Jurnal Teknologi Informasi dan Multimedia

p-ISSN : 2715-2529

e-ISSN : 2684-9151

https://journal.sekawan-org.id/index.php/jtim

Implementasi Model *Machine Learning* untuk Deteksi *Phishing* dengan Pendekatan Ekstraksi Fitur yang Dioptimalkan

Adam Pradana [1,*], Susanto [1]

[1] Program Studi Teknik Informatika, Universitas Semarang, Indonesia

* Korespondensi: adampradanaadam@gmail.com

Sitasi: A. Pradana and S. Susanto, "Implementasi Model *Machine Learning* untuk Deteksi *Phishing* dengan Pendekatan Ekstraksi Fitur yang Dioptimalkan". Jurnal Teknologi Informasi Dan Multimedia, vol. 8, no. 1, pp. 27-40, 2026. https://doi.org/10.35746/jtim.v8i1.881

Diterima: 25-10-2025
Direvisi: 11-12-2025
Disetujui: 19-12-2025

Abstract: Phishing is a common form of cybercrime used by digital criminals to steal sensitive information such as passwords, personal data, and financial details through fake websites designed to resemble legitimate pages. However, conventional detection methods such as blacklists and manual inspection are currently considered ineffective due to their static nature, often failing to recognize new, evolving and increasingly sophisticated attack patterns. To address this issue, this study developed a machine learning-based phishing detection model focused on improving the accuracy and efficiency of identifying malicious sites. This model applies an optimized feature extraction technique to enable the system to analyze URL characteristic patterns more comprehensively and targeted. The research dataset was taken from the Kaggle platform, which provides a dataset of phishing and benign URLs with a high reputation. The data was then processed through normalization, cleaning, and extraction of important features such as URL structure and domain attributes. The classification process was carried out using an ensemble learning approach that combines four popular algorithms: Random Forest, Gradient Boosting, Logistic Regression, and AdaBoost through a soft voting mechanism. The evaluation results show that the proposed model has excellent performance with an accuracy of 98.10%, a precision of 97.81%, a recall of 93.90%, an F1-Score of 95.82%, and a ROC-AUC of 98.62%. These findings confirm that the ensemble approach with optimized features has great potential for application in artificial intelligence-based cybersecurity systems capable of adaptive and real-time phishing detection.

Keywords: Phishing Detection; Machine Learning; Feature Extraction; URL Detection

Abstrak: Phishing merupakan salah satu bentuk kejahatan siber yang umum digunakan oleh pelaku kriminal digital untuk mencuri informasi sensitif seperti kata sandi, data pribadi, dan rincian keuangan melalui situs palsu yang dirancang menyerupai halaman resmi. Namun, metode deteksi konvensional seperti *blacklist* maupun inspeksi manual saat ini dinilai kurang efektif karena sifatnya yang statis, sehingga sering kali gagal mengenali pola serangan baru yang terus berevolusi dan semakin canggih. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan model deteksi phishing berbasis machine learning yang berfokus pada peningkatan akurasi dan efisiensi dalam mengenali situs berbahaya. Model ini menerapkan teknik ekstraksi fitur yang dioptimalkan agar sistem mampu menganalisis pola karakteristik URL secara lebih komprehensif dan terarah. Dataset penelitian diambil dari platform Kaggle yang menyediakan kumpulan data URL phishing dan benign dengan reputasi tinggi. Data kemudian diproses melalui tahap normalisasi, pembersihan, serta ekstraksi fitur penting seperti struktur URL, dan atribut domain. Proses klasifikasi dilakukan dengan pendekatan ensemble learning yang mengombinasikan empat algoritma popular Random Forest, Gradient Boosting, Logistic Regression, dan AdaBoost melalui mekanisme soft voting. Hasil evaluasi menunjukkan bahwa model yang diusulkan memiliki kinerja sangat baik dengan akurasi 98,10%, precision 97,81%, recall 93,90%, F1-Score 95,82%, dan ROC-AUC 98,62%.

Temuan ini menegaskan bahwa pendekatan ensemble dengan fitur teroptimasi berpotensi besar untuk diterapkan pada sistem keamanan siber berbasis kecerdasan buatan yang mampu mendeteksi phishing secara adaptif dan real-time.

Kata kunci: Deteksi Phishing; Machine Learning; Ekstraksi Fitur; Deteksi URL

1. Pendahuluan

Phishing merupakan salah satu bentuk kejahatan dunia maya yang berkembang pesat dan kerap memanfaatkan teknik rekayasa sosial dengan cara meniru tampilan situs resmI [1]. Tujuan utama dari praktik ini adalah memperoleh informasi sensitif, seperti data login, identitas pribadi, hingga rincian keuangan korban. Laporan global menunjukkan bahwa frekuensi serangan phishing meningkat setiap tahunnya seiring dengan intensifikasi aktivitas digital masyarakat modern. Hal ini menjadikan phishing sebagai ancaman serius yang tidak hanya menimbulkan kerugian finansial, tetapi juga berpotensi merusak reputasi individu maupun lembaga yang menjadi sasarannya. Metode deteksi phishing konvensional, seperti penggunaan blacklist maupun inspeksi manual, semakin dipandang tidak efektif menghadapi variasi serangan yang semakin canggih. Oleh karena itu, penerapan pendekatan machine learning mulai banyak diteliti, mengingat kemampuannya dalam mengenali pola data yang kompleks dan sulit ditangkap oleh metode tradisional. Pendekatan ini menjadi solusi yang lebih efektif karena kemampuannya untuk belajar secara otomatis dari data historis, memungkinkannya beradaptasi terhadap variasi serangan *phishing* yang terus berevolusi tanpa perlu pendefinisian aturan secara manual.

Beberapa penelitian sebelumnya telah memberikan kontribusi signifikan dalam pengembangan model deteksi phishing berbasis machine learning. Misalnya, A. D. Harahap [2] mengembangkan sistem pendeteksi tautan phishing berbasis web dengan memanfaatkan algoritma Random Forest, yang berhasil mencapai akurasi sebesar 96,3%. Kelemahan utama dari pendekatan ini adalah ketergantungannya pada satu jenis model, yang sering kali rentan terhadap *overfitting* dan kurang tangguh dalam menghadapi data dengan distribusi yang berubah-ubah. Sementara itu, Mahmud dan Wirawan [3] melakukan perbandingan antara beberapa algoritma, yaitu Decision Tree, Random Forest, dan KNN, dengan hasil terbaik ditunjukkan oleh Random Forest yang mencapai akurasi 83,4%. Rendahnya akurasi ini menunjukkan adanya kekurangan dalam tahap pra-pemrosesan data, di mana aspek optimasi ekstraksi fitur belum diterapkan secara mendalam.

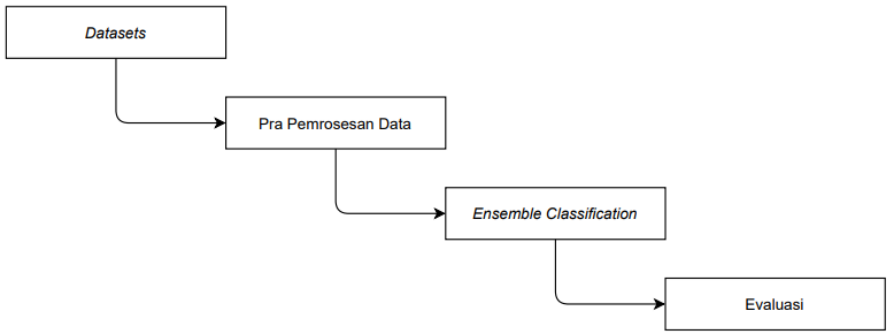
Pendekatan lain ditawarkan oleh H. A. K Afandi [4], melalui pengembangan sebuah ekstensi browser bernama GuardSurfing, yang memanfaatkan algoritma XGBoost untuk mendeteksi URL phishing. Inovasi ini tidak hanya menitikberatkan pada penerapan metode algoritmik, tetapi juga memperhatikan aspek lain yang krusial, seperti proses akuisisi data secara langsung di browser, penyediaan mekanisme untuk menjelaskan hasil keputusan sistem, serta keterlibatan pengguna melalui konsep human-in-the-loop. Hasil uji coba menunjukkan kinerja yang sangat baik, dengan capaian akurasi 97,39%, presisi 97,64%, recall 96,47%, serta nilai F1-score sebesar 97,04% pada dataset yang terdiri atas URL phishing dan non-phishing. Jika dibandingkan dengan metode lain seperti Logistic Regression, SVM, dan Random Forest, algoritma XGBoost terbukti mampu memberikan keseimbangan yang optimal antara sensitivitas dan stabilitas. Temuan tersebut menegaskan bahwa pendekatan berbasis ensemble boosting yang diterapkan langsung pada sistem nyata lebih efektif dalam menangani permasalahan ketidakseimbangan data sekaligus lebih adaptif dalam menghadapi pola serangan

phishing yang terus berkembang. Secara teoretis, penggunaan model tunggal (meskipun sekuat XGBoost) memiliki risiko stabilitas prediksi yang lebih rendah dibandingkan metode *ensemble* yang menggabungkan berbagai karakteristik algoritma (seperti *bagging*, *boosting*, dan linear). Selain itu, mayoritas penelitian tersebut, berfokus pada hasil akurasi akhir tanpa menguraikan strategi efisiensi komputasi (waktu eksekusi) yang krusial untuk penerapan *real-time* pada dataset berskala besar.

Sebagai pembeda dari penelitian-penelitian sebelumnya, studi dalam penelitian ini menawarkan pendekatan baru berupa implementasi model ensemble yang dipadukan dengan strategi ekstraksi fitur yang dioptimalkan. Berbeda dengan penelitian [2] dan [3] yang menggunakan model tunggal, serta penelitian [4] yang belum menyentuh aspek efisiensi pemrosesan, studi ini menerapkan mekanisme *batch processing*, *parallel execution*, dan *caching* pada tahap pra-pemrosesan. Pendekatan ini bertujuan untuk tidak hanya melampaui capaian akurasi sebelumnya (di atas 97,39%) tetapi juga memastikan proses deteksi berjalan efisien. Dengan mengombinasikan kekuatan *Random Forest*, *Gradient Boosting*, *Logistic Regression*, dan *AdaBoost*, penelitian ini dirancang untuk menutupi kelemahan masing-masing algoritma, sehingga menghasilkan sistem yang memiliki stabilitas prediksi tinggi dan adaptif terhadap pola serangan dinamis. Tujuan akhirnya adalah menyediakan model deteksi phishing yang superior baik dari segi akurasi maupun efisiensi komputasi untuk lingkungan *real-time*.

2. Bahan dan Metode

Penelitian ini dilaksanakan melalui serangkaian tahapan yang tersusun secara berurutan sebagaimana ditunjukkan pada Gambar 1. Setiap tahap dirancang untuk memastikan proses pembangunan model machine learning berjalan sistematis, sehingga model yang dihasilkan memiliki kinerja optimal dan sesuai dengan tujuan penelitian.



Gambar 1. Alur Penelitian

Alur penelitian ini mencakup beberapa tahapan utama. Sebagaimana diilustrasikan pada Gambar 1, alur penelitian ini dirancang secara sistematis dalam empat tahapan utama. Proses diawali dengan akuisisi dataset berskala besar dari referensi publik (Kaggle) yang mencakup kategori *malicious* dan *benign* untuk memastikan keseimbangan data. Tahap selanjutnya adalah pra-pemrosesan data, di mana strategi ekstraksi fitur teroptimasi diterapkan melalui mekanisme *parallel execution* dan *batch processing* guna meningkatkan efisiensi komputasi serta menormalisasi data input. Data yang telah terstruktur kemudian masuk ke tahap *Ensemble Classification*, yang mengintegrasikan kekuatan algoritma *Random Forest*, *Gradient Boosting*, *Logistic Regression*, dan *AdaBoost* melalui metode *Soft Voting* untuk prediksi yang lebih stabil. Rangkaian penelitian ditutup dengan tahap evaluasi kinerja model menggunakan data uji terpisah (20%), yang diukur berdasarkan metrik komprehensif seperti Akurasi, F1-Score, dan ROC-AUC guna memvalidasi keandalan sistem deteksi yang diusulkan [5].

2.1 Datasets

Kualitas dan kuantitas data merupakan fondasi utama dalam membangun model machine learning yang andal untuk deteksi phishing. Tahap awal penelitian dimulai dengan pengumpulan dataset. Dalam penelitian ini, digunakan dataset publik yang bersumber dari platform Kaggle, berjudul "Malicious and Benign URLs". Dataset ini terdiri dari lebih dari 449.778 entri URL yang telah diklasifikasikan secara jelas ke dalam dua kategori: benign (aman) dan malicious (berbahaya). Dari keseluruhan data tersebut, 345.737 URL (76,85%) termasuk kategori *benign*, sedangkan 104.019 URL (23,15%) tergolong *malicious*. Dataset ini dipilih karena memiliki ukuran besar dan mencakup variasi pola struktur URL yang luas, mulai dari karakteristik leksikal, reputasi domain, hingga parameter keamanan SSL.

Tabel 1. Contoh data

No	URL	Label	Kategori
1	https://www.google.com	0	Benign
2	https://www.youtube.com	0	Benign
3	http://wikipedia.org	0	Benign
4	http://paypal-secure-update-account.com/login	1	Malicious
5	http://192.168.1.5/banking/confirm.php	1	Malicious

Sebagaimana terlihat pada Tabel 1, URL dengan kategori *benign* umumnya memiliki struktur domain yang pendek, jelas, dan menggunakan protokol standar. Sebaliknya, URL *malicious* sering kali menunjukkan pola mencurigakan, seperti penggunaan alamat IP secara langsung, subdomain yang terlalu panjang, atau penggunaan kata kunci merek (seperti "paypal" atau "apple") yang disisipkan pada domain palsu untuk mengelabui pengguna [6].

2.2 Pra Pemrosesan Data

Pra-pemrosesan data dalam penelitian ini difokuskan pada tahap ekstraksi fitur, yang menjadi fondasi keberhasilan model machine learning dalam mendeteksi phishing. Ekstraksi fitur tidak hanya dilakukan secara konvensional, tetapi dioptimalkan agar mampu menangani dataset besar dengan efisien serta menghasilkan representasi data yang kaya informasi.

Optimasi pertama dilakukan pada aspek arsitektur komputasi. Proses ekstraksi fitur dilaksanakan secara parallel menggunakan ThreadPoolExecutor sehingga ribuan URL dapat dianalisis secara bersamaan. Teknik ini secara signifikan mengurangi waktu pemrosesan dibanding pendekatan sekuensial. Selanjutnya, untuk menjaga stabilitas sistem, data diproses dalam bentuk batch (misalnya 10.000 URL per batch), sehingga konsumsi memori lebih terkendali dan risiko kegagalan sistem dapat diminimalisasi.

Optimasi kedua terletak pada mekanisme penanganan kesalahan (error handling) dan caching. Bila terjadi kegagalan saat ekstraksi (misalnya WHOIS lookup atau validasi SSL yang sering bermasalah), sistem secara otomatis mengisi nilai default sehingga data tetap konsisten. Selain itu, hasil ekstraksi disimpan dalam cache (feature_cache.pkl), memungkinkan penggunaan ulang tanpa perlu melakukan ekstraksi ulang yang memakan waktu.

Optimasi ketiga adalah pada validasi dan normalisasi data. Setiap fitur dipastikan bernilai numerik melalui konversi eksplisit ke tipe float. Nilai kosong, NaN, maupun infinity diganti dengan 0.0, sehingga dataset menjadi bersih dan siap digunakan oleh algoritme pembelajaran mesin.

Hasil dari optimasi ini ditunjukkan pada analisis feature importance. Fitur has_https muncul sebagai faktor dominan dengan bobot terbesar, diikuti oleh fitur-fitur struktural seperti count_dots dan has_multiple_subdomains. Hal ini membuktikan bahwa

pendekatan multi-aspek yang diterapkan (meliputi leksikal, reputasi domain, serta keamanan SSL) berhasil diekstrak secara efisien dan konsisten, sekaligus relevan bagi proses klasifikasi selanjutnya. Rincian fitur-fitur yang digunakan dalam penelitian ini disajikan pada Tabel 2.

Tabel 2. Rincian Fitur yang Diekstraksi

Kategori Fitur	Fitur yang Diekstrak	Deskripsi
Karakter Leksikal URL	url_length, count_dots, count_slashes, count_at_symbol, count_question_mark, count_hyphens_in_domain, count_suspicious_chars, count_percentage_symbol	Fitur-fitur yang mengukur panjang total URL dan jumlah karakter spesifik yang sering digunakan dalam penyamaran *phishing* (seperti . atau @).
Keamanan SSL & Protokol	has_https	Mengindikasikan apakah URL menggunakan protokol HTTPS (aman) atau HTTP (tidak aman).
Struktur Path/Subdomain	has_multiple_subdomains, has_brand_name_in_path	Mengecek keberadaan banyak subdomain atau penyisipan nama merek dagang pada *path* URL
Atribut Domain	domain_length, domain_age, has_ip_address, has_suspicious_tld	Mengevaluasi usia domain, keberadaan alamat IP numerik di URL, dan apakah TLD (*Top-Level Domain*) termasuk kategori mencurigakan.

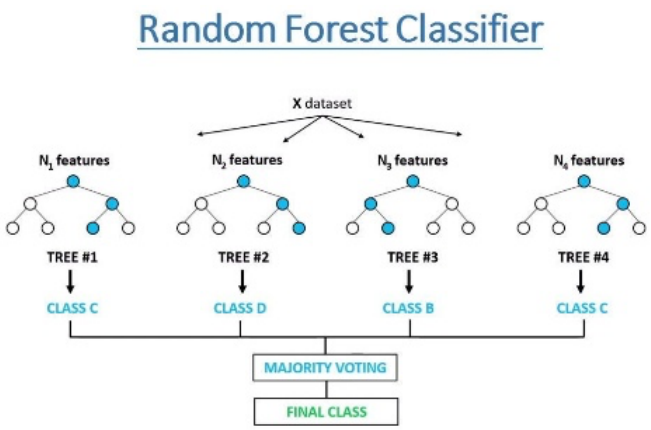
Dengan optimasi ini, tahap pra-pemrosesan data tidak hanya lebih cepat dan hemat sumber daya, tetapi juga menghasilkan representasi fitur yang kaya, valid, dan siap mendukung model ensemble untuk mencapai performa deteksi phishing yang tinggi.

2.3 Ensemble Classification

Dalam upaya meningkatkan akurasi sekaligus memperkuat ketahanan sistem terhadap serangan phishing, penelitian ini menerapkan pendekatan ensemble classification. Pendekatan ini telah banyak diinvestigasi dan terbukti efektif untuk deteksi phishing [7]. Metode ini mengombinasikan beberapa algoritma pembelajaran mesin yang berbeda sehingga kelemahan dari satu model dapat dikompensasi oleh keunggulan model lainnya[8]. Pada implementasinya, digunakan empat algoritma utama, yaitu Random Forest Classifier, Gradient Boosting Classifier, Logistic Regression, dan AdaBoost Classifier. Keempat model tersebut kemudian digabungkan melalui Voting Classifier dengan mekanisme soft voting, sehingga keputusan akhir yang dihasilkan menjadi lebih akurat dan seimbang [9].

a. Random Forest Classifier

Random Forest pertama kali diperkenalkan melalui penelitian yang menyoroti keunggulannya dalam menyelesaikan permasalahan klasifikasi maupun regresi [10]. Algoritma ini merupakan pengembangan dari *decision tree* yang dirancang untuk mengatasi kelemahan berupa tingginya risiko *overfitting* saat menggunakan *decision tree* secara tunggal. Sebagai bagian dari pendekatan *ensemble learning*, Random Forest bekerja dengan mengombinasikan sejumlah pohon keputusan yang berfungsi sebagai *classifier* dasar untuk menghasilkan prediksi yang lebih akurat terhadap suatu data.

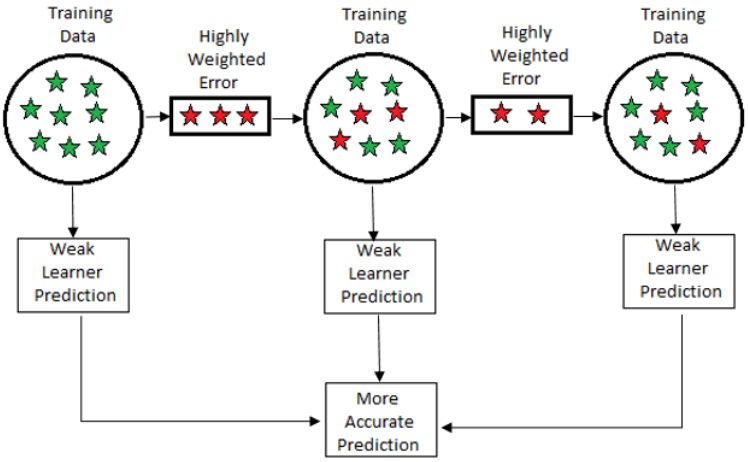


Gambar 2. *Random Forest Classifier*

Pada Gambar 2, dataset yang akan di proses menggunakan algoritma random frorest akan diambil nilai atributnya secara acak (random) untuk membangkitkan sejumlah pohon keputusan. Pada penelitian ini, Random Forest digunakan sebagai *base learner* utama untuk menangkap pola non-linear dari fitur phishing. Model dibangun dengan parameter *n_estimators=100*, yang berarti sistem membentuk seratus pohon keputusan. Kedalaman pohon dibatasi hingga *max_depth=15* untuk mencegah terjadinya *overfitting*. Selain itu, parameter *class_weight='balanced'* diterapkan guna mengatasi kemungkinan ketidakseimbangan pada dataset phishing. Proses pelatihan juga dioptimalkan dengan *n_jobs=N_JOBS*, yang memungkinkan pemanfaatan seluruh inti CPU melalui paralelisasi sehingga mempercepat waktu komputasi

b. Gardient Boosting Classifier

Gradient Boosting merupakan algoritma yang membangun model secara bertahap dengan tujuan memperbaiki kesalahan prediksi dari model sebelumnya. Setiap pohon keputusan yang ditambahkan berfungsi untuk mengurangi error yang dihasilkan pada tahap terdahulu melalui mekanisme *gradient descent[11]*. Keunggulan utama metode ini terletak pada kemampuannya memberikan performa prediksi yang tinggi, terutama pada data yang kompleks, serta efektivitasnya dalam menangani kesalahan klasifikasi [12].



Gambar 3. *Gardient Boosting Classifier*

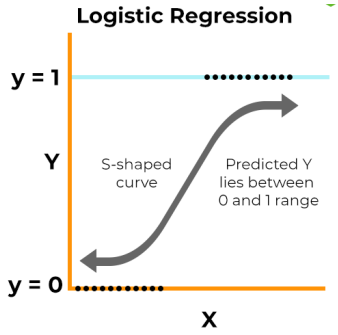
Pada Gambar 3 menjelaskan dimana algoritma *gradient boosting* bekerja dengan pendekatan iteratif, di mana setiap model lemah (*weak learner*) dibangun secara bertahap untuk memperbaiki kesalahan prediksi dari model sebelumnya [13].

Dalam penelitian ini, implementasi *Gradient Boosting* dilakukan dengan parameter n_estimators=100, yang berarti model membangun seratus pohon keputusan secara

bertahap. Setiap pohon berkontribusi dalam proses pembelajaran dengan tingkat pengaruh yang dikendalikan oleh *learning_rate*=0.1, sehingga pembaruan model berlangsung lebih stabil. Untuk mengurangi risiko *overfitting*, diterapkan *subsample*=0.8, di mana hanya 80% data digunakan pada setiap iterasi. Dengan mekanisme tersebut, *Gradient Boosting* mampu menghasilkan performa prediksi yang tinggi pada data kompleks sekaligus meningkatkan ketahanan model terhadap kesalahan klasifikasi.

c. Logistic Regression Classifier

Logistic Regression merupakan salah satu algoritma klasifikasi yang paling sederhana namun tetap efektif dalam memodelkan hubungan antara sekumpulan fitur dengan probabilitas suatu kejadian [14].



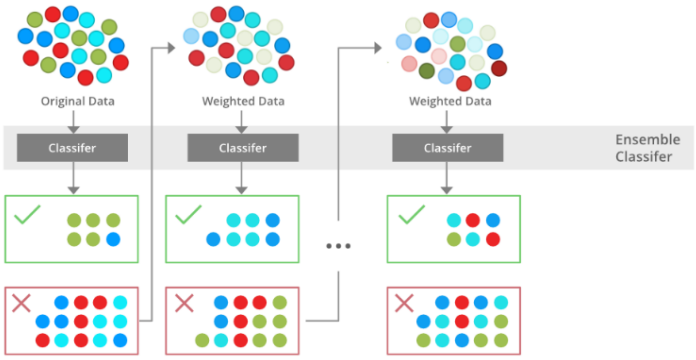
Gambar 4. *Logistic Regression Classifier*

Pada Gambar 4 bekerja dengan menggunakan fungsi sigmoid untuk memetakan nilai input menjadi probabilitas antara 0 dan 1, sehingga sangat sesuai untuk permasalahan klasifikasi biner seperti deteksi phishing. Salah satu keunggulan *Logistic Regression* adalah kemampuannya menghasilkan prediksi dalam bentuk probabilitas, bukan sekadar label biner, sehingga memungkinkan interpretasi yang lebih baik terhadap tingkat keyakinan model dalam menentukan apakah suatu URL termasuk phishing atau tidak.

Dalam penelitian ini, *Logistic Regression* dipilih sebagai salah satu algoritma utama karena memiliki kecepatan komputasi yang lebih baik dibandingkan dengan *Support Vector Machine* (SVM), terutama ketika diterapkan pada dataset berukuran besar. Untuk menjaga keseimbangan antara akurasi dan kompleksitas model, digunakan parameter C=1.0 sebagai bentuk regularisasi agar model tidak terlalu kompleks dan terhindar dari *overfitting*. Selain itu, pengaturan *class_weight*='*balanced*' diterapkan untuk mengatasi potensi ketidakseimbangan kelas dalam dataset, sehingga prediksi tidak bias terhadap kelas mayoritas (*benign*). Parameter *max_iter*=200 juga disesuaikan agar proses pelatihan memiliki cukup iterasi untuk mencapai konvergensi dengan baik.

d. AdaBoost *Classifier*

AdaBoost (*Adaptive Boosting*) merupakan algoritma *ensemble* berbasis *boosting* yang bekerja dengan menggabungkan sejumlah *weak learners*, biasanya berupa pohon keputusan sederhana, untuk membentuk sebuah model prediktif yang lebih kuat [15].



Gambar 5. AdaBoost *Classifier*

Pada Gambar 5, Prinsip utama dari AdaBoost adalah memberikan bobot lebih besar pada data yang salah diklasifikasikan oleh model sebelumnya, sehingga model pada iterasi berikutnya dapat lebih fokus memperbaiki kesalahan tersebut. Dengan mekanisme adaptif ini, AdaBoost mampu meningkatkan akurasi prediksi secara bertahap dan efektif, meskipun menggunakan model dasar yang relatif sederhana.

Pada Gambar 5, AdaBoost digunakan dengan konfigurasi *n_estimators*=50, yang berarti proses pelatihan dilakukan melalui pembangunan lima puluh *weak learners* secara bertahap. Setiap *learner* memiliki kontribusi dalam menentukan hasil prediksi akhir, dengan pembaruan bobot kesalahan dikendalikan oleh *learning_rate*=1.0. Pengaturan ini memungkinkan setiap model memperoleh pengaruh penuh untuk memperbaiki kesalahan yang muncul pada iterasi sebelumnya. Dengan demikian, proses pelatihan tidak hanya berfokus pada data yang sulit diklasifikasikan, tetapi juga tetap menjaga efisiensi komputasi.

Pendekatan tersebut diharapkan mampu meningkatkan kinerja sistem deteksi phishing, karena selain memberikan akurasi yang lebih baik, AdaBoost juga membantu menjaga keseimbangan kemampuan generalisasi model. Integrasinya ke dalam skema *ensemble* menjadikan metode ini sebagai salah satu komponen penting dalam memperkuat performa keseluruhan sistem.

Dalam penelitian ini, digunakan pendekatan *Voting Classifier* sebagai strategi *ensemble learning* untuk menggabungkan keunggulan dari beberapa model pembelajaran mesin yang telah dilatih sebelumnya, yaitu *Random Forest*, *Gradient Boosting*, *Logistic Regression*, dan AdaBoost, di mana masing-masing model memiliki kekuatan tersendiri dalam mengenali pola data. Metode ini tidak sekadar mengandalkan hasil suara terbanyak seperti pada *hard voting*, melainkan menerapkan mekanisme *soft voting* yang mengombinasikan probabilitas prediksi dari seluruh *estimator*, sehingga keputusan akhir tidak hanya menghasilkan label biner, tetapi juga menyajikan tingkat keyakinan terhadap suatu URL apakah termasuk phishing (1) atau *legitimate* (0). Sebagai ilustrasi, jika tiga model memberikan probabilitas phishing sebesar 0.8, 0.9, dan 0.7, sementara satu model lain memberikan probabilitas 0.2 untuk kategori aman, maka hasil rata-rata probabilitas tetap menunjukkan kecenderungan kuat terhadap phishing; mekanisme ini menjadikan prediksi lebih konsisten, informatif, dan stabil dibandingkan *hard voting* yang semata-mata mengandalkan jumlah suara mayoritas tanpa mempertimbangkan bobot keyakinan tiap model. Oleh karena itu, penerapan *soft voting* dalam penelitian ini memberikan keunggulan signifikan dalam meningkatkan akurasi sistem sekaligus menjaga reliabilitas deteksi phishing.

2.4 Evaluasi

Dalam penelitian ini, penilaian terhadap kinerja model dilakukan dengan memanfaatkan sejumlah metrik standar yang umum dipakai dalam ranah machine learning, terutama pada permasalahan klasifikasi biner seperti deteksi phishing. Berbagai

metrik tersebut digunakan sebagai acuan untuk menggambarkan sejauh mana model mampu melakukan prediksi secara tepat, dan masing-masing akan dijelaskan secara lebih rinci pada bagian berikutnya [16].

a. Akurasi

Akurasi merupakan metrik fundamental yang merepresentasikan persentase total prediksi yang benar (baik positif maupun negatif) terhadap keseluruhan data uji. Dalam konteks deteksi phishing, akurasi memberikan gambaran umum mengenai seberapa sering model membuat keputusan yang tepat [17]. Rumus akurasi dinyatakan sebagai:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

b. Presisi

Presisi merepresentasikan tingkat ketepatan model dalam mengidentifikasi ancaman *phishing*. Nilai presisi yang tinggi mengindikasikan rendahnya tingkat kesalahan positif (*False Positive*), yang berarti situs aman tidak salah dideteksi sebagai berbahaya [18]. Rumusnya adalah:

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

c. Recall

Recall mengukur kemampuan model dalam menemukan kembali seluruh data *phishing* yang ada dalam dataset. Metrik ini krusial untuk memastikan bahwa seminimal mungkin serangan *phishing* lolos dari deteksi [16]. Rumus recall didefinisikan sebagai:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

d. Spesifisitas

Spesifisitas menunjukkan kemampuan sistem dalam mengklasifikasikan data negatif (*benign*) dengan benar. Hal ini penting untuk menjaga kenyamanan pengguna dengan tidak memblokir situs yang sah [17]. Rumusnya adalah:

$$\text{Spesifisitas} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

e. *F1-Score*

F1-Score merupakan rata-rata harmonis antara presisi dan *recall*. Metrik ini menjadi indikator kinerja utama ketika terdapat ketidakseimbangan kelas atau ketika keseimbangan antara ketepatan dan kelengkapan deteksi sangat diutamakan [18]. Rumusnya adalah:

$$FI - Score = \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}}$$

f. ROC AUC

Kurva ROC (*Receiver Operating Characteristic*) memvisualisasikan *trade-off* antara *True Positive Rate* dan *False Positive Rate* pada berbagai ambang batas. Nilai AUC (*Area Under Curve*) digunakan untuk mengukur kemampuan diskriminatif model secara keseluruhan; semakin mendekati nilai 1, semakin baik model memisahkan kelas *malicious* dan *benign* [17].

3. Hasil

Bagian ini menyajikan hasil eksperimen yang dilakukan secara sistematis mengikuti tahapan metodologi yang telah dirancang. Pembahasan dimulai dari hasil akuisisi dan

pembagian data, kinerja pra-pemrosesan yang dioptimalkan, implementasi model *ensemble*, hingga evaluasi kinerja akhir menggunakan metrik standar.

3.1. Pembagian Data

Bagian ini menyajikan hasil eksperimen yang dilakukan untuk mengevaluasi kinerja model *machine learning* dalam mendeteksi situs phishing. Untuk menjaga objektivitas serta validitas temuan, dataset yang terdiri atas 449.778 URL dibagi dengan rasio 80:20, di mana 80% data digunakan sebagai pelatihan, sementara 20% sisanya atau sebanyak 89.956 URL dialokasikan sebagai *test* set yang tidak pernah dilihat oleh model sebelumnya. Evaluasi kinerja sepenuhnya didasarkan pada hasil pengujian terhadap *test* set tersebut. Fokus utama penyajian hasil ini adalah menampilkan efektivitas model *ensemble* yang diusulkan, yang dibangun menggunakan pendekatan ekstraksi fitur yang dioptimalkan.

3.2. Hasil Pra Pemrosesan Data

Pada tahap ini, strategi optimasi yang diterapkan melalui mekanisme *batch processing*, *parallel execution*, dan *caching* terbukti memberikan dampak signifikan terhadap efisiensi komputasi. Proses ekstraksi fitur, yang mencakup analisis struktur URL, reputasi domain, dan keamanan SSL, berhasil dijalankan dengan waktu eksekusi total sebesar 104,06 detik untuk memproses ribuan URL.

```
--- Analisis Kepentingan Fitur (Feature Importances) ---
                    Fitur  Tingkat Kepentingan
7               has_https             0.779390
1              count_dots             0.100555
11  has_multiple_subdomains            0.032239
13   has_brand_name_in_path            0.024706
9             domain_length            0.019555
0                url_length            0.014680
5             count_slashes            0.011195
6           has_ip_address             0.004796
8   count_hyphens_in_domain            0.003894
3      count_question_mark             0.003846
2           count_at_symbol            0.002188
12       has_suspicious_tld            0.001675
10   count_suspicious_chars            0.000887
4   count_percentage_symbol            0.000371
14               domain_age            0.000024

Analisis: Fitur di atas menunjukkan faktor apa yang paling dipertimbangkan model.
ains, has_brand_name_in_path, domain_length

Menyimpan model dan scaler...
Model disimpan ke: phishing_model.pkl
Scaler disimpan ke: scaler_model.pkl

Total waktu eksekusi: 104.06 detik

Proses pelatihan selesai.
```

Gambar 6. Hasil *Feature Importances*

Sebagaimana ditunjukan pada Gambar 6 hasil analisis bahwa fitur has_https mendominasi secara absolut dengan kontribusi hampir 78% terhadap kekuatan prediktif model. Temuan ini mengindikasikan bahwa keberadaan protokol HTTPS dipandang sebagai indikator paling kuat dalam menentukan legitimasi sebuah situs, sebagaimana dipelajari dari dataset yang digunakan. Selain itu, terdapat sejumlah fitur lain yang juga memberikan pengaruh signifikan, khususnya karakteristik yang berkaitan dengan teknik penyamaran URL. Beberapa di antaranya adalah *count_dots* (jumlah titik pada URL), *has_multiple_subdomains* (keberadaan banyak subdomain), serta *has_brand_name_in_path* (penyematan nama merek dalam jalur URL). Ketiga fitur ini secara kolektif berfungsi sebagai faktor sekunder yang penting dalam proses klasifikasi. Temuan menarik lainnya adalah rendahnya tingkat kepentingan fitur domain_age, yang nilainya hampir mendekati nol (0.000024). Kondisi ini menunjukkan bahwa umur domain tidak berperan signifikan sebagai pembeda dalam dataset ini. Salah satu kemungkinan penyebabnya adalah semakin banyaknya situs legitimasi baru yang bermunculan, serta kecenderungan pelaku

phishing untuk menggunakan domain yang sudah berumur (*aged domain*) guna menghindari deteksi, sehingga menurunkan nilai prediktif dari fitur tersebut.

Selain itu, proses pelatihan model berlangsung cukup efisien dengan waktu eksekusi total sekitar 104,06 detik untuk ribuan URL, berkat penerapan optimasi melalui mekanisme batch processing dan caching. Hasil ini menunjukkan bahwa tahap pra-pemrosesan tidak hanya mampu menghasilkan fitur-fitur yang relevan bagi klasifikasi, tetapi juga berhasil menjaga efisiensi komputasi. Dengan demikian, model dapat dilatih dalam waktu relatif singkat tanpa harus mengorbankan akurasi maupun kualitas prediksi yang dihasilkan.

3.3. Hasil Implementasi Model Ensemble Classification

Setelah data diproses, tahap pemodelan dilakukan dengan membangun arsitektur ensemble yang menggabungkan empat algoritma: Random Forest, Gradient Boosting, Logistic Regression, dan AdaBoost. Penerapan mekanisme Soft Voting Classifier pada tahap ini berhasil mengintegrasikan probabilitas prediksi dari keempat model tersebut.

Hasil pengujian menunjukkan bahwa pendekatan *ensemble* ini mampu menciptakan stabilitas prediksi yang lebih baik dibandingkan penggunaan model tunggal. Mekanisme *soft voting* memungkinkan sistem untuk tidak hanya menghasilkan label biner (0 atau 1), tetapi juga memberikan tingkat keyakinan (*confidence level*) yang lebih akurat dalam menentukan apakah sebuah URL berbahaya atau aman.

Tabel 3. Perbandinga Model

Model Referensi	Akurasi	Presisi
Random Forest	96,3%	96,3%
Decision Tree	83,3%	86%
XGBoost	97,39%	97,62%
RF+GB+LR+AdaBoost	98,10%	97,81%

Tabel 3 menunjukkan bahwa model ensemble yang diusulkan pada penelitian ini mampu melampaui kinerja penelitian sebelumnya [19]. Jika Random Forest dan Decision Tree hanya mencapai akurasi 83–96% dengan presisi di bawah 88%, model kombinasi *Random Forest*, *Gardient Boosting*, *Logistic Regression*, dan AdaBoost berhasil mencapai akurasi 98,10% serta presisi 97,81%, sehingga lebih efektif dalam deteksi phishing.

3.4. Hasil Evaluasi Kinerja Model

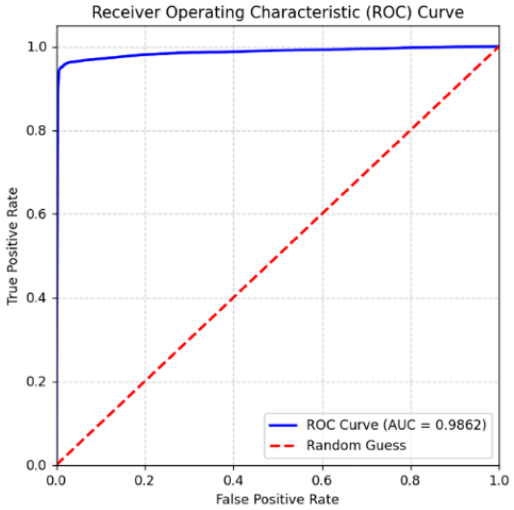
Tahap akhir adalah evaluasi kinerja model terhadap 89.956 data uji. Berdasarkan pengujian yang dilakukan, model usulan mencatatkan performa yang sangat baik di seluruh metrik evaluasi. Rincian hasil evaluasi disajikan pada Tabel 3 berikut:

Tabel 3. Hasil Evaluasi Matriks

No	Matriks	Nilai
1	Akurasi	0.9810
2	Presisi	0.9781
3	Recall	0.9390
4	F1-Score	0.9582
5	Spesifisitas	0.9937
6	ROC AUC	0.9862

Berdasarkan hasil yang tersaji pada Tabel 3, terlihat bahwa model *ensemble* yang diusulkan mampu mencapai tingkat akurasi hingga 98,10%. Angka ini menunjukkan ketepatan klasifikasi yang sangat tinggi terhadap data uji. Nilai precision sebesar 97,81% memperlihatkan bahwa hampir seluruh URL yang diklasifikasikan sebagai phishing

benar-benar termasuk kategori phishing, sehingga potensi terjadinya false positive dapat diminimalkan. Di sisi lain, capaian recall sebesar 93,90% menandakan bahwa mayoritas URL phishing berhasil teridentifikasi, meskipun masih ada sebagian kecil yang tidak terdeteksi.Selanjutnya, nilai spesifisitas sebesar 99,37% memperkuat bukti bahwa model memiliki kemampuan sangat baik dalam mengenali URL yang aman. Sementara itu, skor F1 sebesar 95,82% mencerminkan keseimbangan optimal antara precision dan recall, sehingga performa model tidak hanya akurat tetapi juga konsisten.



Gambar 7. Hasil ROC

Gambar 7. Ini adalah gambar nilai ROC AUC yang mendekati 1,0 mengindikasikan kapasitas diskriminasi yang sangat tinggi dalam membedakan URL phishing dari URL benign. Temuan ini membuktikan bahwa penerapan Teknik ekstraksi fitur yang dioptimalkan, dipadukan dengan strategi ensemble classification, menghasilkan sistem deteksi phishing yang tidak hanya efektif, tetapi juga efisien dan andal dalam konteks keamanan siber. Selanjutnya, penelitian berlanjut pada tahap ekstraksi fitur untuk mengidentifikasi faktor-faktor utama yang memengaruhi proses pengambilan keputusan model. Analisis dilakukan melalui pengukuran *feature importance* guna memahami sejauh mana masing-masing fitur berkontribusi terhadap kinerja klasifikasi. Hasil analisis ini kemudian diurutkan berdasarkan tingkat pengaruhnya, mulai dari fitur yang paling dominan hingga yang kurang signifikan.

4. Pembahasan

Hasil penelitian ini menunjukkan bahwa model ensemble yang dikembangkan berhasil mencapai performa deteksi phishing yang sangat unggul, dengan tingkat akurasi mencapai 98,10% dan nilai ROC-AUC sebesar 98,62%. Temuan tersebut memperkuat hipotesis bahwa penggabungan empat algoritma *Random Forest*, *Gradient Boosting*, *Logistic Regression*, dan AdaBoost melalui mekanisme soft voting yang didukung oleh proses ekstraksi fitur teroptimasi mampu menghasilkan sistem deteksi yang lebih akurat, stabil, dan adaptif. Keunggulan utama dari model ini tercermin pada nilai presisi yang tinggi, yaitu 97,81%, serta spesifisitas mencapai 99,37%, yang menandakan tingkat kesalahan deteksi positif palsu (*false positive*) sangat rendah. Kondisi ini penting karena memastikan situs aman tidak salah terklasifikasi sebagai phishing, sehingga meningkatkan kepercayaan pengguna terhadap sistem keamanan yang digunakan. Pencapaian ini bahkan melampaui beberapa penelitian sebelumnya yang hanya mengandalkan satu algoritma, seperti *Random Forest* [2] atau XGBoost [4]. Selain itu, waktu pra-pemrosesan yang efisien, yaitu sekitar 104,06 detik, menunjukkan bahwa model ini layak diterapkan pada sistem dunia nyata.

Dalam penerapan dunia nyata, model deteksi phishing berpotensi menghadapi berbagai threat model, seperti serangan *zero-day phishing* yang menggunakan domain sah atau melakukan *URL redirection* untuk menghindari deteksi berbasis pola. Jenis serangan ini sering kali meniru halaman resmi menggunakan konten dan elemen visual serupa, sehingga mempersulit sistem berbasis pembelajaran mesin dalam mengenalinya. Dalam sistem keamanan siber, *false negative* dapat berdampak lebih serius dibandingkan *false positive* karena memungkinkan pengguna tetap mengakses situs berbahaya tanpa peringatan. Berdasarkan hasil pengujian, meskipun nilai *recall* model mencapai 93,90%, masih terdapat kemungkinan sekitar 6,1% situs phishing lolos dari deteksi. Jika diterapkan di lingkungan nyata, hal ini dapat dimanfaatkan oleh pelaku untuk mengeksploitasi celah kecil yang tersisa.

Untuk mengatasi risiko tersebut, penelitian lanjutan perlu mengembangkan pendekatan adaptif berbasis *real-time learning,* di mana model dapat terus diperbarui menggunakan data phishing terbaru dari *threat intelligence feeds*. Selain itu, integrasi dengan sistem keamanan multi-layer seperti *DNS filtering* dan *email gateway protection* dapat membantu menekan dampak *false negative* dan memperkuat ketahanan sistem terhadap ancaman dunia nyata.

Analisis terhadap feature importance mengungkapkan bahwa fitur *has_https* (keberadaan sertifikat SSL) memiliki kontribusi paling dominan, dengan bobot hampir mencapai 78%, sedangkan fitur *domain_age* (umur domain) menunjukkan pengaruh yang sangat rendah. Fenomena ini diduga disebabkan oleh strategi baru pelaku phishing yang kini memanfaatkan domain berumur untuk mengelabui sistem deteksi otomatis. Meski hasil yang diperoleh sangat menjanjikan, ketergantungan model terhadap fitur *has_https* perlu mendapat perhatian khusus dalam penelitian lanjutan. Hal ini karena sertifikat SSL kini dapat diperoleh dengan mudah, bahkan oleh pihak berbahaya. Oleh karena itu, penelitian ke depan disarankan untuk mengembangkan fitur yang lebih kuat dan beragam, sekaligus meningkatkan nilai recall (93,90%) guna menekan tingkat kesalahan negatif palsu serta menguji ketahanan model terhadap serangan phishing *zero-day* yang semakin dinamis.

5. Kesimpulan

Penelitian ini merancang model machine learning berbasis ensemble classification dengan ekstraksi fitur yang dioptimalkan untuk mendeteksi phishing. Hasil evaluasi pada 89.956 URL menunjukkan kinerja yang sangat tinggi, dengan akurasi 98,10%, precision 97,81%, recall 93,90%, specificity 99,37%, F1-score 95,82%, dan ROC AUC 0,9862. Temuan ini menegaskan bahwa kombinasi ekstraksi fitur yang tepat dan strategi ensemble mampu menghasilkan sistem deteksi phishing yang akurat, stabil, dan andal.

Kontribusi praktis penelitian ini terletak pada potensi integrasi sistem ke berbagai platform, seperti email gateway organisasi atau ekstensi peramban, yang dapat memberikan peringatan real-time kepada pengguna. Dengan demikian, selain memberi sumbangan teoritis, model ini juga membuka peluang implementasi nyata dalam memperkuat pertahanan siber dan menghadapi pola serangan phishing yang terus berkembang.

Referensi

[1] L. A. Febrika Ardy, I. Istiqomah, A. E. Ezer, And S. N. Neyman, "Phishing Di Era Media Sosial: Identifikasi Dan Pencegahan Ancaman Di Platform Sosial," *Journal Of Internet And Software Engineering*, Vol. 1, No. 4, pp. 1-11, Jun. 2024, https://doi.org/10.47134/Pjise.V1i4.2753.

[2] A. D. Harahap, D. Juardi, And A. S. Y. Irawan, "Rancang Bangun Sistem Pendeteksi Link Phishing Menggunakan Algoritma Random Forest Berbasis Web," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 12, No. 3, Aug. 2024, https://doi.org/10.23960/jitet.v12i3.4858.

[3] A. F. Mahmud And S. Wirawan, "Deteksi Phishing Website Menggunakan Machine Learning Metode Klasifikasi Phishing Website Detection Using Machine Learning Classification Method," *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no.4, 2024. https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/3456

[4] H. A. K. Afandi, M. Lazaro Fa. Al-Dzaki, N. Qomariasih, And R. A. Wildana, "Guardsurfing : Ekstensi Browser Sebagai Alat Bantu Deteksi Website Phishing Dengan Metode Klasifikasi Xgboost Untuk Deteksi Url Phishing Berbasis Flask Framework," *Info Kripto*, Vol. 19, No. 2, pp. 73–85, Sep. 2025, https://doi.org/10.56706/ik.v19i2.124.

[5] A. F. Nugraha, R. Faticha, A. Aziza, And Y. Pristyanto, "Penerapan Metode Stacking Dan Random Forest Untuk Meningkatkan Kinerja Klasifikasi Pada Proses Deteksi Web Phishing," *Jurnal Infomedia : Teknik Informatika, Multimedia, dan Jaringan*, Vol. 7, No. 1, 2022, https://doi.org/10.30811/jim.v7i1.2959.

[6] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, And S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based On Url," *Ieee Access*, Vol. 11, pp. 36805–36822, 2023, https://doi.org/10.1109/access.2023.3252366.

[7] Y. A. Alsariera, M. H. Alanazi, Y. Said, And F. Allan, "An Investigation Of Ai-Based Ensemble Methods For The Detection Of Phishing Attacks," *Engineering, Technology And Applied Science Research*, Vol. 14, No. 3, pp. 14266–14274, Jun. 2024, https://doi.org/10.48084/Etasr.7267.

[8] F. C. Dalgic, A. S. Bozkir, And M. Aydos, "Phish-Iris: A New Approach For Vision Based Brand Prediction Of Phishing Web Pages Via Compact Visual Descriptors," In *Ismsit 2018 - 2nd International Symposium On Multidisciplinary Studies And Innovative Technologies, Proceedings*, Institute Of Electrical And Electronics Engineers Inc., Dec. 2018. https://doi.org/10.1109/Ismsit.2018.8567299.

[9] F. P. Saputra And O. Suria, "Penggabungan Model Svm Dan Naive Bayes Dengan Pendekatan Soft Voting Untuk Analisis Sentimen Tong Tji Tea House," *Sistemasi: Jurnal Sistem Informasi* , vo. 14, no.5, 2025. https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/5481.

[10] R. Saputra And E. Hartati, "Deteksi Website Phising Menggunakan Algoritma Random Forest Dengan Optimalisasi Gridsearch," vol. 14, no. 6, 2025. https://jurnal.univbinainsan.ac.id/index.php/jutim/article/view/2674.

[11] R. Allauddin Mulla, S. Saini, P. Suresh Mane, B. W. Balkhande, M. Eknath Pawar, And K. Arjun Deshmukh, "A Novel Hybrid Approach For Stock Market Index Forecasting Using Cnn-Lstm Fusion Model," *International Journal Of Intelligent Systems And Applications In Engineering*, 2024. https://ijisae.org/index.php/ijisae/article/view/4513.

[12] J. D. Duarte *Et Al.*, "Machine Learning For Early Detection Of Phishing Urls In Parked Domains: An Approach Applied To A Financial Institution," *Ieee Access*, 2025, https://doi.org/10.1109/Access.2025.3599454.

[13] D. Komalasari, T. B. Kurniawan, D. A. Dewi, M. Z. Zakaria, Z. Abdullah, And A. Alanda, "Phishing Domain Detection Using Machine Learning Algorithms," *Int J Adv Sci Eng Inf Technol*, Vol. 15, No. 1, Pp. 318–327, Feb. 2025, https://doi.org10.18517/ijaseit.15.1.12553.

[14] Z. Z. Hulaifah Al Abrori And E. R. Subhiyakto, "Analisis Komparatif Akurasi Prediksi Kanker Payudara Menggunakan Algoritma Random Forest Dan Logistic Regression," *Jurnal Algoritma*, Vol. 22, No. 1, Pp. 300–311, May 2025, https://doi.org/10.33364/algoritma/v.22-1.2164.

[15] R. Nurcahyo, F. Tanjung, And S. Rahman, "Meningkatkan Deteksi Email Phising Melalui Pendekatan SVM Yang Dioptimalkan NLP," 2025. https://repositori.uma.ac.id/jspui/handle/123456789/27660.

[16] R. Fauzan, A. V. Vitianingsih, D. Cahyono, A. L. Maukar, And Y. A. B. Suprio, "Penerapan Algoritma Klasifikasi Pada Machine Learning Untuk Deteksi Phishing," *Malcom: Indonesian Journal Of Machine Learning And Computer Science*, Vol. 5, No. 2, Pp. 531–540, Mar. 2025, https://doi.org/10.57152/malcom.v5i2.1968.

[18] Y. A. Alsariera, M. H. Alanazi, Y. Said, And F. Allan, "An Investigation Of Ai-Based Ensemble Methods For The Detection Of Phishing Attacks," *Engineering, Technology And Applied Science Research*, Vol. 14, No. 3, Pp. 14266–14274, Jun. 2024, https://doi.org/10.48084/etasr.7267.

[19] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, And A. Setiawan, "Deteksi Website Phishing Menggunakan Teknik Filter Pada Model Machine Learning," *Information System Journal*, Vol. 6, No. 01, Aug. 2023, https://Doi.Org/10.24076/Infosjournal.2023v6i01.1268.