



Comparison of Web Phishing Detection with Features Using Support Vector Machine, Random Forest, and Gradient Boosting Classifier

Linda Khoirul Inayah¹, Bambang Soegiantoro²
Jurusan Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta

¹linda.inayah@uin-suka.ac.id, ²bambang.sugiantoro@uin-suka.ac.id

ABSTRACT

Along with the rapid growth of internet usage, cybercrime incidents have increased significantly, with phishing remaining one of the most prevalent threats. This study aims to compare the performance of three machine learning algorithms, namely Support Vector Machine (SVM), Random Forest, and Gradient Boosting Classifier, in detecting phishing websites based on URL features. The dataset consists of 11,055 URL records with 31 features and was evaluated using two experimental scenarios: the Holdout Method with an 80:20 training-testing split and 10-Fold Cross Validation. The performance of each algorithm was assessed using accuracy, precision, recall, and F1-score metrics. The experimental results show that Random Forest algorithm consistently achieved the highest performance across both evaluation scenarios. Furthermore, feature importance analysis identified the HTTPS feature as the most influential factor in phishing website classification, followed by AnchorURL, LinksInScriptTags, and WebsiteTraffic. The findings indicate that the best-performing model provides stable and reliable performance for phishing website detection, while the use of 10-Fold Cross Validation demonstrates good model generalization and evaluation consistency.

Keywords: phishing detection, Support Vector Machine, Random Forest, Gradient Boosting Classifier, machine learning, 10-Fold Cross Validation

ABSTRAK

Seiring meningkatnya pemanfaatan internet, kasus kejahatan siber juga terus bertambah, salah satunya adalah *phishing*. Penelitian ini bertujuan membandingkan performa tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier*, dalam mendeteksi website *phishing* berdasarkan fitur URL. Dataset yang digunakan terdiri atas 11.055 URL dengan 31 fitur, kemudian dievaluasi menggunakan dua skenario pengujian, yaitu *Holdout Method* dengan rasio pembagian data 80:20 dan *10-Fold Cross Validation*. Kinerja setiap algoritma dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil pengujian menunjukkan bahwa algoritma Random Forest memperoleh performa terbaik pada kedua skenario pengujian. Selain itu, analisis *feature importance* menunjukkan bahwa fitur HTTPS merupakan fitur yang paling berpengaruh dalam proses klasifikasi website *phishing*. Hasil penelitian menunjukkan bahwa penggunaan metode evaluasi *Holdout* dan *10-Fold Cross Validation* menghasilkan performa yang konsisten sehingga model memiliki kemampuan generalisasi yang baik dalam mendeteksi website *phishing*.

Kata kunci: deteksi *phishing*, *Support Vector Machine*, *Random Forest*, *Gradient Boosting Classifier*, *machine learning*, *10-Fold Cross Validation*

1. PENDAHULUAN

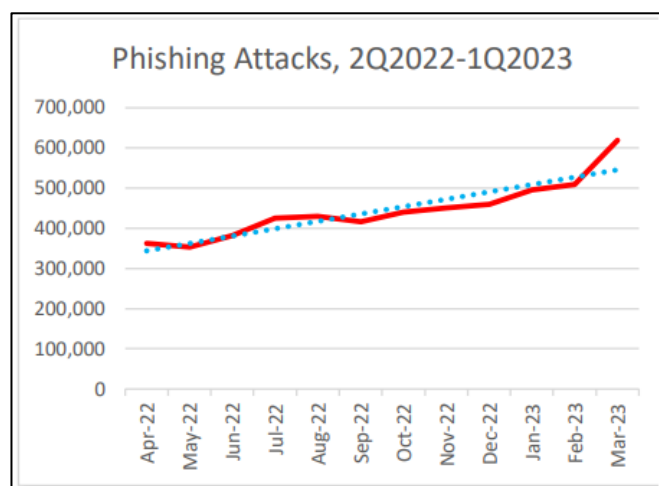
Di era digital, penggunaan internet dan teknologi informasi terus meningkat dalam berbagai aktivitas masyarakat, seperti komunikasi, transaksi, dan akses informasi. Perkembangan teknologi tersebut memberikan banyak kemudahan bagi manusia, namun juga menimbulkan berbagai ancaman dalam bidang *cyber security* [1]. Perkembangan teknologi tersebut memberikan banyak kemudahan bagi manusia, namun juga menimbulkan berbagai ancaman dalam bidang *cyber security* [2].

Menurut *International Organization for Standardization* (ISO), ISO/IEC 27032 mendefinisikan *cyber security* adalah preservasi dari kerahasiaan, integritas, dan ketersediaan informasi di *cyberspace* [2]. Sementara menurut

Kaspersky, mendefinisikan *cyber security* sebagai suatu praktik melindungi komputer, server, perangkat *mobile*, sistem elektronik, jaringan, dan data dari serangan - serangan siber [3]. Oleh karena itu, *cyber security* merupakan tindakan yang dilakukan untuk melindungi informasi di dunia maya dari berbagai ancaman [4]. *Cyber security* menjadi semakin penting karena meningkatnya penggunaan komputer seperti desktop, laptop, smartphone, server, dan perangkat IoT (Internet of Things), serta meluasnya penggunaan jaringan komputer seperti internet dalam kehidupan sehari-hari [5]. Salah satu hal yang paling umum dan berbahaya dari *cyber crime* adalah serangan *phishing* [6].

Phishing merupakan teknik penipuan yang dilakukan dengan menyamar sebagai pihak terpercaya untuk memperoleh informasi sensitif pengguna, seperti kata sandi, data pribadi, dan informasi keuangan [7]. *Phishing* adalah sejenis pencurian identitas elektronik dimana kombinasi rekayasa sosial dan metode pembuatan *website* palsu digunakan untuk menipu pengguna agar mengungkapkan rincian rahasia dan berharga miliknya [8]. Istilah '*phishing*' berasal dari kata '*fishing*', karena melibatkan memikat korban untuk mengambil umpan [9].

Phishing adalah salah satu masalah besar yang dihadapi dunia *cyber* dan menyebabkan kerugian finansial baik bagi industri maupun individu [10]. Serangan *phishing* dapat mengakibatkan kerugian finansial, pencurian identitas, dan pelanggaran data yang dapat merugikan pengguna. Berdasarkan laporan survei yang dilakukan oleh Anti-Phishing Working Group (APWG) pada bulan April 2022 hingga Maret 2023, seperti terlihat pada gambar berikut [11].



Gambar 1. Statistik Phishing (Sumber : APWG)

Dari Gambar 2 di atas terlihat bahwa tindakan *phishing* mengalami kenaikan setiap bulannya mulai dari April 2022 sampai dengan Maret 2023. APWG juga mencatat bahwa tiga bulan pertama pada tahun 2023 merupakan periode dengan jumlah serangan *phishing* tertinggi yang pernah diamati. APWG juga melakukan penelitian darimana saja tindakan *phishing* terjadi. Data pada tabel berikut merupakan penelitian yang dilakukan oleh APWG pada bulan Januari 2023 sampai dengan Maret 2023 [11].

Tabel 1. Sumber Tindakan Phishing

Category	January	February	March
Number of unique phishing Web sites (attacks) detected	495,690	509,394	619,060
Unique phishing email campaigns	40,863	45,259	40,742
Number of brands targeted by phishing campaigns	561	549	576

Berdasarkan Tabel 1, *web phishing* merupakan sumber tindakan *phishing* dengan jumlah tertinggi dibandingkan *email phishing* dan *brand phishing*. Selain itu, jumlah *web phishing* juga mengalami peningkatan setiap bulannya. Kondisi ini menunjukkan bahwa deteksi *website phishing* menjadi sangat penting untuk meningkatkan keamanan pengguna internet. Umumnya informasi yang dicari *phisher* adalah berupa *username*, *password*, baik itu akun media sosial atau akun nomor kartu kredit dengan cara diarahkan ke sebuah situs *website* palsu [12].

Dalam konteks deteksi web *phishing*, ketergantungan manusia pada teknologi dan internet menunjukkan pentingnya pengembangan metode deteksi *phishing* yang efektif dan dapat diandalkan. Dikarenakan manusia semakin bergantung pada teknologi untuk berkomunikasi, bertransaksi, dan mengakses informasi, risiko menjadi korban serangan *phishing* juga semakin meningkat.

Untuk mengklasifikasikan URL *website* yang merupakan *phishing* atau bukan *phishing* dapat digunakan berbagai metode *machine learning* [12]. Beberapa algoritma klasifikasi yang sering digunakan antara lain *Support Vector Machine* (SVM), *Random Forest*, *Gradient Boosting Classifier*, *Naïve Bayes Classifier* (NBC), dan *Classification and Regression Tree* (CART) [13] [14]. Algoritma-algoritma tersebut memiliki kemampuan yang berbeda dalam melakukan klasifikasi data sehingga diperlukan analisis untuk mengetahui performa terbaik dalam mendeteksi *phishing* berbasis URL.

Penelitian yang dilakukan oleh [15] dengan judul Implementasi Metode *Random Forest* Klasifikasi untuk *Phishing Link Detection* didapatkan hasil akurasi sebesar 94,36% dengan menggunakan 2.457 dataset. Kemudian penelitian yang dilakukan oleh [16] dengan judul *Comparison of SVM and Gradient Boosting with PCA for Website Phishing Detection* menunjukkan bahwa algoritma *Support Vector Machine* (SVM) memiliki performa yang lebih unggul dalam mendeteksi *phishing* dibandingkan *Gradient Boosting Classifier*, terutama pada nilai akurasi dan *recall*. Namun, *Gradient Boosting Classifier* tetap memberikan performa yang stabil dan seimbang pada nilai *precision* dan *recall*. Penelitian sebelumnya telah menerapkan berbagai algoritma *machine learning* dalam deteksi *phishing* dan menunjukkan hasil performa yang cukup baik. Namun, sebagian besar penelitian hanya berfokus pada penggunaan satu atau dua algoritma klasifikasi sehingga belum diketahui algoritma yang paling optimal dalam mendeteksi web *phishing* menggunakan dataset dan fitur URL yang sama secara komprehensif. Selain itu, perbedaan karakteristik data dapat mempengaruhi tingkat akurasi masing-masing algoritma sehingga diperlukan analisis lebih lanjut untuk menentukan metode yang paling efektif dalam mendeteksi *phishing* berbasis URL.

Oleh karena itu, penelitian ini bertujuan untuk membandingkan performa algoritma *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier* dalam mendeteksi web *phishing* berdasarkan fitur URL. Hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi *phishing* yang lebih efektif dan membantu meningkatkan keamanan pengguna internet terhadap ancaman *phishing*.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Tahapan penelitian dilakukan secara sistematis mulai dari pengumpulan data hingga evaluasi model. Proses penelitian terdiri atas beberapa tahap sebagai berikut.

1. Pengumpulan Dataset
Dataset *phishing* diperoleh dari Kaggle dalam format CSV yang berisi 11.055 data URL beserta 30 fitur dan satu label kelas.
2. Pra-pemrosesan Data
Dataset diperiksa untuk memastikan tidak terdapat missing value maupun *outlier*. Seluruh fitur telah berbentuk numerik sehingga tidak diperlukan proses *encoding*.
3. Pembagian Data
Pada skenario pertama, dataset dibagi menggunakan *Holdout Method* dengan rasio 80% data latih dan 20% data uji. Pada skenario kedua digunakan metode *10-Fold Cross Validation*.
4. Pelatihan Model
Algoritma yang digunakan yaitu *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier*, dilatih menggunakan data latih sesuai dengan skenario pengujian.
5. Prediksi
Model yang telah dilatih digunakan untuk melakukan klasifikasi URL ke dalam kelas *phishing* atau *non-phishing* pada data uji.
6. Evaluasi Model
Hasil klasifikasi dievaluasi menggunakan *Confusion Matrix* untuk memperoleh nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Pada skenario kedua, nilai evaluasi dihitung sebagai rata-rata dari seluruh *fold*.
7. Analisis Hasil

Nilai evaluasi dari ketiga algoritma dibandingkan untuk menentukan algoritma yang memiliki performa terbaik dalam mendeteksi website *phishing*.

2.2. Pengumpulan Data

Data yang digunakan adalah dataset yang bersumber dari Kaggle dengan format csv. Data yang digunakan berjumlah 11.054 data dengan 31 fitur. Dari 31 fitur, 30 fitur merupakan fitur independen dan 1 fitur dependen. Dari data yang digunakan tidak ada *outlier* dan *missing value* dalam data. Setiap fitur dalam data menggunakan format integer. Jadi, tidak perlu menggunakan *LabelEncoder*.

Setiap data memiliki 30 fitur dan sebuah kelas yang mendefinisikan data tersebut merupakan phishing atau bukan. Data yang merupakan link phishing dilambangkan dengan 1, sedangkan data yang bukan link *phishing* dilambangkan dengan -1 pada fitur *class*-nya. Pada setiap fitur data, ada data dengan nilai 1, 0, dan -1. Pada tabel 2, terdapat contoh dataset.

Tabel 2. Contoh Dataset

Index	UsingIP	LongURL	ShortURL	Symbols@	Redirecting//	PrefixSuffix-	Subdomains	HTTPS	DomainRegLen	Favicon	NonStdPort	HTTPSDomainURL	RequestURL	AnchorURL	LinksInScriptTags
0	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1
1	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1
2	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0
ServerFormHandler	InfoEmail	AbnormalURL	WebsiteForwarding	StatusBarCustom	DisableRightClick	UsingPopUpWindow	IframeRedirection	AgeofDomain	DNSRecording	WebsiteTraffic	PageRank	GoogleIndex	LinksPointingToPage	StatsReport	class
-1	1	1	0	1	1	1	1	-1	-1	0	-1	1	1	1	-1
-1	-1	-1	0	1	1	1	1	1	-1	1	-1	1	0	-1	-1
-1	1	1	0	1	1	1	1	-1	-1	1	-1	1	-1	1	-1

2.3. Pembagian Data

Untuk pembagian data, penelitian ini menggunakan dua pendekatan, yaitu *Holdout Method* dan *10-Fold Cross Validation*. Kedua metode tersebut digunakan sebagai skenario pengujian untuk mengevaluasi performa algoritma *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier* dalam mendeteksi website *phishing*.

Metode pertama adalah *Holdout Method*, atau yang sering disebut *train-test split*. Metode ini membagi dataset menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Data latih digunakan untuk membangun model klasifikasi, sedangkan data uji digunakan untuk mengukur kemampuan model dalam melakukan prediksi terhadap data yang belum pernah digunakan selama proses pelatihan [17].

Proporsi pembagian data yang digunakan adalah 80% sebagai data latih dan 20% sebagai data uji. Pemilihan rasio tersebut bertujuan agar model memperoleh jumlah data yang cukup besar untuk mempelajari karakteristik URL *phishing* dan *non-phishing*, sekaligus menyediakan data yang memadai untuk proses evaluasi. Metode *Holdout*

dipilih karena sederhana, efisien, dan banyak digunakan dalam penelitian klasifikasi berbasis *machine learning*, terutama pada dataset berukuran besar.

Selain *Holdout*, penelitian ini juga menerapkan *10-Fold Cross Validation* sebagai metode validasi untuk memperoleh hasil evaluasi yang lebih stabil dan objektif. Pada metode ini, dataset dibagi menjadi sepuluh bagian (*fold*) dengan ukuran yang relatif sama. Setiap iterasi menggunakan satu *fold* sebagai data uji dan sembilan *fold* lainnya sebagai data latih. Proses tersebut dilakukan sebanyak sepuluh kali sehingga setiap data menjadi data uji tepat satu kali.

Penggunaan *10-Fold Cross Validation* bertujuan untuk mengurangi bias akibat pembagian data secara acak pada metode *Holdout* serta menghasilkan estimasi performa model yang lebih representatif. Pada setiap iterasi dihitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*, kemudian seluruh hasil dirata-ratakan sebagai performa akhir masing-masing algoritma.

Berdasarkan kedua metode pembagian data tersebut, penelitian ini menyusun dua skenario pengujian yang dijelaskan pada subbab berikutnya.

2.4. Skenario Penelitian

Penelitian ini menggunakan dua skenario pengujian untuk membandingkan performa algoritma *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier*. Setiap skenario menggunakan metode evaluasi yang berbeda sehingga dapat diketahui tidak hanya tingkat akurasi, tetapi juga konsistensi performa masing-masing algoritma.

2.4.1. Skenario 1

Pada skenario pertama, dataset dibagi menggunakan *Holdout Method* dengan rasio 80% data latih dan 20% data uji. Ketiga algoritma, yaitu *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier*, dilatih menggunakan data latih dan kemudian diuji menggunakan data uji.

Hasil klasifikasi dievaluasi menggunakan *Confusion Matrix* untuk memperoleh nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Skenario ini bertujuan untuk membandingkan performa ketiga algoritma pada satu kali pembagian data.

2.4.2. Skenario 2

Pada skenario kedua, evaluasi dilakukan menggunakan *10-Fold Cross Validation*. Seluruh dataset dibagi menjadi sepuluh *fold*, kemudian setiap *fold* secara bergantian digunakan sebagai data uji, sedangkan sembilan *fold* lainnya digunakan sebagai data latih. Proses ini dilakukan sebanyak sepuluh iterasi sehingga setiap data pernah digunakan sebagai data uji.

Pada setiap iterasi dihitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Selanjutnya seluruh hasil dirata-ratakan untuk memperoleh performa akhir setiap algoritma. Skenario ini digunakan untuk mengetahui konsistensi dan kemampuan generalisasi model dibandingkan dengan hasil pengujian menggunakan metode *Holdout*.

2.5. Algoritma Machine Learning

Penelitian ini mengimplementasikan tiga algoritma machine learning untuk melakukan klasifikasi URL menjadi kategori *phishing* dan *non-phishing*. Ketiga algoritma tersebut adalah *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier*. Pemilihan ketiga metode ini didasarkan pada perbedaan karakteristik pendekatan klasifikasi, sehingga dapat dilakukan perbandingan performanya. Setiap algoritma dilatih menggunakan dataset yang sama dan diuji menggunakan data uji yang identik agar hasil evaluasi yang diperoleh dapat dibandingkan secara langsung.

2.5.1. Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma *supervised learning* yang banyak digunakan untuk klasifikasi dan regresi. Dalam penelitian ini, SVM digunakan untuk mengelompokkan URL ke dalam dua

kelas, yaitu *phishing* dan *non-phishing*. Cara kerja SVM didasarkan pada pencarian sebuah *hyperplane* yang dapat memisahkan dua kelas data dengan jarak pemisah (*margin*) yang paling optimal [18].

Hyperplane tersebut dipilih dengan mempertimbangkan jarak terjauh terhadap titik data terdekat dari masing-masing kelas yang disebut *support vector*. Jarak ini dikenal sebagai *margin*. Semakin besar margin yang terbentuk, maka semakin baik pula kemampuan model dalam melakukan generalisasi terhadap data baru. Selain itu, SVM juga dapat digunakan pada data yang tidak dapat dipisahkan secara linear dengan memanfaatkan fungsi kernel. Fungsi ini memungkinkan data dipetakan ke ruang berdimensi lebih tinggi sehingga pemisahan antar kelas dapat dilakukan dengan lebih efektif [18].

Dengan kemampuan tersebut, SVM cukup andal untuk menangani data dengan jumlah fitur yang banyak dan sering digunakan dalam penelitian di bidang keamanan informasi, termasuk untuk mendeteksi website *phishing* [18].

2.5.2. Random Forest

Random Forest merupakan algoritma *ensemble learning* yang menggunakan banyak model *decision tree* secara acak dan menggabungkan hasil prediksinya untuk menghasilkan keputusan akhir. Setiap pohon keputusan dibangun dari subset data dan subset fitur yang dipilih secara acak, sehingga menghasilkan variasi antar model [19].

Pendekatan ini bertujuan untuk mengurangi risiko *overfitting* yang sering terjadi pada model *decision tree* tunggal. Prediksi akhir pada *Random Forest* ditentukan berdasarkan hasil *voting* mayoritas dari seluruh pohon keputusan yang terbentuk. Dengan cara ini, kesalahan pada satu pohon dapat dikompensasi oleh pohon lainnya.

Random Forest dikenal memiliki performa yang stabil, mampu menangani data dengan jumlah fitur yang besar, serta relatif tahan terhadap *noise*, sehingga cocok digunakan pada dataset URL *phishing* yang memiliki banyak karakteristik fitur.

2.5.3. Gradient Boosting Classifier

Gradient Boosting Classifier (GBC) merupakan metode *ensemble* yang membangun model secara bertahap dan berurutan. Berbeda dengan *Random Forest* yang membangun pohon secara independen, GBC membangun setiap model baru berdasarkan kesalahan yang dihasilkan oleh model sebelumnya. Dengan demikian, setiap tahap pembelajaran berfokus pada peningkatan performa terhadap data yang sebelumnya sulit diklasifikasikan [20].

Model dasar yang digunakan dalam GBC umumnya berupa *decision tree* berukuran kecil yang disebut sebagai *weak learner*. Prediksi akhir diperoleh dari kombinasi seluruh model yang dibangun secara berurutan. Proses optimasi dalam GBC dilakukan dengan meminimalkan fungsi kerugian (*loss function*) menggunakan pendekatan gradien [20].

Keunggulan utama *Gradient Boosting* terletak pada tingkat akurasi yang tinggi serta kemampuannya dalam menangkap pola kompleks pada data. Oleh karena itu, algoritma ini sering digunakan dalam berbagai kompetisi data science dan aplikasi klasifikasi dengan tingkat kompleksitas tinggi, termasuk dalam deteksi *phishing* berbasis fitur URL [20].

2.6. Confusion Matrix

Evaluasi kinerja setiap algoritma dalam penelitian ini dilakukan menggunakan *confusion matrix*. *Confusion matrix* merupakan tabel evaluasi yang menggambarkan perbandingan antara label prediksi yang dihasilkan oleh model dengan label aktual pada data uji. Matriks ini terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [21].

Berdasarkan nilai-nilai tersebut, beberapa metrik evaluasi digunakan untuk menilai performa model, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* menunjukkan proporsi seluruh data yang berhasil diklasifikasikan dengan benar oleh model. *Precision* menggambarkan tingkat ketepatan prediksi positif, yaitu perbandingan antara jumlah data yang benar-benar positif dengan seluruh data yang diprediksi sebagai positif.

Recall menunjukkan kemampuan model dalam menemukan seluruh data positif yang sebenarnya, yaitu perbandingan antara jumlah data positif yang berhasil terdeteksi dengan seluruh data positif yang ada pada dataset. Sementara itu, *F1-score* merupakan nilai rata-rata harmonis antara *precision* dan *recall*, yang digunakan untuk menyeimbangkan kedua metrik tersebut, terutama ketika terjadi ketidakseimbangan jumlah kelas dalam dataset.

Penggunaan keempat metrik ini memberikan gambaran yang lebih komprehensif terhadap performa model dibandingkan hanya menggunakan *accuracy* saja [21].

3. Hasil dan Pembahasan

Penelitian ini membandingkan performa tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier* dalam mendeteksi website *phishing*. Pengujian dilakukan menggunakan dua skenario, yaitu *Holdout Method* dengan pembagian data 80:20 dan *10-Fold Cross Validation*. Evaluasi dilakukan menggunakan *confusion matrix* untuk memperoleh nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

3.1. Hasil Pengujian Skenario 1 (*Holdout 80:20*)

Pada skenario pertama, dataset dibagi menggunakan *Holdout Method* dengan komposisi 80% sebagai data latih dan 20% sebagai data uji. Ketiga algoritma dilatih menggunakan data latih dan kemudian dievaluasi menggunakan data uji. Hasil dari perbandingan ketiga algoritma dapat dilihat pada tabel berikut.

Tabel 3. Hasil Perbandingan Algoritma dengan Skenario 1

Algoritma	Accuracy	Precision	Recall	F1 Score
Support Vector Machine (SVM)	95.4%	94.9%	96.9%	95.9%
Random Forest	97.3%	96.8%	98.4%	97.6%
Gradient Boosting Classifier	95.3%	94.9%	96.8%	95.9%

Berdasarkan Tabel 3, algoritma *Gradient Boosting Classifier* memperoleh nilai *accuracy* tertinggi dibandingkan dua algoritma lainnya. Selain itu, nilai *precision*, *recall*, dan *F1-score* yang dihasilkan juga menunjukkan bahwa algoritma ini mampu mengidentifikasi website *phishing* dengan lebih baik dibandingkan *Support Vector Machine* dan *Random Forest*.

3.2. Hasil Pengujian Skenario 2 (*10-Fold Cross Validation*)

Pada skenario kedua, evaluasi dilakukan menggunakan metode *10-Fold Cross Validation*. Dataset dibagi menjadi sepuluh bagian (*fold*), kemudian setiap *fold* secara bergantian digunakan sebagai data uji sementara sembilan *fold* lainnya digunakan sebagai data latih. Nilai *accuracy*, *precision*, *recall*, dan *F1-score* dihitung pada setiap iterasi, kemudian dirata-ratakan sebagai hasil akhir.

Tabel 4. Hasil Perbandingan Algoritma dengan Skenario 2

Algoritma	Accuracy	Precision	Recall	F1 Score
Support Vector Machine (SVM)	94.9%	94.2%	96.7%	95.4%
Random Forest	97.2%	96.9%	98.1%	97.5%
Gradient Boosting Classifier	95.0%	94.8%	96.3%	95.5%

Hasil pengujian menggunakan *10-Fold Cross Validation* menunjukkan bahwa performa ketiga algoritma relatif stabil pada setiap *fold*. *Gradient Boosting Classifier* tetap memperoleh nilai rata-rata *accuracy*, *precision*, *recall*, dan *F1-score* tertinggi dibandingkan algoritma lainnya. Hal ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah digunakan selama proses pelatihan.

3.3. Perbandingan Hasil Antar Skenario

Untuk mengetahui pengaruh metode evaluasi terhadap performa model, dilakukan perbandingan hasil antara *Holdout Method* dan *10-Fold Cross Validation*.

Tabel 5. Hasil Perbandingan Algoritma

Algoritma	Holdout Accuracy	Cross Validation Accuracy
SVM	95.4%	94.9%
Random Forest	97.3%	97.2%

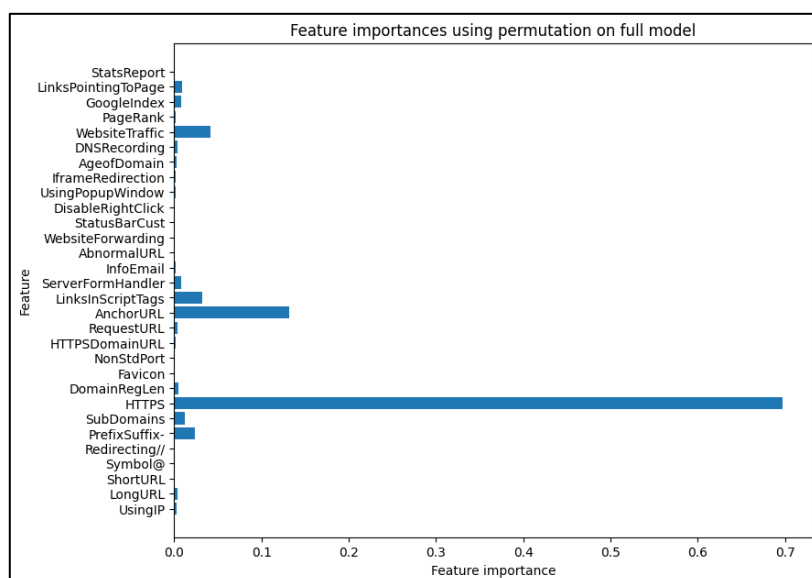
Gradient Boosting	95.3%	95.0%
--------------------------	-------	-------

Berdasarkan Tabel 5, ketiga algoritma menunjukkan hasil yang relatif konsisten pada kedua skenario pengujian. Algoritma Support Vector Machine (SVM) memperoleh nilai *accuracy* sebesar 95,4% pada metode Holdout dan 94,9% pada metode 10-Fold Cross Validation. Algoritma Random Forest memperoleh nilai *accuracy* tertinggi, yaitu 97,3% pada metode Holdout dan 97,2% pada metode 10-Fold Cross Validation. Sementara itu, Gradient Boosting Classifier memperoleh nilai *accuracy* sebesar 95,3% pada metode Holdout dan 95,0% pada metode 10-Fold Cross Validation.

Perbedaan nilai *accuracy* antara kedua skenario pengujian relatif kecil pada seluruh algoritma, dengan selisih kurang dari 1%. Hal ini menunjukkan bahwa performa model tidak dipengaruhi secara signifikan oleh metode evaluasi yang digunakan. Selain itu, hasil *10-Fold Cross Validation* memberikan validasi bahwa model memiliki kemampuan generalisasi yang baik karena menghasilkan performa yang konsisten pada setiap pembagian data. Berdasarkan kedua skenario tersebut, Random Forest menjadi algoritma dengan performa terbaik sekaligus paling stabil dalam mendeteksi website *phishing*.

3.4. Analisis Fitur Terpenting

Analisis fitur terpenting atau yang disebut *feature importance* adalah ukuran yang menunjukkan sejauh mana fitur (atau variabel) dalam dataset mempengaruhi keputusan yang dibuat oleh model pembelajaran mesin. Analisis fitur terpenting digunakan untuk membantu dalam memahami kontribusi masing-masing fitur terhadap prediksi model dan dapat digunakan untuk berbagai tujuan seperti pemilihan fitur, interpretabilitas model, dan pengurangan dimensi. Dari Gambar 2, dapat disimpulkan bahwa fitur terpenting pada penelitian ini adalah *HTTPS*, *AnchorURL*, *LinksinScriptTags*, dan *WebsiteTraffic*.



Gambar 2. Feature Importance

3.5. Analisis Performa Model

Hasil penelitian menunjukkan bahwa Random Forest merupakan algoritma dengan performa terbaik dibandingkan Support Vector Machine (SVM) dan Gradient Boosting Classifier. Pada skenario Holdout, Random Forest memperoleh nilai *accuracy* sebesar 97,3%, sedangkan pada skenario 10-Fold Cross Validation memperoleh nilai *accuracy* sebesar 97,2%. Selisih sebesar 0,1% menunjukkan bahwa algoritma ini memiliki kemampuan generalisasi yang sangat baik dan mampu mempertahankan performanya pada berbagai pembagian data. Keunggulan Random Forest dipengaruhi oleh mekanisme *ensemble learning*, yaitu membangun sejumlah *decision tree* secara acak (*bagging*) dan menggabungkan hasil prediksi seluruh pohon melalui mekanisme *majority voting*.

Pendekatan ini mampu mengurangi risiko *overfitting* sekaligus meningkatkan akurasi klasifikasi dibandingkan penggunaan satu model *decision tree*. Selain itu, proses pemilihan subset fitur secara acak pada setiap pohon membantu model menangkap pola yang beragam pada karakteristik URL phishing sehingga menghasilkan prediksi yang lebih akurat.

Sementara itu, Support Vector Machine (SVM) menghasilkan performa yang cukup baik dengan nilai *accuracy* sebesar 95,4% pada metode Holdout dan 94,9% pada 10-Fold Cross Validation. Walaupun memiliki kemampuan klasifikasi yang tinggi, performanya masih berada di bawah Random Forest karena karakteristik data URL phishing yang kompleks lebih efektif dimodelkan menggunakan pendekatan berbasis *ensemble*. Gradient Boosting Classifier juga menunjukkan performa yang baik dengan nilai *accuracy* sebesar 95,3% pada metode Holdout dan 95,0% pada 10-Fold Cross Validation. Namun, pada dataset yang digunakan, pendekatan *boosting* belum mampu melampaui performa Random Forest.

Secara keseluruhan, hasil penelitian menunjukkan bahwa Random Forest merupakan algoritma yang paling efektif untuk mendeteksi website *phishing* pada dataset yang digunakan. Konsistensi hasil pada metode Holdout dan 10-Fold Cross Validation menunjukkan bahwa model memiliki tingkat stabilitas dan kemampuan generalisasi yang baik sehingga layak digunakan sebagai pendekatan klasifikasi pada kasus deteksi website *phishing*.

4. KESIMPULAN

Penelitian ini berhasil membandingkan performa algoritma *Support Vector Machine* (SVM), *Random Forest*, dan *Gradient Boosting Classifier* dalam mendeteksi website *phishing* menggunakan dua skenario pengujian, yaitu *Holdout Method* dengan rasio pembagian data 80:20 dan *10-Fold Cross Validation*. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur kinerja masing-masing algoritma.

Hasil pengujian menunjukkan bahwa algoritma *Random Forest* memberikan performa terbaik pada kedua skenario pengujian. Pada skenario *Holdout Method*, *Random Forest* memperoleh nilai *accuracy* sebesar 97,3%, lebih tinggi dibandingkan *Support Vector Machine* sebesar 95,4% dan *Gradient Boosting Classifier* sebesar 95,3%. Sementara itu, pada skenario *10-Fold Cross Validation*, *Random Forest* tetap menghasilkan nilai *accuracy* tertinggi sebesar 97,2%, sedangkan *Support Vector Machine* dan *Gradient Boosting Classifier* masing-masing memperoleh nilai *accuracy* sebesar 94,9% dan 95,0%. Hasil tersebut menunjukkan bahwa *Random Forest* memiliki kemampuan klasifikasi dan generalisasi yang lebih baik dibandingkan kedua algoritma lainnya dalam mendeteksi website *phishing*. Perbedaan nilai *accuracy* antara metode *Holdout* dan *10-Fold Cross Validation* pada ketiga algoritma relatif kecil, sehingga menunjukkan bahwa performa model bersifat stabil pada berbagai skenario evaluasi. Selain itu, analisis *feature importance* menunjukkan bahwa fitur *HTTPS* merupakan fitur yang paling berpengaruh dalam proses klasifikasi, sehingga keberadaan protokol *HTTPS* menjadi salah satu indikator penting dalam membedakan website *phishing* dan *non-phishing*.

Penelitian selanjutnya dapat mengembangkan model dengan menggunakan dataset yang lebih beragam, melakukan optimasi *hyperparameter*, menerapkan teknik seleksi fitur, serta membandingkan algoritma *machine learning* dengan metode *deep learning* untuk meningkatkan performa deteksi website *phishing*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Dr. Ir. Bambang Sugiantoro, M.T., IPU., ASEAN Eng. selaku dosen pembimbing atas bimbingan, arahan, dukungan, dan saran yang diberikan selama proses penelitian dan penyusunan artikel ini. Ucapan terima kasih juga disampaikan kepada Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga atas dukungan dan kontribusinya terhadap keberhasilan penelitian ini.

DAFTAR RUJUKAN

- [1] “Asosiasi Penyelenggara Jasa Internet Indonesia (APJII),” APJII Official Website. [Online]. Available: <https://web.apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>

-
- [2] "ISO/IEC 27032:2023(en) Cybersecurity — Guidelines for Internet security." [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:27032:en>
 - [3] A. Leahovcenco, "Cybersecurity as a Fundamental Element of the Digital Economy," *MEST*, vol. 9, no. 1, pp. 97–105, Jan. 2021, doi: 10.12709/mest.09.09.01.13.
 - [4] K. Patel and D. Chudasama, "National Security Threats in Cyberspace," *NJCSL*, 2021.
 - [5] the Universiti Kebangsaan Malaysia, Malaysia, N. A. A. Rahman, I. H. Sairi, N. A. M. Zizi, and F. Khalid, "The Importance of Cybersecurity Education in School," *IJIET*, vol. 10, no. 5, pp. 378–382, 2020, doi: 10.18178/ijiet.2020.10.5.1393.
 - [6] S. Wajiha Zahra, S. Riaz, and A. Arshad, "Phishing Attack, Its Detections and Prevention Techniques," *IJWSN*, 2023.
 - [7] L. Ribeiro, I. S. Guedes, and C. S. Cardoso, "Which factors predict susceptibility to phishing? An empirical study," *Computers & Security*, vol. 136, p. 103558, Jan. 2024, doi: 10.1016/j.cose.2023.103558.
 - [8] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," *Expert Systems with Applications*, vol. 53, pp. 231–242, Jul. 2016, doi: 10.1016/j.eswa.2016.01.028.
 - [9] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," *Front. Comput. Sci.*, vol. 3, p. 563060, Mar. 2021, doi: 10.3389/fcomp.2021.563060.
 - [10] A. K. Jain and B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," *Security and Communication Networks*, vol. 2017, pp. 1–20, 2017, doi: 10.1155/2017/5421046.
 - [11] APWG, "Phishing Activity Trends Report," vol. 3rd Quarter 2022, 2022.
 - [12] 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, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.
 - [13] A.-A.-R. Asif *et al.*, "Performance Evaluation and Comparative Analysis of Different Machine Learning Algorithms in Predicting Cardiovascular Disease," vol. 29, no. 2, 2021.
 - [14] S. Zhang, A. Khattak, C. M. Matara, A. Hussain, and A. Farooq, "Hybrid feature selection-based machine learning Classification system for the prediction of injury severity in single and multiple-vehicle accidents," *PLoS ONE*, vol. 17, no. 2, p. e0262941, Feb. 2022, doi: 10.1371/journal.pone.0262941.
 - [15] N. A. Syam, N. Arifin, W. Firgiawan, and M. F. Rasyid, "Comparison of SVM and Gradient Boosting with PCA for Website Phising Detection," *J. Tek. Inform. (JUTIF)*, vol. 6, no. 2, pp. 691–708, Apr. 2025, doi: 10.52436/1.jutif.2025.6.2.4344.
 - [16] A. K. Kencana, F. D. Ananda, A. D. Hartanto, and H. Hartatik, "Implementasi Metode Random Forest Klasifikasi untuk Phishing Link Detection," *Inf. Tech. J.*, vol. 4, no. 2, pp. 55–59, Dec. 2022, doi: 10.24076/intechnojournal.2022v4i2.1562.
 - [17] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, Apr. 2022, doi: 10.1080/00401706.2021.1921037.
 - [18] N. Mekni, C. Coronello, T. Langer, M. D. Rosa, and U. Perricone, "Support Vector Machine as a Supervised Learning for the Prioritization of Novel Potential SARS-CoV-2 Main Protease Inhibitors," *IJMS*, vol. 22, no. 14, p. 7714, Jul. 2021, doi: 10.3390/ijms22147714.
 - [19] P. V. A G, A. K. K, and V. Varadarajan, "Estimating Software Development Efforts Using a Random Forest-Based Stacked Ensemble Approach," *Electronics*, vol. 10, no. 10, p. 1195, May 2021, doi: 10.3390/electronics10101195.
 - [20] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, 2013, doi: 10.3389/fnbot.2013.00021.
 - [21] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models," in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, Online: Association for Computational Linguistics, 2020, pp. 79–91. doi: 10.18653/v1/2020.eval4nlp-1.9.