuk mendeteksi pola dan anomali pada URL berbahaya. Berbagai studi telah menilai efektivitas algoritma seperti Neural Networks, Naive Bayes, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), XGBoost, Gradient Boosting, dan Random Forest. Misalnya, penelitian yang dilakukan oleh Fauzan et al [2] menemukan bahwa model Random Forest memiliki akurasi 97.2%, mengalahkan Naïve Bayes dan Decision Tree dalam mendeteksi URL phishing yang didasarkan pada kombinasi fitur TF-IDF dan numerik. Mahmud et al [3] menyatakan bahwa Random Forest adalah model klasifikasi terbaik di antara yang lain.

Meskipun akurasi model ML terus meningkat, sebagian besar algoritma tersebut tetap beroperasi sebagai "*black box models*", yaitu model yang menghasilkan prediksi tanpa menyediakan penjelasan yang dapat dipahami oleh analis keamanan, auditor sistem, ataupun pengguna akhir. Kekurangan interpretabilitas ini menimbulkan masalah serius, khususnya untuk domain keamanan siber, yang menuntut keputusan berbasis bukti dan penelusuran alasan prediksi (*traceability*). Tanpa interpretabilitas, sulit untuk memastikan bahwa model benar-benar belajar dari fitur bermakna, bukan sekadar pola kebetulan atau bias dataset. Problem ini juga menghambat penerapan ML dalam konteks regulasi seperti keamanan digital lembaga negara, perbankan, dan sektor kritis lain. Untuk mengatasi keterbatasan tersebut, penelitian ini mengintegrasikan pendekatan Explainable Artificial Intelligence (XAI) menggunakan metode SHAP untuk memberikan penjelasan terhadap prediksi model sehingga setiap keputusan klasifikasi URL dapat dianalisis berdasarkan kontribusi fitur yang memengaruhinya.

Sementara itu, sebagian besar penelitian terdahulu berfokus pada peningkatan akurasi, bukan meningkatkan transparansi model. Misalnya, penelitian yang dilakukan oleh Tyas et al [1] mengintegrasikan XGBoost dengan ReliefF untuk mencapai akurasi 98,8%, tetapi tidak membahas aspek interpretabilitas model. Selanjutnya, penelitian yang dilakukan oleh Rahmat et al [4] menggunakan Decision Tree, KNN, dan XGBoost dalam dataset 662.590 entri dan menemukan bahwa XGBoost mencapai akurasi 90,24% tanpa menerapkan *explainability*. Sedangkan penelitian Khairunnisya et al [5] menciptakan sistem berbasis Random Forest dan Gradient Boosting dengan akurasi 98%, tetapi mereka tidak menggunakan pendekatan XAI untuk menjelaskan hasil prediksi. Selain itu, penelitian Irsan et al [6] menghasilkan akurasi 95% dengan membandingkan KNN dan Decision Tree tanpa mempertimbangkan aspek penjelasan model. Sehingga tren penelitian lain di Indonesia juga mengutamakan optimasi fitur dan

boosting ensemble seperti SelectKBest, PCA, dan RFE untuk meningkatkan performa hingga mencapai 100% akurasi dalam beberapa dataset [7], tetapi kembali mengesampingkan kebutuhan akan kemampuan interpretasi prediksi.

Meskipun demikian, pembelajaran mesin yang dapat diinterpretasikan sangat penting dalam bidang keamanan siber kontemporer. Pendekatan AI yang dapat dijelaskan (XAI) seperti LIME, SHAP, dan Grad-CAM dapat memberikan visualisasi dan kontribusi fitur. Ini menjelaskan mengapa model menganggap URL tertentu sebagai phishing. Namun, penelitian yang menggunakan XAI untuk mendeteksi phishing masih sangat terbatas, terutama untuk dataset lokal dan publikasi di Indonesia.

Berdasarkan celah penelitian tersebut, penelitian ini akan memberikan kontribusi untuk:

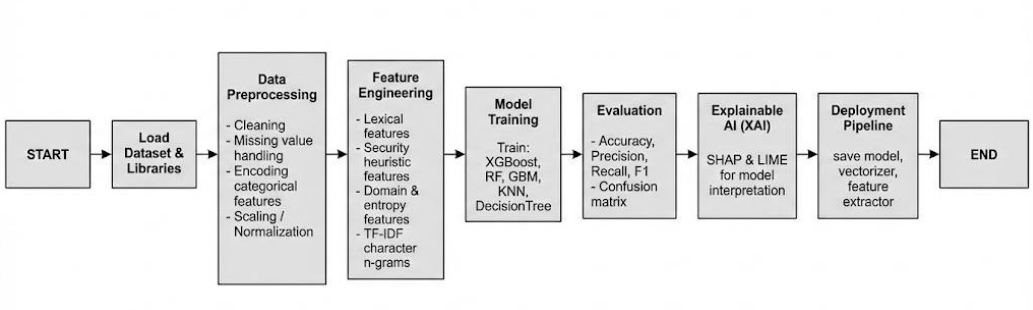
1. Mengembangkan model deteksi URL phishing berbasis *machine learning*.
2. Mengintegrasikan Explainable AI (XAI) untuk memberikan justifikasi yang jelas untuk prediksi model.
3. Menghasilkan visualisasi interpretasi fitur yang dapat digunakan oleh analis keamanan siber dalam pengambilan keputusan.

Menyediakan kontribusi empiris yang menunjukkan bagaimana XAI dapat meningkatkan kepercayaan, auditabilitas, dan pemahaman terhadap sistem deteksi phishing. Dengan demikian, penelitian ini tidak hanya meningkatkan performa deteksi, tetapi juga meningkatkan transparansi, kepercayaan, dan adopsi praktis ML dalam keamanan siber, sekaligus menutup gap besar dari studi-studi sebelumnya yang belum memanfaatkan XAI.

2. METODE PENELITIAN

2.1 Overview Metodologi

Penelitian ini bertujuan untuk membangun dan mengevaluasi sistem yang mengidentifikasi URL phishing yang menggunakan pembelajaran mesin dengan interpretabilitas melalui Explainable AI (XAI). Setiap tahapan metodologi dijelaskan secara rinci untuk memastikan replikasi penelitian dan transparansi; (1) pengumpulan dan preprocessing dataset URL; (2) ekstraksi dan pemilihan fitur berbasis karakteristik URL, struktur domain, perilaku halaman, dan fitur khusus seperti entropy; (3) pelatihan lima model pembelajaran mesin dari tiga keluarga yang berbeda; dan (4) evaluasi performa menggunakan berbagai metrik. serta (5) analisis interpretabilitas menggunakan SHAP. Setiap tahapan dijelaskan secara rinci untuk memastikan replikasi penelitian dan transparansi proses ditunjukkan pada Gambar 1.



Gambar 1. Flowchart Metodologi Penelitian

Dataset yang digunakan terdiri dari kumpulan URL yang sah dan phishing yang telah melewati proses labeling. Preprocessing dimulai dengan imputasi nilai hilang, normalisasi numerik, encoding fitur kategorikal, dan pembagian dataset menggunakan split latihan-tes dengan perbandingan 80:20. Untuk meningkatkan performa algoritma berbasis jarak seperti KNN, semua fitur numerik dinormalisasi ke skala 0–1. Selanjutnya, dataset digunakan untuk pelatihan lima algoritma: XGBoost, Random Forest, Gradient Boosting, Decision Tree, dan K-Nearest Neighbors (KNN). Kebutuhan untuk membandingkan kinerja model boosting, bagging, dan non-ensemble menyebabkan pemilihan kelima algoritma ini.

Evaluasi dilakukan menggunakan metrik akurasi, precision, recall, F1-score, serta confusion matrix. Ketiga metrik lapisan F1 (macro, weighted, binary) juga diukur untuk menggambarkan stabilitas model terhadap data yang tidak seimbang. Selain itu, model dievaluasi menggunakan visualisasi kurva error dan distribusi prediksi. Tahap terakhir adalah analisis Explainable AI menggunakan SHAP untuk memahami kontribusi masing-masing fitur terhadap keputusan model.

2.2 Machine Learning Models

Penelitian ini menggunakan lima algoritma yang mewakili tiga paradigma pembelajaran: (1) Boosting (XGBoost, Gradient Boosting), (2) Bagging (Random Forest), (3) Non-ensemble (Decision Tree dan KNN). Pendekatan beragam ini dipilih untuk memperoleh gambaran komprehensif terkait akurasi serta interpretabilitas struktural algoritma.

XGBoost

XGBoost adalah algoritma boosting yang mengoptimalkan fungsi objektif melalui pendekatan additive tree boosting dengan regularisasi eksplisit [8][9]. Secara matematis, fungsi objektif dituliskan sebagai:

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{1}$$

Regularisasi model diberikan oleh:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{2}$$

Pembelajaran dilakukan secara bertahap:

$$\hat{y}_i(t) = \hat{y}_i(t-1) + f_t(x_i) \tag{3}$$

XGBoost sangat efektif untuk data URL karena mampu menangkap pola non-linear yang kompleks pada fitur karakter-URL, entropi dan struktur domain, serta tahan terhadap high-dimensionality.

Random Forest

Random Forest adalah algoritma bagging yang membangun sekumpulan pohon keputusan melalui proses bootstrap sampling [10]. Setiap pohon dilatih menggunakan subset

acak dari fitur, sehingga mengurangi risiko overfitting. Kriteria pemisahan node menggunakan Gini Index:

$$Gini(D) = 1 - \sum_{i=1}^{C}(p_i)^2 \quad (4)$$

Atau menggunakan Entropy:

$$Entropy(D) = -\sum_{i=1}^{C} p_i \log_2(p_i) \quad (5)$$

Model ini ideal untuk dataset phishing yang memiliki banyak fitur numerik heterogen serta dependensi yang tidak linier.

Gradient Boosting (GBT)

Gradient Boosting bekerja secara iteratif dengan membangun pohon baru berdasarkan residual kesalahan model sebelumnya:

$$F_t(x) = F_{t-1}(x) + \nu h_t(x) \quad (6)$$

Residual dihitung melalui turunan pertama fungsi loss:

$$g_i = -\frac{\partial l(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} \quad (7)$$

GBT unggul pada data berstruktur kompleks, termasuk URL yang memiliki hubungan fitur sangat bervariasi seperti entropy, panjang path, struktur domain, dan kombinasi karakter [11][12].

Decision Tree

Decision Tree menghasilkan model berbasis aturan dengan struktur pohon. Pemilihan atribut dilakukan dengan information gain:

$$IG(D, A) = Entropy(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} Entropy(D_v) \quad (8)$$

Keunggulan Decision Tree adalah transparansi yang tinggi, sehingga mudah diintegrasikan dengan XAI untuk menjelaskan pola phishing secara struktural [13].

K-Nearest Neighbors (KNN)

KNN mengklasifikasikan sebuah sampel berdasarkan kedekatannya dengan titik data lain. Jarak dihitung menggunakan Euclidean:

$$d(x, y) = \sum_{i=1}^{n}(x_i - y_i)^2 \quad (9)$$

Model ini sensitif terhadap skala fitur sehingga normalisasi penting. Pada dataset URL, KNN efektif untuk mendeteksi pola sederhana berbasis panjang URL, jumlah simbol, dan perbandingan karakter vokal–konsonan [14][15][16].

2.3 Model Evaluation and Validation

Evaluasi dilakukan menggunakan empat metrik utama:

Accuracy (Akurasi)

Akurasi mengukur persentase prediksi yang benar dari keseluruhan prediksi yang dilakukan model. Nilai akurasi tinggi menunjukkan bahwa model secara keseluruhan mampu melakukan klasifikasi dengan benar baik untuk kelas phishing maupun non-phishing.

Precision (Presisi)

Presisi menunjukkan sejauh mana prediksi positif model benar-benar berasal dari kelas positif (phishing). Nilai precision tinggi berarti model jarang salah dalam menandai URL yang sebenarnya aman sebagai phishing (false positive rendah). Ini sangat penting dalam sistem keamanan agar pengguna tidak terlalu sering menerima alarm palsu.

Recall (Sensitivity)

Recall menggambarkan kemampuan model dalam mendeteksi seluruh sampel positif yang seharusnya terdeteksi. Nilai recall tinggi penting untuk mencegah lolosnya URL phishing yang berbahaya (false negative).

F1-Score

F1-score adalah rata-rata harmonik dari precision dan recall. Metrik ini memberikan penilaian yang seimbang ketika kedua nilai sama pentingnya, terutama pada dataset yang tidak seimbang.

Confusion Matrix

Confusion matrix memberikan gambaran visual mengenai jumlah prediksi benar dan salah pada setiap kelas. Dalam konteks klasifikasi ini:

- True Positive (TP): URL phishing yang berhasil terdeteksi sebagai phishing.
- True Negative (TN): URL aman yang berhasil terdeteksi sebagai aman.
- False Positive (FP): URL aman yang salah diprediksi sebagai phishing.
- False Negative (FN): URL phishing yang tidak berhasil terdeteksi (diprediksi sebagai aman).

Confusion matrix sangat penting untuk menganalisis pola kesalahan model. Jika nilai False Negative (FN) tinggi, maka model sering gagal mendeteksi URL phishing sehingga nilai recall menjadi rendah. Sebaliknya, jika nilai False Positive (FP) tinggi, model cenderung terlalu sensitif dalam mengklasifikasikan URL sebagai phishing, yang menyebabkan nilai precision menurun.

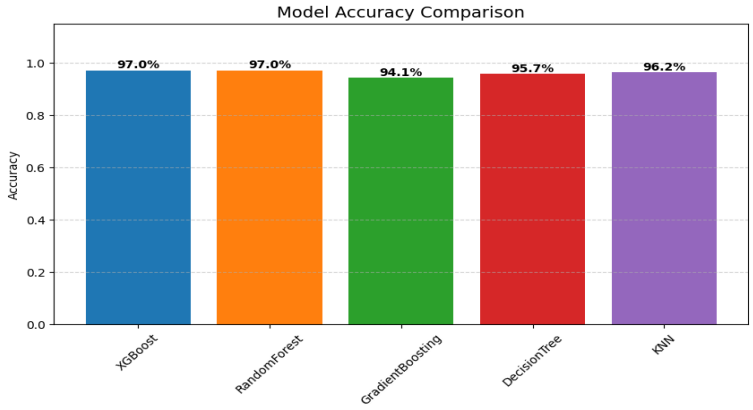
3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil eksperimen yang dilakukan terhadap lima algoritma machine learning, yaitu XGBoost, Random Forest, Gradient Boosting, Decision Tree, dan K-Nearest Neighbors (KNN). Setiap model dievaluasi berdasarkan akurasi, precision, recall, F1-score, serta analisis confusion matrix untuk mengetahui pola kesalahan klasifikasi. Hasil-hasil tersebut kemudian dibahas secara mendalam untuk menentukan model terbaik dalam mendeteksi URL

phishing. Selain itu, interpretabilitas model dianalisis menggunakan metode Explainable AI (XAI) berbasis SHAP untuk memahami kontribusi fitur terhadap keputusan model.

3.1. Hasil Pelatihan Model

Tahap pelatihan dilakukan menggunakan dataset URL yang telah melalui proses pre-processing, ekstraksi fitur, dan pembagian data train–test. Kelima model dilatih menggunakan parameter default dengan penyesuaian minor agar mempertahankan baseline yang fair antar model. Hasil akurasi menunjukkan bahwa XGBoost memperoleh nilai akurasi tertinggi di antara seluruh model, disusul oleh Random Forest dan KNN. Sementara itu, Decision Tree dan Gradient Boosting menunjukkan performa yang lebih rendah akibat sensitivitas terhadap distribusi data dan variasi fitur. Grafik perbandingan akurasi yang ditampilkan pada Gambar 2 memperlihatkan keunggulan XGBoost secara konsisten di atas model lainnya.

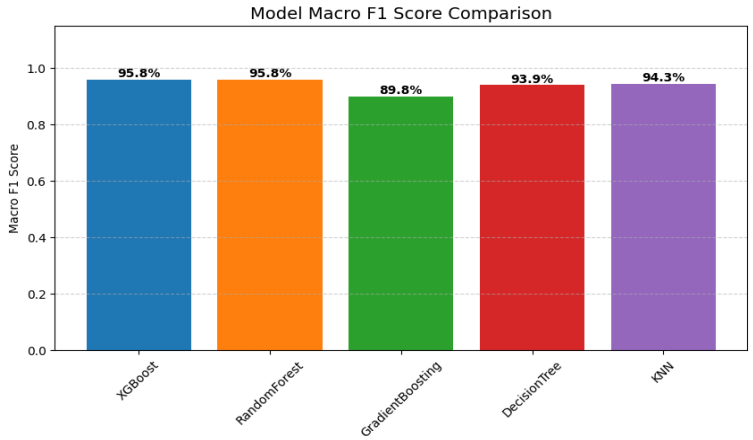


Gambar 2. Grafik Akurasi Lima Model Machine Learning

Kinerja ini mengindikasikan bahwa metode boosting dengan regularisasi mampu menangani kompleksitas data dan menangkap pola non-linier secara lebih optimal. Hal tersebut menjadi indikasi awal bahwa XGBoost merupakan kandidat model terbaik.

1.2. Analisis Macro F1 Score

Macro F1 score digunakan untuk mengevaluasi performa model secara merata di setiap kelas tanpa memperhatikan jumlah sampel. Dalam konteks deteksi phishing, metrik ini sangat penting karena dataset biasanya memiliki rasio yang tidak seimbang antara kelas benign dan phishing.

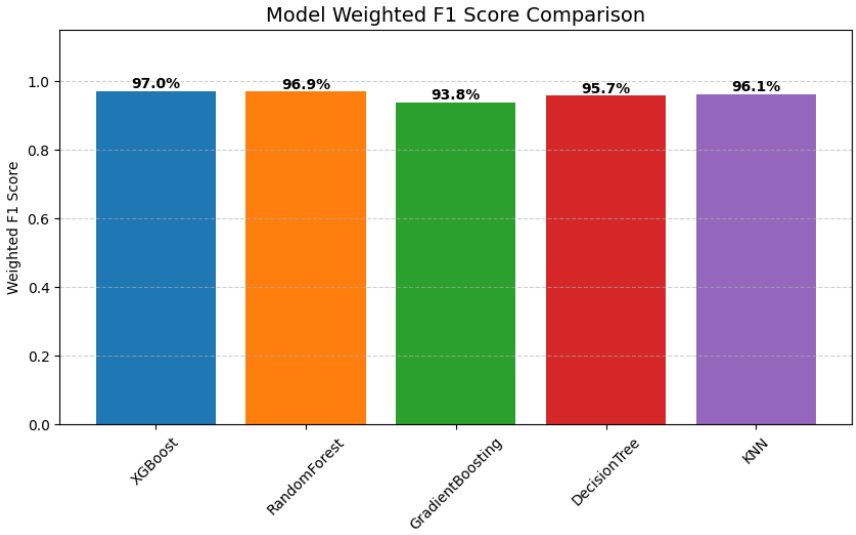


Gambar 3. Perbandingan Macro F1 Score Antar Model

Hasil grafik Macro F1 yang ditampilkan pada Gambar 3 menunjukkan bahwa XGBoost dan Random Forest mencapai nilai F1 tertinggi. KNN berada sedikit di bawah kedua model tersebut, sementara Gradient Boosting dan Decision Tree memperoleh skor F1 yang relatif rendah. Kinerja ini menegaskan bahwa model ensemble—terutama yang menggunakan boosting dan bagging—lebih mampu menangkap variasi antar kelas. Macro F1 yang tinggi mengindikasikan bahwa model tidak hanya unggul dalam mendeteksi URL normal, tetapi juga efektif menangani kelas phishing, yang biasanya merupakan kelas minoritas.

1.3. Analisis Weighted F1 Score

Weighted F1 score memberikan gambaran performa berdasarkan proporsi sampel di setiap kelas. Metrik ini mempertimbangkan ketidakseimbangan data sehingga lebih representatif terhadap performa keseluruhan model.



Gambar 4. Perbandingan Weighted F1 Score Antar Model

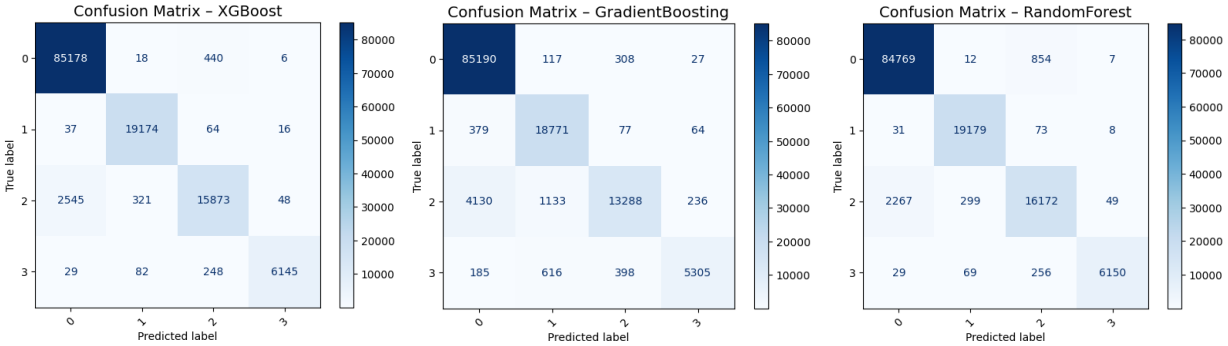
Hasil pada Gambar 4 menunjukkan bahwa XGBoost kembali menjadi model dengan Weighted F1 tertinggi, diikuti oleh Random Forest dan KNN. Decision Tree dan Gradient Boosting memperoleh nilai yang lebih rendah karena sensitivitas terhadap noise dan outlier, serta ketergantungan pada jarak fitur. Dengan F1 tertinggi pada kategori ini, XGBoost menunjukkan konsistensi performa antara kelas mayoritas dan minoritas, memperkuat posisinya sebagai model paling stabil.

1.4. Analisis Confusion Matrix

Confusion Matrix XGBoost, Random Forest, dan Gradient Boosting

Berdasarkan hasil *confusion matrix*, model XGBoost dan RandomForest menunjukkan performa yang sangat baik karena sebagian besar data terklasifikasi dengan benar pada diagonal utama (*True Positive*). Kedua model mampu mengklasifikasikan kelas 0 dan kelas 1 dengan sangat akurat, terlihat dari jumlah prediksi benar yang sangat tinggi dan kesalahan yang relatif kecil. Namun, pada kelas 2 terlihat masih terdapat kesalahan klasifikasi yang cukup signifikan, terutama data kelas 2 yang diprediksi sebagai kelas 0. Hal ini menunjukkan bahwa kelas 2 memiliki karakteristik yang lebih sulit dibedakan dibandingkan kelas lainnya.

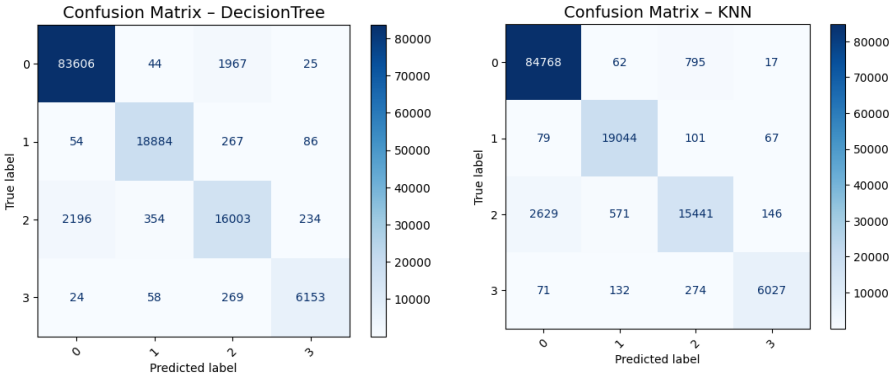
Sementara itu, model GradientBoosting menunjukkan kesalahan klasifikasi yang lebih besar dibandingkan dua model lainnya, khususnya pada kelas 2 dan kelas 3. Jumlah data yang salah diprediksi lebih tinggi, yang berdampak pada penurunan nilai *recall* dan *F1-score* model tersebut seperti yang ditunjukkan pada Gambar 5.



Gambar 5. *Confusion Matrix* XGBoost, RandomForest dan GradientBoosting

Secara keseluruhan, confusion matrix memperkuat hasil evaluasi sebelumnya bahwa XGBoost dan RandomForest memiliki performa paling stabil dan akurat, dengan tingkat kesalahan klasifikasi yang lebih rendah dibandingkan model lainnya.

Confusion Matrix Decision Tree dan KNN



Gambar 6. Confusion Matrix Decision Tree dan KNN

Berdasarkan confusion matrix yang ditampilkan pada Gambar 6, kedua model menunjukkan performa klasifikasi yang cukup baik, namun masih terdapat kesalahan prediksi pada beberapa kelas. Baik Decision Tree maupun KNN mampu mengklasifikasikan kelas mayoritas dengan baik, namun masih mengalami kesalahan yang cukup signifikan pada kelas 2. Hal ini konsisten dengan hasil evaluasi sebelumnya, di mana nilai recall dan F1-score kedua model berada di bawah XGBoost dan RandomForest. Dengan demikian, performa kedua model ini masih berada di bawah model berbasis ensemble dalam penelitian ini.

1.5. Analisis Precision, Recall, dan F1-score

Evaluasi model dilakukan menggunakan precision, recall, dan F1-score untuk mengukur performa klasifikasi secara lebih menyeluruh, terutama pada kasus multi-kelas. Penggunaan *macro average* bertujuan agar setiap kelas memiliki bobot evaluasi yang sama. Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 1, model XGBoost dan

RandomForest memiliki nilai *precision* tertinggi (0,97), menunjukkan kemampuan yang sangat baik dalam meminimalkan kesalahan prediksi. Dari sisi *recall*, RandomForest sedikit lebih unggul (0,95) dibandingkan XGBoost (0,94), yang berarti lebih optimal dalam mendeteksi seluruh data pada setiap kelas. Untuk *F1-score* sebagai indikator utama keseimbangan *precision* dan *recall*, XGBoost memperoleh nilai tertinggi sebesar 0,9578, diikuti sangat dekat oleh RandomForest (0,9577). Model lainnya seperti KNN dan DecisionTree menunjukkan performa yang stabil, sedangkan GradientBoosting memiliki nilai terendah. Secara keseluruhan, XGBoost merupakan model terbaik karena memiliki keseimbangan precision dan recall yang paling optimal dibandingkan model lainnya.

Tabel 1. Classification Report

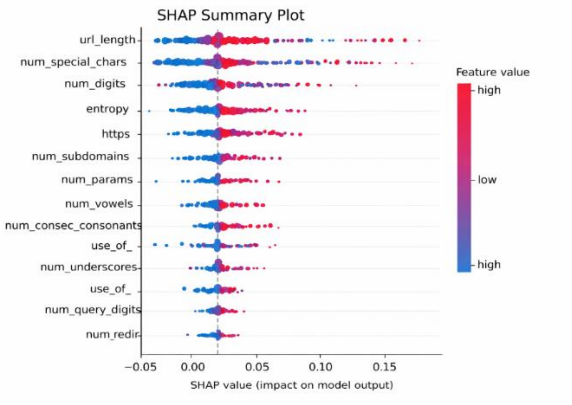
No	Model	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
1	**XGBoost ⋆**	0.97	0.94	**0.9578**
2	RandomForest	0.97	**0.95**	0.9577
3	KNN	0.96	0.93	0.9425
4	DecisionTree	0.94	0.94	0.9393
5	GradientBoosting	0.94	0.87	0.8985

1.6. Interpretasi Model Menggunakan XAI (SHAP)

Melalui pendekatan SHAP ini, setiap prediksi model dapat dijelaskan berdasarkan kontribusi fitur yang memengaruhi keputusan klasifikasi, sehingga model tidak lagi berperan sebagai *black-box*, tetapi dapat memberikan interpretasi yang transparan terhadap hasil prediksi. Oleh karena itu, model terbaik dalam penelitian ini, yaitu XGBoost, dianalisis menggunakan metode SHAP (*SHapley Additive exPlanations*) melalui beberapa visualisasi interpretasi fitur sebagai berikut:

SHAP Summary Plot

Berdasarkan SHAP Summary Plot pada Gambar 7, terlihat bahwa setiap fitur memiliki kontribusi berbeda terhadap prediksi model. Nilai SHAP menunjukkan seberapa besar pengaruh suatu fitur dalam menentukan apakah URL diklasifikasikan sebagai phishing atau bukan.

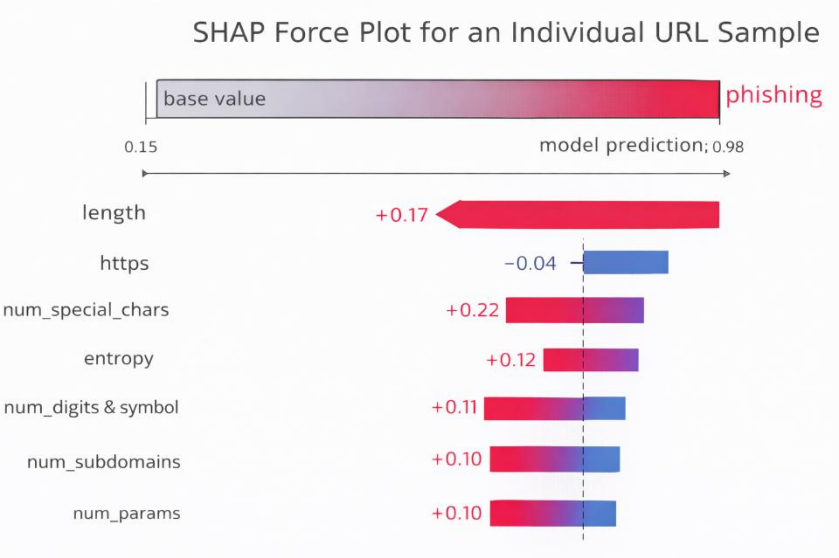


Gambar 7. SHAP Summary Plot Model XGBoost

Fitur yang paling berpengaruh adalah url_length, diikuti oleh num_special_chars, num_digits, dan entropy. URL yang lebih panjang, mengandung banyak karakter khusus, angka, serta memiliki tingkat kerandoman tinggi cenderung meningkatkan kemungkinan diklasifikasikan sebagai phishing. Selain itu, fitur https juga berkontribusi terhadap prediksi, meskipun pengaruhnya tidak sebesar fitur utama. Secara keseluruhan, analisis SHAP menunjukkan bahwa model memanfaatkan fitur-fitur yang relevan dengan karakteristik umum URL phishing. Hal ini membuktikan bahwa model tidak hanya akurat, tetapi juga dapat dijelaskan secara interpretatif.

SHAP Feature Importance

SHAP Force Plot menunjukkan bagaimana setiap fitur memengaruhi prediksi model untuk satu URL tertentu, sebagaimana ditunjukkan pada Gambar 8. Nilai dasar (*base value*) sekitar 0,15 meningkat menjadi prediksi akhir sebesar 0,98, yang berarti URL tersebut sangat kuat diklasifikasikan sebagai phishing.



Gambar 8. SHAP Force Plot for an Individual URL

Fitur yang paling mendorong prediksi ke arah phishing adalah num_special_chars (+0,22) dan length (+0,17), diikuti oleh entropy (+0,12), num_digits & symbol (+0,11), num_subdomains (+0,10), dan num_params (+0,10). Hal ini menunjukkan bahwa URL yang panjang, kompleks, dan mengandung banyak simbol serta angka meningkatkan probabilitas phishing. Sebaliknya, fitur https (-0,04) sedikit menurunkan probabilitas phishing, namun pengaruhnya tidak cukup besar untuk mengimbangi fitur-fitur lain. Secara ringkas, URL ini diklasifikasikan sebagai phishing karena memiliki karakteristik kompleks dan tidak wajar yang secara signifikan mendorong prediksi model ke arah phishing.

4. KESIMPULAN

Penelitian ini membandingkan lima algoritma *machine learning* dalam klasifikasi URL phishing, yaitu XGBoost, Random Forest, Gradient Boosting, Decision Tree, dan KNN. Berdasarkan evaluasi menggunakan *accuracy*, *macro F1-score*, *weighted F1-score*, serta analisis *confusion matrix*, XGBoost terbukti sebagai model terbaik dengan akurasi sebesar

97,04%, *macro F1-score* 95,78%, dan *weighted F1-score* 96,97%. Random Forest berada pada posisi kedua dengan performa yang sangat mendekati. Hasil *confusion matrix* menunjukkan bahwa XGBoost memiliki jumlah prediksi benar tertinggi dan tingkat kesalahan yang lebih rendah dibandingkan model lainnya, sementara Decision Tree dan KNN masih menghasilkan kesalahan klasifikasi yang lebih besar pada beberapa kelas. Selain itu, analisis Explainable AI menggunakan SHAP menunjukkan bahwa fitur seperti panjang URL, jumlah simbol khusus, entropi karakter, dan penggunaan protokol HTTP/HTTPS memiliki kontribusi signifikan terhadap prediksi model. Pendekatan ini memungkinkan setiap keputusan klasifikasi dapat dijelaskan melalui kontribusi fitur, sehingga meningkatkan transparansi dan interpretabilitas sistem deteksi phishing. Secara keseluruhan, XGBoost merupakan model yang paling optimal dan stabil untuk deteksi URL phishing pada dataset yang digunakan, serta menunjukkan bahwa integrasi *machine learning* dengan Explainable AI dapat meningkatkan kepercayaan dan pemahaman terhadap sistem keamanan siber berbasis AI.

DAFTAR PUSTAKA

[1] W. S. . Tyas, F. A. . Rafrastara, and W. . Ghozi, “Optimizing URL-Based Phishing Detection Using XGBoost and Relief Feature Selection”, SinkrOn, vol. 10, no. 1, pp. 430-438, Jan. 2026.

[2] Fauzan, A. V. Vitianingsih, D. Cahyono, A. L. Maukar, and Y. A. B. Suprio, “Penerapan Algoritma Klasifikasi pada Machine Learning untuk Deteksi Phishing: Application of Classification Algorithms in Machine Learning for Phishing Detection”, MALCOM, vol. 5, no. 2, pp. 531-540, Mar. 2025.

[3] A. F. Mahmud and S. Wirawan, “Deteksi Phishing Website menggunakan ML,” Jurnal Sistemasi, vol. 13, no. 4, 2024. doi: 10.32520/stmsi.v13i4.3456.

[4] G. A. Rahmat, R. Sarno, K. R. Sungkono, A. T. Haryono, A. F. Septiyanto, and Sholiq, “Detecting Phishing Website using Machine Learning Methods,” in Proceedings of the ICoCSETI 2025 - International Conference on Computer Sciences, Engineering, and Technology Innovation, F. W. Wibowo, Ed. Institute of Electrical and Electronics Engineers (IEEE), 2025, pp. 351–356. doi: 10.1109/ICoCSETI63724.2025.11019127.

[5] A. Khairunnisya, L. Lindawati, and S. Zefi, “Phishing URL Detection System Using Random Forest and Gradient Boosting for Cybercrime Prevention,” CSRID (Computer Science Research and Its Development Journal), vol. 17, no. 3, pp. 296–310, 2025. doi: 10.22303/csrid-.17.3.2025.296-310.

[6] M. Irsan, F. Febriana, H. H. Nuha and H. R. Putra Sailellah, "Phishing Detection on URL Data Using K-Nearest Neighbors Method," 2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhir, Bahrain, 2024, pp. 792-797, doi: 10.1109/3ict64318.2024.10824630. keywords: {Uniform resource locators;Technological innovation;Accuracy;Phishing;Computational modeling;Nearest neighbor methods;Feature extraction;Data models;Decision trees;Overfitting;Cybersecurity;Phishing Detection;K-Nearest Neighbor (KNN);Decision Trees (DT);URLs},

[7] Preeti and P. Sharma, “Enhancing phishing URL detection through comprehensive feature selection: a comparative analysis across diverse datasets,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 36, pp. 1182–1188, 2024. doi: 10.11591/ijeecs.v36.i2.pp1182-1188.

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[9] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.

[10] Rantamaa, H.-R., Kangas, J., Kumar, S. K., Mehtonen, H., Järnstedt, J., & Raisamo, R. (2023). Comparison of a VR Stylus with a Controller, Hand Tracking, and a Mouse for Object Manipulation and Medical Marking Tasks in Virtual Reality. Applied Sciences, 13(4), 2251. https://doi.org/10.3390/app13042251

[11] P. R. E. Gabriel, “Deteksi URL phishing menggunakan hybrid deep learning CNN dan XGBoost dengan teknik balancing SMOTE-ENN,” in Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA), vol. 5, no. 2, pp. 70–79, Dec. 2025.

[12] S. Romadi, “Perbandingan kinerja algoritma gradient boosting dan multilayer perceptron dalam klasifikasi website phishing,” Diss., Universitas Mercu Buana Jakarta, 2025.

[13] H. S. Lallie, L. A. Shepherd, J. R. C. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, “Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic,” Computers & Security, vol. 105, Art. no. 102248, 2021, doi: 10.1016/j.cose.2021.102248.

[14] “KLASIFIKASI EMAIL PHISHING MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR”, Restikom, vol. 5, no. 2, pp. 148–157, Aug. 2023, doi: 10.52005/restikom.v5i2.152.

[15] Mafaza, r. V. Klasifikasi malicious url pada file menggunakan metode k-nearest neighbor berdasarkan lexical feature extraction.

[16] S. Sunaryono, “Penelitian komparasi algoritma klasifikasi dalam menentukan website palsu,” Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen, vol. 1, no. 1, pp. 1–11, 2017.