Terbit online pada laman web jurnal: https://jurnal.plb.ac.id/index.php/tematik/index

TEMATIK

Jurnal Teknologi Informasi Komunikasi (e-Journal)

Vol. 12 No. 1 (2025) 41 - 49 ISSN Media Elektronik: 2443-3640

Klasifikasi Aktivitas Pengguna yang Berpotensi Menyebabkan Kebocoran Informasi Sensitif Menggunakan Algoritma Random Forest

Classification of User Activities that Potentially Lead to Sensitive Information Leakage Using the Random Forest Algorithm

Alda Amorita Azza[1*], Asep Id Hadiana[2], Agus Komarudin[3]

[1,2,3]Informatika, Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani

[1]aldaamorita21@if.unjani.ac.id, [2]asep.hadiana@lecture.unjani.ac.id, [3]agus.komarudin@lecture.unjani.ac.id

Abstract

Sensitive information leaks are a growing concern in cybersecurity, often caused by insider threats. To address this, a Random Forest classification model was developed to detect user activities that may lead to data leaks. By applying SMOTE-ENN for class balancing and optimizing model parameters, the study achieved remarkable accuracy. The model demonstrated a strong performance with an average F1-Score of 0.9167 in cross-validation and 0.9231 on the test data, reflecting its ability to identify abnormal activities with a balanced approach to precision and recall. Specifically, the model detected abnormal activities with Recall of 94.28%, meaning it effectively identified most of the risky activities while minimizing false positives. The AUC-ROC score of 0.9721 highlights the model's ability to distinguish between normal and abnormal behaviors. The results indicate that Random Forest, paired with SMOTE-ENN and parameter optimization, is an effective tool for detecting data leakage risks and insider threats, with potential for use in information security systems to monitor suspicious activities.

Keywords: Random Forest, SMOTE-ENN, Insider Threat, Sensitive Information Leak, Machine Learning.

Abstrak

Kebocoran informasi sensitif menjadi perhatian utama dalam dunia keamanan siber, yang sering kali disebabkan oleh ancaman dari dalam (*insider threat*). Untuk mengatasi hal ini, sebuah model klasifikasi menggunakan algoritma *Random Forest* dikembangkan untuk mendeteksi aktivitas pengguna yang berpotensi menyebabkan kebocoran data. Dengan penerapan teknik SMOTE-ENN untuk penyeimbangan kelas dan optimasi parameter model, penelitian ini berhasil mencapai akurasi yang sangat baik. Model ini menunjukkan kinerja yang solid dengan *F1-Score* rata-rata sebesar 0.9167 pada *cross-validation* dan 0.9231 pada data uji, yang mencerminkan kemampuannya dalam mendeteksi aktivitas abnormal dengan keseimbangan antara presisi dan *recall*. Secara spesifik, model ini dapat mendeteksi aktivitas abnormal dengan *recall* mencapai 94.28%, yang berarti model berhasil mengidentifikasi hampir semua aktivitas berisiko, sambil meminimalkan *false positives*. Nilai AUC-ROC sebesar 0.9721 menegaskan kemampuan model untuk membedakan antara aktivitas normal dan abnormal. Hasil penelitian ini menunjukkan bahwa *Random Forest*, dikombinasikan dengan SMOTE-ENN dan optimasi parameter, merupakan alat yang efektif untuk mendeteksi potensi kebocoran data dan ancaman dari dalam (*insider threat*), serta dapat diterapkan dalam sistem keamanan informasi untuk memantau aktivitas mencurigakan.

Kata kunci: Random Forest, SMOTE-ENN, Ancaman Dari Dalam, Kebocoran Informasi Sensitif, Pembelajaran Mesin.

1. Pendahuluan

Antara tahun 2012 hingga 2018, telah diidentifikasi berbagai ancaman terhadap sistem siber yang dapat mempengaruhi keamanan informasi, di antaranya adalah *ransomware*, *phishing*, serangan *denial of service*, *spam*, *botnet*, kebocoran data, ancaman dari dalam, manipulasi fisik / kerusakan / kecurian / kehilangan, pencurian identitas, *cryptojacking*, serangan berbasis *web*, serangan aplikasi *web*, mata-mata siber, dan *exploit kits* [1]. Meningkatnya ancaman-ancaman ini semakin mempertegas pentingnya perlindungan data dan implementasi kebijakan keamanan yang proaktif di lingkungan organisasi. Dalam keamanan siber, kebocoran informasi berarti pengiriman data secara tidak sah dari lingkungan internal organisasi menuju pihak eksternal. Salah satu penyebab utama kebocoran informasi sensitif adalah ancaman dari dalam (*insider threat*). Kebocoran informasi ini dapat terjadi baik dengan sengaja oleh karyawan yang memiliki akses sah ke jaringan, sistem, atau data sensitif organisasi, maupun secara tidak sengaja karena kelalaian [2]. Seiring dengan meningkatnya ancaman siber, penting bagi organisasi untuk mengimplementasikan langkah-langkah yang

Diterima Redaksi: 13-05-2025 | Selesai Revisi: 03-06-2025 | Diterbitkan Online: 25-06-2025

kuat untuk melindungi informasi sensitif dari akses yang tidak sah [3].

Mencegah kebocoran informasi sensitif, yang sering disebut *data leak* atau *data loss*, kepada pihak yang tidak berwenang, menjadi tujuan utama dari sistem keamanan informasi di sebuah organisasi [4]. Meskipun kebocoran informasi mungkin tidak selalu bisa dicegah sepenuhnya, berbagai langkah dapat diterapkan untuk mengurangi kemungkinan terjadinya hal tersebut [4], [5]. Selain strategi pencegahan, pendekatan berbasis deteksi juga berperan penting dalam mitigasi kebocoran informasi. Untuk secara efektif mendeteksi ancaman yang ditimbulkan oleh pengguna internal dalam suatu perusahaan, diperlukan penggunaan teknik deteksi anomali terhadap perilaku pengguna [6]. Hampir semua bidang, termasuk keamanan siber, menggunakan metode deteksi anomali untuk menemukan kejadian tidak biasa dalam data. Strategi ini bertujuan mengidentifikasi aktivitas abnormal (aktivitas yang menyimpang) dari pola normal dalam suatu konteks [7].

Dalam upaya meningkatkan efektivitas deteksi anomali, pendekatan berbasis kecerdasan buatan seperti *machine learning* semakin banyak diterapkan dalam keamanan siber [8]. Pendekatan berbasis *machine learning* dapat mengidentifikasi serangan dari dalam dengan menggunakan berbagai metode. Dalam *machine learning*, sistem mempelajari data, mengenali pola-pola perilaku ancaman dari dalam, menganalisis pola tersebut, dan mengambil keputusan berdasarkan analisis data [9]. Model *machine learning* dapat secara efisien memfilter dokumen yang mengandung data sensitif dan memberi peringatan kepada pengguna [10].

Teknik yang banyak digunakan dalam *machine learning* adalah *ensemble learning*, yang menggabungkan beberapa model untuk meningkatkan akurasi dan ketahanan terhadap data yang bervariasi. Algoritma *Random Forest* adalah salah satu metode *ensemble learning* yang menghasilkan banyak pohon keputusan dari sampel data yang dipilih secara acak dimana setiap pohon membuat prediksi, dan solusi terbaik dipilih melalui *voting* [11]. *Random Forest* secara konsisten menggunakan *bootstrapping*, *averaging*, dan *bagging* untuk melatih banyak pohon keputusan. Dengan menggunakan subset karakteristik yang berbeda, sejumlah pohon keputusan independen dapat dibangun secara bersamaan pada segmen-segmen berbeda dari data pelatihan. *Bootstrapping* memastikan bahwa setiap pohon dalam *Random Forest* bersifat unik, sehingga mengurangi *variance* [12]. Keunggulan dari *Random Forest* terletak pada kemampuannya yang dapat diterapkan baik untuk regresi maupun klasifikasi. Model ini juga dikenal cukup efisien, sehingga proses pelatihan dan pengujian dapat dilakukan dengan relatif cepat [13].

Penelitian terdahulu menunjukkan berbagai pendekatan dalam mendeteksi kebocoran informasi sensitif dengan memanfaatkan algoritma *Random Forest* untuk analisis perilaku pengguna. Salah satu penelitian menggunakan algoritma ini untuk mendeteksi ancaman dari dalam dengan menganalisis *log* aktivitas pengguna di jaringan, yang berhasil meningkatkan akurasi deteksi serangan dan menunjukkan efektivitas *Random Forest* dalam mengklasifikasikan ancaman [14]. Penelitian lain yang berfokus pada pencegahan kebocoran data melalui sistem DLP *email* menggunakan *Random Forest* juga berhasil memprediksi sekitar 95% insiden kebocoran data yang sah dengan akurasi tinggi dan mengatasi masalah *false positive* yang umum terjadi pada DLP tradisional [15]. Di sisi lain, terdapat penelitian yang menggunakan analitik data besar untuk mendeteksi pengguna jahat dengan menganalisis pola perilaku mereka di aplikasi *web*. Dalam penelitian ini, *Random Forest* digunakan untuk mengklasifikasikan pengguna berdasarkan aktivitas mereka, dengan akurasi prediksi sekitar 65-70% pada data *real-time*. Proses ini dilakukan langsung dengan memanfaatkan data yang terus mengalir selama lebih dari sebulan, dan model ini dirancang untuk mendeteksi ancaman sebelum serangan terjadi [16].

Meskipun demikian, sebagian besar penelitian masih menghadapi tantangan utama dalam menangani data yang tidak seimbang, variasi perilaku pengguna, serta pemilihan fitur yang tepat untuk klasifikasi yang lebih akurat. Oleh karena itu, penelitian ini bertujuan untuk mengklasifikasikan aktivitas pengguna yang berpotensi menyebabkan kebocoran informasi sensitif dengan menggunakan algoritma *Random Forest*. Penelitian ini akan berfokus pada peningkatan akurasi klasifikasi dengan menangani ketidakseimbangan data serta mengembangkan fitur-fitur baru yang lebih relevan dalam mendeteksi pola anomali yang berisiko. Untuk meningkatkan kinerja model, penelitian ini juga melakukan optimasi terhadap parameter-parameter penting dalam model, yang diharapkan dapat meningkatkan performa deteksi secara keseluruhan. Pendekatan ini bertujuan untuk mengatasi keterbatasan model sebelumnya dan memberikan solusi yang lebih efektif dalam mengklasifikasikan aktivitas yang berpotensi menyebabkan kebocoran informasi sensitif.

2. Metode Penelitian

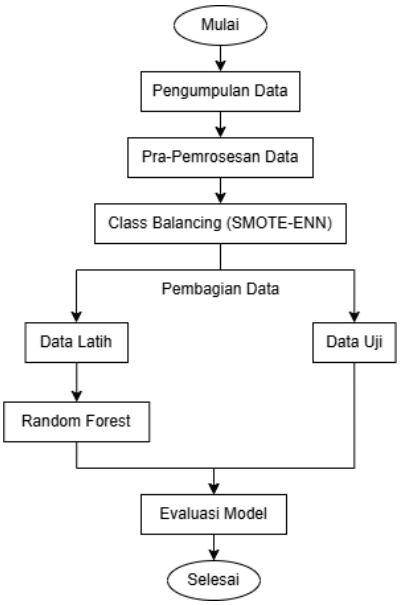
Terdapat beberapa alur tahapan yang akan dilakukan dalam penelitian ini. Rancangan tahapan alur penelitian terdapat pada Gambar 1.

2.1. Pengumpulan Data

Pada tahap pengumpulan data, data yang digunakan dalam penelitian ini adalah dataset bernama "*Data Leakage Detection*" yang diambil dari platform Kaggle, berisi 49.500 aktivitas pengguna dalam jaringan komputer dan sistem yang mencakup informasi penting seperti pada Tabel 1. *Abnormality* digunakan sebagai label yang menunjukkan apakah suatu aktivitas

DOI: https://doi.org/10.38204/tematik.v12i1.2325

dianggap abnormal atau tidak. Nilai 1 menunjukkan bahwa aktivitas tersebut dikategorikan sebagai abnormal (ya), yang berarti terdapat indikasi perilaku mencurigakan atau tidak biasa dalam jaringan komputer dan sistem, seperti akses tidak sah ke data sensitif atau transfer file ke tujuan eksternal. Sebaliknya, nilai 0 berarti aktivitas tersebut dianggap normal (tidak), yang menunjukkan bahwa tindakan pengguna tidak menimbulkan potensi risiko atau ancaman keamanan.



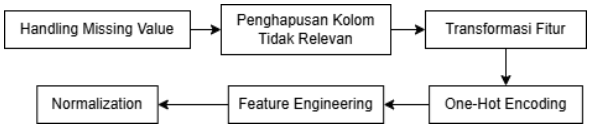
Gambar 1. Metode Penelitian

Tabel 1. Fitur-Fitur dalam Dataset "*Data Leakage Detection*"

No	Nama Fitur	Deskripsi
1	*id*	Identifikasi unik untuk setiap catatan dalam dataset.
2	*date*	Tanggal dan waktu ketika aktivitas tercatat.
3	*user*	Identifikasi pengguna, menunjukkan individu yang terkait dengan aktivitas tersebut.
4	*pc*	Identifikasi PC, yang menunjukkan komputer yang terlibat dalam aktivitas yang tercatat.
5	*Authority*	Tingkat otoritas atau peran pengguna (misalnya, manajer, teknisi utama, manajer senior).
6	*Through _pwd*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah pengguna mengakses sistem melalui kata sandi.
7	*Through _pin*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah pengguna mengakses sistem melalui PIN.
8	*Through _MFA*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah pengguna mengakses sistem melalui *Multi-Factor Authentication* (MFA).
9	*Data Modification*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah ada modifikasi data selama aktivitas yang tercatat.
10	*Confidential Data Access*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah pengguna mengakses data yang bersifat rahasia.
11	*Confidential File Transfer*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah ada transfer file rahasia selama aktivitas yang tercatat.
12	*External Destination*	Indikator biner (1.0 atau 0.0) yang menunjukkan apakah aktivitas melibatkan tujuan eksternal.
13	*File Operation*	Jenis operasi file yang dilakukan selama aktivitas (misalnya, *move*, *write*, *read*).
14	*Data Sensitivity Level*	Tingkat sensitivitas data yang diakses (misalnya, rendah, tinggi).
15	*Abnormality*	Indikator biner (1 atau 0) yang menunjukkan apakah aktivitas yang tercatat dianggap tidak normal.

2.2. Pra-Pemrosesan Data

Pra-pemrosesan data adalah tahapan awal dalam siklus pengolahan data, yang bertujuan untuk membersihkan dan memastikan data agar siap digunakan dalam model pembelajaran mesin [17]. Pra-pemrosesan mencakup beberapa tahap, dapat dilihat pada Gambar 2.



Gambar 2. Pra-Pemrosesan Data

Tahap pra-pemrosesan data dimulai dengan menangani data yang hilang menggunakan teknik imputasi. Kolom yang tidak relevan dihapus agar tidak memengaruhi model. Lalu, dilakukan transformasi fitur dari kolom tanggal menjadi fitur baru seperti jam, hari, bulan, dan tahun untuk mendeteksi pola waktu yang relevan.

Kemudian, *One-Hot Encoding* diterapkan pada kolom kategorikal untuk mengubahnya menjadi format numerik yang dapat diproses oleh model. Pada tahap *feature engineering*, beberapa fitur baru dikembangkan untuk menambah relevansi data. Normalisasi juga dilakukan agar data memiliki skala yang konsisten. Terakhir, untuk mengatasi ketidakseimbangan data, SMOTE-ENN (*Synthetic Minority Oversampling Technique-Edited Nearest Neighbors*) diterapkan untuk *oversampling* dan *undersampling* secara bersamaan [18]. Setelah pra-pemrosesan selesai, data dibagi menjadi data pelatihan (80%) yang digunakan untuk melatih model, dan data pengujian (20%) yang digunakan untuk mengevaluasi kinerja model.

2.3. *Random Forest*

Pada tahapan pelatihan model, setelah pra-pemrosesan data selesai, model *Random Forest* dilatih menggunakan data pelatihan yang telah diproses. Dalam proses ini, beberapa pohon keputusan dibangun secara bersamaan dengan menggunakan data yang berbeda-beda. Setiap pohon membuat prediksi berdasarkan pola yang ditemukan dalam segmen data, dan hasil dari semua pohon ini digabungkan untuk menghasilkan prediksi yang lebih akurat.

DOI: https://doi.org/10.38204/tematik.v12i1.2325

Setelah model dilatih, langkah selanjutnya adalah memilih fitur yang paling berpengaruh terhadap prediksi. Teknik ini memanfaatkan model *Random Forest* untuk memilih fitur yang paling relevan dan membuang fitur yang redundan serta tidak relevan untuk klasifikasi, agar dapat mempertahankan informasi yang terkandung dalam seluruh set variabel *input* terkait dengan kelas target, sehingga membantu meningkatkan performa model [19].

Untuk meningkatkan kinerja model secara keseluruhan, dilakukan pencarian *hyperparameter* untuk memastikan bahwa model bekerja dengan pengaturan terbaik, sehingga akurasi dan efektivitasnya meningkat [20]. Selanjutnya, untuk mengevaluasi kinerja model yang lebih general dan menghindari *overfitting*, dilakukan evaluasi dengan membagi data menjadi beberapa *fold* yang seimbang, memastikan bahwa proporsi kelas dalam setiap *fold* terjaga [21]. Skor F1 dihitung pada setiap *fold* untuk mendapatkan hasil yang lebih stabil dan meminimalkan variasi dalam evaluasi model.

Terakhir, dilakukan *ensemble training* dengan melatih beberapa model *Random Forest* yang berbeda. Setiap model dilatih menggunakan parameter yang berbeda dengan berbagai nilai *seed*. Setelah model-model tersebut dilatih, prediksi dari masing-masing model digabungkan berdasarkan bobot model untuk menghasilkan prediksi akhir yang lebih kuat dan *robust*, sehingga meningkatkan performa deteksi kebocoran informasi sensitif.

2.4. Evaluasi Model

Tahap akhir dari penelitian ini adalah melakukan evaluasi dan mengukur kinerja dari berbagai skenario model yang diusulkan [22].

Tabel 2. *Confusion Matrix*

Aktual	Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Tabel 2 menggambarkan *confusion matrix* yang digunakan sebagai alat untuk mengukur dan mengevaluasi kinerja klasifikasi, terutama dalam kasus klasifikasi dua kelas (*binary classification*). *Confusion matrix* berfungsi untuk membandingkan hasil prediksi sistem dengan nilai aktual dalam dataset. Evaluasi dilakukan dengan mengacu pada komponen-komponen dalam *confusion matrix*, yang melibatkan jumlah nilai pada TP (*True Positive*), FP (*False Positive*), TN (*True Negative*), dan FN (*False Negative*) [22].

TP menggambarkan jumlah prediksi positif yang benar, yaitu data yang diprediksi positif dan memang benar positif secara aktual. FP merujuk pada jumlah prediksi positif yang salah, yaitu data yang diprediksi positif tetapi sebenarnya negatif. FN menunjukkan jumlah data yang diprediksi negatif namun sebenarnya positif, sementara TN menggambarkan jumlah prediksi negatif yang benar, yaitu data yang diprediksi negatif dan memang negatif secara aktual [23].

Setelah nilai dari setiap komponen dihitung, dilakukan evaluasi kinerja klasifikasi dengan menggunakan metrik seperti akurasi, presisi, *recall*, *F1-Score*, dan *Area Under Curve* (AUC). Kelima metrik ini dipilih karena kemampuannya yang menyeluruh, khususnya untuk menangani masalah kelas dalam dataset yang tidak seimbangan [22].

Akurasi adalah metrik yang digunakan untuk menilai perbandingan antara proporsi prediksi yang akurat oleh sistem (TP dan TN) dengan total prediksi yang dihasilkan (TP, FP, FN, TN) [22]. Nilai Akurasi dapat dihitung dengan menggunakan Persamaan (1).

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (1)$$

Selanjutnya, presisi digunakan untuk menilai seberapa besar proporsi prediksi nilai positif yang sesuai dengan nilai sebenarnya (TP) dibandingkan dengan keseluruhan hasil prediksi yang bernilai positif (TP dan FP) [22]. Nilai Presisi dapat dihitung dengan menggunakan Persamaan (2).

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

Metrik lain yang digunakan adalah *recall*, yang mengukur proporsi data yang diprediksi sebagai positif (TP) dibandingkan dengan total data yang sebenarnya bernilai positif (TP dan FN) [22]. Nilai *Recall* dapat dihitung dengan menggunakan Persamaan (3).

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

Kemudian, *F1-Score* adalah metrik yang digunakan untuk mengukur rata-rata perbandingan antara presisi dan *recall* [22]. Nilai *F1-Score* dapat dihitung dengan menggunakan Persamaan (4).

$$F1 - Score = 2 \text{ x } \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

Terakhir, *Area Under Curve* (AUC) digunakan untuk menilai apakah model yang dihasilkan tergolong sebagai klasifikasi yang efektif atau tidak efektif [22].

3. Hasil dan Pembahasan

Rangkaian proses analisis dilakukan secara bertahap untuk membangun model yang mampu mengidentifikasi aktivitas berisiko terhadap kebocoran data sensitif. Dimulai dari tahap pra-pemrosesan data hingga evaluasi model, setiap langkah dirancang untuk meningkatkan kualitas data dan akurasi prediksi. Hasil dari setiap tahapan menunjukkan bagaimana transformasi data, penyeimbangan kelas, serta pemilihan algoritma berperan dalam menghasilkan

DOI: https://doi.org/10.38204/tematik.v12i1.2325

model yang efektif dalam mendeteksi pola-pola mencurigakan.

3.1. Pra-Pemrosesan

Tahap awal yang dilakukan dalam pengolahan data adalah melakukan pra-pemrosesan. Terdapat 5 tahap dalam pra-pemrosesan yang akan dilakukan.

Handling Missing Value: Pada tahap ini, data yang hilang pada kolom tertentu ditangani menggunakan teknik imputasi. Kolom yang berisi data biner seperti *Through_pwd*, *Through_pin*, *Through_MFA*, *Data Modification*, *Confidential Data Access*, dan *Confidential File Transfer* diisi dengan nilai 0, sementara kolom kategorikal seperti *Authority*, *External Destination*, *File Operation*, dan *Data Sensitivity Level* diisi dengan modus (nilai yang paling sering terjadi) pada setiap kolom. Langkah ini diambil untuk memastikan bahwa data yang hilang tidak mengganggu proses pelatihan model dan mengurangi bias dalam data. Perbandingan hasil dari sebelum dan sesudah penanganan data yang hilang dapat dilihat pada Gambar 3 dan Gambar 4.

	Missing Values
id	0
date	497
user	496
pc	498
Authority	493
Through_pwd	498
Through_pin	494
Through_MFA	353
Data Modification	494
Confidential Data Access	500
Confidential File Transfer	495
External Destination	494
File Operation	494
Data Sensitivity Level	482
Abnormality	0

Gambar 3. Sebelum *Handling Missing Value*

	Missing Values
id	0
date	497
user	496
pc	498
Authority	0
Through_pwd	0
Through_pin	0
Through_MFA	0
Data Modification	0
Confidential Data Access	0
Confidential File Transfer	0
External Destination	0
File Operation	0
Data Sensitivity Level	0
Abnormality	0

Gambar 4. Setelah *Handling Missing Value*

Pembuangan Kolom yang Tidak Relevan: Kolom yang tidak relevan untuk model, seperti *id*, *date*, *user*, dan *pc*, dihapus dari dataset. Kolom-kolom ini dianggap tidak memberikan informasi signifikan terhadap prediksi apakah suatu aktivitas berpotensi menyebabkan kebocoran informasi sensitif, sehingga penghapusan kolom ini memungkinkan model untuk fokus pada fitur-fitur yang lebih relevan. Hal ini juga membantu mengurangi kompleksitas data yang akan diproses.

Transformasi Fitur Tanggal: Kolom *date* yang berisi informasi waktu diubah menjadi beberapa fitur baru seperti *hour*, *day*, *month*, *year*, dan *dayofweek*. Fitur-fitur ini bertujuan untuk memungkinkan model mendeteksi pola yang berhubungan dengan waktu, seperti aktivitas pada jam kerja atau akhir pekan.

One-Hot Encoding: Untuk fitur-fitur kategorikal seperti *Authority*, *External Destination*, *File Operation*, dan *Data Sensitivity Level*, *One-Hot Encoding* diterapkan untuk mengubah kategori menjadi format numerik yang dapat diproses oleh model. Proses ini memungkinkan model untuk bekerja dengan data numerik yang dapat diolah oleh algoritma *Random Forest*.

Feature Engineering: Beberapa fitur baru dikembangkan untuk meningkatkan kemampuan model dalam memprediksi aktivitas yang berpotensi abnormal. Fitur-fitur baru tersebut dapat dilihat pada Tabel 3.

Tabel 3. *Feature Engineering*

No	Nama Fitur	Deskripsi
1	*total_auth*	Jumlah cara otentikasi yang digunakan oleh pengguna (kata sandi, PIN, atau MFA).
2	*risk_score*	Menilai tingkat risiko berdasarkan akses data sensitif dan transfer file sensitif.
3	*working_hour*	Menandai apakah aktivitas dilakukan pada jam kerja (07:00-18:00).
4	*weekend*	Mengidentifikasi apakah aktivitas terjadi pada akhir pekan.
5	*is_night*	Menandai apakah aktivitas terjadi pada malam hari.

Fitur-fitur ini dikembangkan untuk menambah relevansi data dan memperkaya model dalam mendeteksi pola-pola yang dapat memicu kebocoran informasi sensitif.

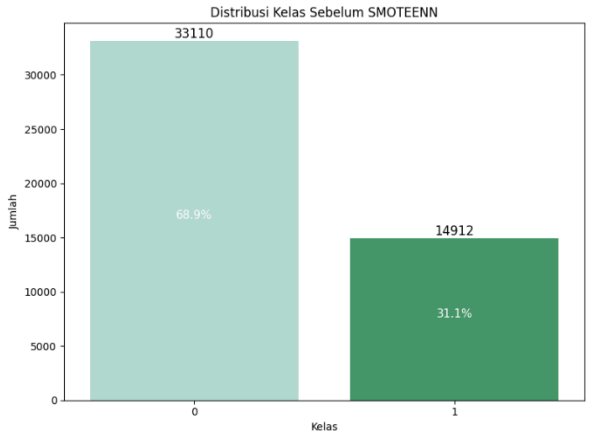
Normalisasi Data: Normalisasi dilakukan menggunakan *StandardScaler*, yang memastikan bahwa semua fitur berada pada skala yang seragam (rata-rata 0 dan standar deviasi 1). Hal ini penting agar model tidak terpengaruh oleh perbedaan skala antar fitur.

3.2. *Class Balancing* (SMOTE-ENN)

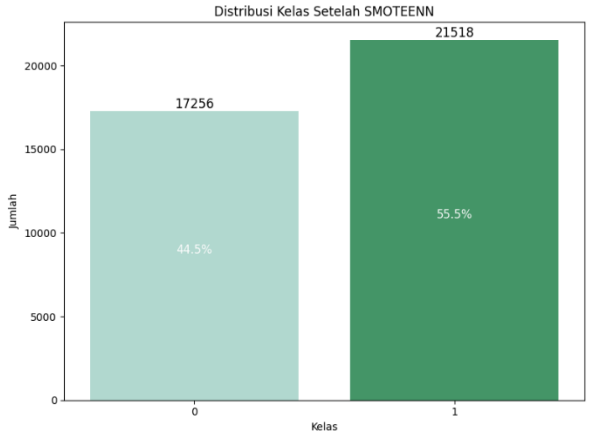
Selanjutnya, untuk mengatasi kelas yang tidak seimbangan, diterapkan teknik SMOTE-ENN (*Synthetic Minority Over-sampling Technique-Edited Nearest Neighbors*). SMOTE menghasilkan data sintetis dari kelas minoritas, sementara ENN menghapus data yang ambigu dari kelas mayoritas. Hasilnya adalah distribusi kelas yang lebih seimbang, sehingga model tidak bias terhadap kelas mayoritas. Distribusi kelas pada dataset sebelum dan sesudah

DOI: https://doi.org/10.38204/tematik.v12i1.2325

penerapan teknik SMOTE-ENN dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Distribusi Kelas Sebelum SMOTE-ENN



Gambar 6. Distribusi Kelas Setelah SMOTE-ENN

Pada dataset asli, distribusi kelas tidak seimbang dengan kelas normal (label 0) sebanyak 33.110 data (68,9%) dan kelas abnormal (label 1) sebanyak 14.912 data (31,1%). Kondisi ketidakseimbangan ini dapat menyebabkan model menjadi bias terhadap kelas mayoritas sehingga mengurangi kemampuan deteksi aktivitas berisiko pada kelas minoritas. Setelah penerapan SMOTE-ENN, kelas minoritas di-*oversample* untuk menghasilkan data sintetis, sementara kelas mayoritas mengalami pengurangan data ambigu. Sehingga distribusi kelas menjadi lebih seimbang dengan 17.256 data (44,5%) pada kelas normal dan 21.518 data (55,5%) pada kelas abnormal. Dengan distribusi kelas yang lebih seimbang ini, model memiliki peluang lebih baik untuk mempelajari pola dari kedua kelas secara proporsional dan meningkatkan kinerja dalam mendeteksi aktivitas yang berpotensi menyebabkan kebocoran informasi sensitif.

3.3. Pembagian Data

Pada tahap ini, data yang telah melalui proses pra-pemrosesan dibagi menjadi dua bagian utama, yaitu data pelatihan (*training data*) yang digunakan untuk melatih model dan data pengujian (*test data*) yang digunakan untuk mengevaluasi kinerja model setelah proses pelatihan selesai. Data pelatihan terdiri dari 80% dari total data yang ada, sedangkan data pengujian mencakup 20% sisanya. Pembagian ini bertujuan untuk memastikan bahwa model tidak belajar dari data yang akan diuji, serta untuk mengevaluasi sejauh mana model mampu menggeneralisasi dari data pelatihan ke data yang belum pernah dijumpai sebelumnya. Pembagian dataset dapat dilihat pada Tabel 4.

Tabel 4. Pembagian Data

Data Latih (80%)	Data Uji (20%)
31019 Dataset	7755 Dataset

Setelah pembagian data, tahap berikutnya adalah pelatihan model. Model *RandomForestClassifier* dilatih menggunakan data pelatihan yang telah diproses. Pelatihan ini dilakukan dengan menggunakan sejumlah pohon keputusan, yang jumlahnya diatur dengan parameter *n_estimators*=200 (jumlah pohon keputusan dalam hutan acak). Setiap pohon keputusan dibangun dengan menggunakan segmen berbeda dari data pelatihan, memungkinkan model untuk mempelajari berbagai pola dalam data. Proses ini membantu model mempelajari hubungan antara fitur dan label dalam data, yang kemudian digunakan untuk membuat prediksi pada data yang tidak terlihat.

Lalu, dilakukan seleksi fitur. Dalam hal ini, *Random Forest* digunakan untuk memilih fitur yang paling penting dalam membuat prediksi. Fitur yang kurang relevan atau berlebihan dapat menyebabkan model menjadi terlalu rumit dan rentan terhadap *overfitting*. *SelectFromModel* digunakan untuk menilai pentingnya fitur berdasarkan pelatihan model dengan *Random Forest*, dan hanya fitur yang lebih penting yang dipertahankan untuk melatih model lebih lanjut.

3.4. *Random Forest*

Setelah proses pelatihan model dilakukan, optimasi dilakukan menggunakan *RandomizedSearchCV* untuk menemukan kombinasi parameter terbaik. Parameter yang dioptimasi termasuk jumlah pohon (*n_estimators*), kedalaman pohon (*max_depth*), serta parameter lainnya seperti *min_samples_split* dan *min_samples_leaf*. Proses pencarian ini memungkinkan model untuk menemukan pengaturan yang menghasilkan akurasi prediksi terbaik.

Berikut adalah beberapa parameter utama yang telah dioptimasi:

n_estimators: 4000 pohon keputusan digunakan dalam ensemble model untuk meningkatkan keakuratan.

max_depth: Kedalaman maksimum pohon dibatasi pada 40 untuk menghindari *overfitting*.

DOI: https://doi.org/10.38204/tematik.v12i1.2325

min_samples_split dan *min_samples_leaf* diatur untuk mengatur jumlah data minimum pada setiap pembelahan dan pada daun untuk mengurangi *varians*.

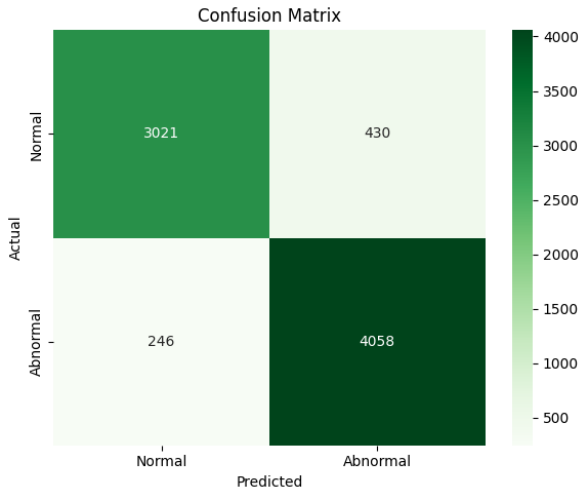
Cross-validation juga diterapkan menggunakan *StratifiedKFold* untuk membagi dataset menjadi 5 *fold* yang seimbang. Teknik ini memastikan bahwa model dievaluasi secara adil dengan membagi data ke dalam bagian yang lebih kecil dan menguji model pada masing-masing *fold* secara bergantian. Hasil *cross-validation* menunjukkan *F1-Score* rata-rata sebesar 0.9167, yang menandakan kinerja yang sangat baik dalam mendeteksi pola-pola mencurigakan.

Setelah itu, dilakukan *Ensemble Training* dengan melatih beberapa model *Random Forest* dengan parameter yang berbeda, menggunakan seed yang bervariasi. Setiap model dalam ensemble ini memberikan kontribusi terhadap prediksi akhir, dengan bobot yang dihitung berdasarkan AUC-ROC. Bobot ini membantu meningkatkan akurasi keseluruhan model dengan memanfaatkan beberapa model yang dilatih secara paralel.

Langkah terakhir adalah *Threshold Selection* untuk memilih *threshold* optimal berdasarkan *Precision-Recall curve*. Dengan memilih nilai *threshold* yang menghasilkan keseimbangan terbaik antara *precision* dan *recall*, nilai *F1-Score* dapat dioptimalkan lebih lanjut.

3.5. Evaluasi Model

Pada tahap evaluasi, model yang telah dilatih diuji untuk menilai kinerjanya menggunakan beberapa metrik evaluasi. Berdasarkan hasil pengujian menggunakan data uji, diperoleh *confusion matrix* yang terdapat pada Gambar 8. Gambar ini memberikan gambaran visual mengenai kinerja model dalam mengklasifikasikan aktivitas pengguna yang berpotensi menyebabkan kebocoran informasi sensitif.



Gambar 7. *Confusion Matrix*

Berdasarkan *confusion matrix*, berikut adalah hasil yang dapat diperoleh:

True Positives (TP): 4058 (Aktivitas abnormal yang terdeteksi dengan benar sebagai abnormal).

True Negatives (TN): 3021 (Aktivitas normal yang terdeteksi dengan benar sebagai normal).

False Positives (FP): 430 (Aktivitas normal yang salah terdeteksi sebagai abnormal).

False Negatives (FN): 246 (Aktivitas abnormal yang salah terdeteksi sebagai normal).

Tabel 4. *Classification Report*

Class	*Precision*	*Recall*	*F1-Score*	*Support*
Normal (0)	0.92	0.88	0.90	3451
Abnormal (1)	0.90	0.94	0.92	4304
Accuracy			0.91	7755
Macro Avg	0.91	0.91	0.91	7755
Weighted avg	0.91	0.91	0.91	7755

Nilai-nilai metrik evaluasi pada Tabel 4 dihitung berdasarkan nilai *True Positive*, *True Negative*, *False Positive*, dan *False Negative* yang diperoleh dari *confusion matrix*. Nilai Akurasi dapat dihitung dengan menggunakan Persamaan (1).

$$Accuracy = \frac{4058+3021}{4058+3021+430+246} = 0.9128 \quad (1)$$

Nilai Presisi dapat dihitung dengan menggunakan Persamaan (2).

$$Precision = \frac{4058}{4058+430} = 0.9042 \quad (2)$$

Nilai *Recall* dapat dihitung dengan menggunakan Persamaan (3).

$$Recall = \frac{4058}{4058+246} = 0.9428 \quad (3)$$

Nilai *F1-Score* dapat dihitung dengan menggunakan Persamaan (4).

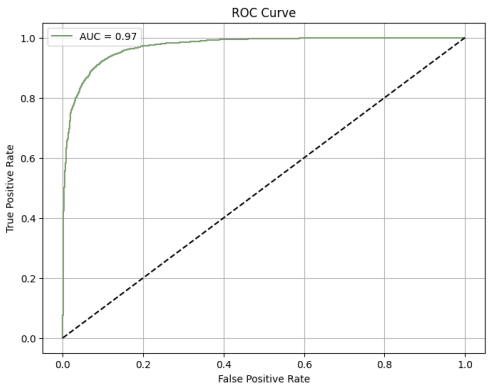
$$F1 - Score = 2 \times \frac{0.9042 \times 0.9428}{0.9042+0.9428} = 0.9231 \quad (4)$$

Terakhir, nilai *Area Under Curve* (AUC) untuk *Receiver Operating Characteristic* (ROC) juga digunakan untuk menilai efektivitas model klasifikasi, dengan hasil sebesar 0.9721 menunjukkan kemampuan model yang sangat baik dalam membedakan kelas normal dan abnormal. ROC *Curve* dapat dilihat pada Gambar 8.

Dari hasil evaluasi yang diperoleh, dapat dilihat bahwa model menunjukkan performa yang memuaskan dalam mengklasifikasikan aktivitas yang berpotensi menyebabkan kebocoran informasi sensitif, dengan jumlah *False Positives* dan *False Negatives* yang relatif rendah. Model mencapai akurasi sebesar 91,28%, yang berarti model mampu memprediksi dengan benar sekitar 91% dari seluruh data. Namun, dalam konteks dataset yang tidak seimbang, akurasi sendiri belum

DOI: https://doi.org/10.38204/tematik.v12i1.2325

cukup untuk menggambarkan performa karena model mungkin lebih mengutamakan kelas mayoritas. Oleh karena itu, metrik lain perlu diperhatikan untuk evaluasi yang lebih komprehensif.



Gambar 8. ROC *Curve*

Nilai presisi sebesar 90,42% menunjukkan bahwa dari seluruh aktivitas yang diprediksi sebagai abnormal oleh model, sekitar 90% benar-benar merupakan aktivitas abnormal. Di sisi lain, hal ini juga berarti terdapat sekitar 9,58% prediksi yang merupakan *False Positives*, yaitu aktivitas normal yang keliru diklasifikasikan sebagai abnormal. Meskipun angka ini relatif kecil, jumlah 430 kasus *False Positives* pada data uji menunjukkan bahwa model masih cukup sensitif dalam mengidentifikasi potensi risiko.

Sementara itu, *Recall* yang mencapai 94,28% mengindikasikan bahwa model mampu menangkap hampir seluruh aktivitas abnormal yang sebenarnya terjadi, dengan jumlah *False Negatives* yang rendah. Strategi ini mencerminkan prioritas model dalam meningkatkan sensitivitas deteksi, guna memastikan bahwa aktivitas berisiko tidak terabaikan.

Namun, tingginya *Recall* tersebut berpotensi menimbulkan *trade-off* dengan presisi. Dalam kasus ini, strategi model yang memprioritaskan sensitivitas tinggi menyebabkan munculnya *False Positives* yang masih cukup signifikan. *Trade-off* ini merupakan kondisi umum dalam klasifikasi risiko dimana menurunkan *threshold* klasifikasi untuk menangkap lebih banyak kasus positif meningkatkan *recall*, tetapi juga menimbulkan lebih banyak *False Positives* [24]. Faktor lain yang memengaruhi *False Positives* adalah karakteristik fitur yang mirip antara aktivitas normal dan abnormal, sehingga model terkadang kesulitan membedakan secara sempurna.

F1-Score, yang merupakan harmonisasi antara presisi dan *recall*, tercatat sebesar 92,31%, menunjukkan performa keseluruhan model cukup stabil dalam menangani baik deteksi aktivitas berisiko maupun meminimalkan kesalahan prediksi. Hal ini mendukung efektivitas model dalam menangani ketidakseimbangan kelas yang ada.

Kemudian, nilai AUC-ROC yang sangat tinggi yaitu 0,9721 menandakan kemampuan model yang kuat dalam membedakan antara kelas normal dan abnormal di berbagai *threshold* klasifikasi. Sementara ini menjadi indikator positif, angka AUC-ROC yang sangat tinggi juga berpotensi menunjukkan risiko *overfitting*, terutama jika perbedaan performa antara data pelatihan dan pengujian cukup besar. Namun, dengan penerapan optimasi *hyperparameter* dan *cross-validation* selama pelatihan, risiko overfitting dapat diminimalisasi.

4. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, model klasifikasi menggunakan algoritma *Random Forest* berhasil dibangun untuk mendeteksi aktivitas pengguna yang berpotensi menyebabkan kebocoran informasi sensitif. Melalui serangkaian proses, mulai dari pra-pemrosesan data, penyeimbangan kelas dengan teknik SMOTE-ENN, hingga optimasi *hyperparameter model*, akurasi yang tinggi berhasil dicapai. Model ini menunjukkan kinerja yang sangat baik dengan *F1-Score* rata-rata sebesar 0.9167 pada *cross-validation* dan 0.9231 pada data uji, yang mengindikasikan bahwa model mampu mendeteksi aktivitas abnormal dengan presisi dan *recall* yang seimbang.

Dari hasil evaluasi ini, dapat disimpulkan bahwa *Random Forest* dengan penerapan SMOTE-ENN dan teknik optimasi *hyperparameter model* dapat menjadi solusi yang efektif dalam mendeteksi potensi kebocoran data dan ancaman dari dalam (*insider threat*) dalam sistem informasi. Hasil ini menunjukkan potensi penerapan model pada berbagai lingkungan dengan kebutuhan pengawasan keamanan informasi yang intensif dan berkelanjutan. Model ini dapat diintegrasikan dalam sistem keamanan informasi, menyediakan pengawasan aktivitas pengguna secara berkelanjutan sekaligus mendukung analisis berkala yang memperkuat audit dan mitigasi risiko kebocoran data. Penerapan model sangat potensial di sektor dengan regulasi ketat terhadap data sensitif seperti perbankan, layanan kesehatan, perusahaan teknologi informasi, dan instansi pemerintahan yang membutuhkan perlindungan data di lingkungan kerja yang dinamis dan kompleks. Fleksibilitas model memungkinkan implementasi pada berbagai skala dan kompleksitas infrastruktur TI, dari pengawasan terpusat hingga sistem terdistribusi, sehingga organisasi dapat lebih efektif mengantisipasi ancaman dari dalam dengan memanfaatkan *machine learning* yang adaptif dan responsif terhadap pola-pola aktivitas pengguna.

Sebagai saran, penelitian selanjutnya dapat mengeksplorasi *ensemble learning* atau kombinasi berbagai model untuk meningkatkan akurasi dan *robustnes* model lebih lanjut. Penerapan teknik seperti *stacking*, *boosting*, atau *bagging* dapat membantu mengatasi variasi dalam data dan meningkatkan kinerja

DOI: https://doi.org/10.38204/tematik.v12i1.2325

model secara keseluruhan.

Daftar Rujukan

[1] K. Inayah and K. Ramli, "Analisis Kinerja Intrusion Detection System Berbasis Algoritma Random Forest Menggunakan Dataset Unbalanced Honeynet BSSN," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 4, pp. 867–876, Aug. 2024, doi: 10.25126/jtiik.1148911.

[2] I. Herrera Montano, J. J. García Aranda, J. Ramos Diaz, S. Molina Cardín, I. de la Torre Díez, and J. J. P. C. Rodrigues, "Survey of Techniques on Data Leakage Protection and Methods to address the Insider threat," *Cluster Comput*, vol. 25, no. 6, pp. 4289–4302, Dec. 2022, doi: 10.1007/s10586-022-03668-2.

[3] O. Arerebo Profit, M. Ifeanyi, and E. Abel, "HANDLING THREAT DETECTION AND PREVENTION VIA RANDOM FOREST AND XGBOOST FOR SENSITIVE DATA SECURITY AND PRIVACY-PRESERVING SYSTEM," Aug. 2024. Accessed: Nov. 24, 2024. [Online]. Available: https://www.researchgate.net/publication/383040258_HANDLING_THREAT_DETECTION_AND_PREVENTION_VIA_RANDOM_FOREST_AND_XGBOOST_FOR_SENSITIVE_DATA_SECURITY_AND_PRIVACY-PRESERVING_SYSTEM

[4] M. Soleh and Z. Tjenreng, "Strategi Pencegahan Kebocoran Data Pelayanan Publik Di Era Digital," *Jurnal Kajian Pemerintah: Journal of Government, Social and Politics*, vol. 11, no. 1, pp. 1–10, Dec. 2024, Accessed: May 12, 2025. [Online]. Available: https://journal.uir.ac.id/index.php/JKP/article/view/20524

[5] A. Guha, D. Samanta, A. Banerjee, and D. Agarwal, "A Deep Learning Model for Information Loss Prevention From Multi-Page Digital Documents," *IEEE Access*, vol. 9, pp. 80451–80465, 2021, doi: 10.1109/ACCESS.2021.3084841.

[6] W. Feng *et al.*, "Multi-Granularity User Anomalous Behavior Detection," *Applied Sciences*, vol. 15, no. 1, p. 128, Dec. 2024, doi: 10.3390/app15010128.

[7] U. Ahmed *et al.*, "Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering," *Sci Rep*, vol. 15, no. 1, p. 1726, Jan. 2025, doi: 10.1038/s41598-025-85866-7.

[8] A. F. Mahmud and S. Wirawan, "Phishing Website Detection Using Machine Learning Classification Method," *SISTEMASI*, vol. 13, no. 4, p. 1368, Jul. 2024, doi: 10.32520/stmsi.v13i4.3456.

[9] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J Big Data*, vol. 7, no. 1, p. 41, Dec. 2020, doi: 10.1186/s40537-020-00318-5.

[10] Mosope Williams and Tina Charles Mbakwe-Obi, "Integrated strategies for database protection: Leveraging anomaly detection and predictive modelling to prevent data breaches," *World Journal of Advanced Research and Reviews*, vol. 24, no. 3, pp. 1098–1115, Dec. 2024, doi: 10.30574/wjarr.2024.24.3.3795.

[11] S. Al and M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Comput Secur*, vol. 110, p. 102435, Nov. 2021, doi: 10.1016/j.cose.2021.102435.

[12] A. K. Balyan *et al.*, "A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method," *Sensors*, vol. 22, no. 16, p. 5986, Aug. 2022, doi: 10.3390/s22165986.

[13] T. Al-Shehari, M. Al-Razgan, T. Alfakih, R. A. Alsowail, and S. Pandiaraj, "Insider Threat Detection Model Using Anomaly-Based Isolation Forest Algorithm," *IEEE Access*, vol. 11, pp. 118170–118185, 2023, doi: 10.1109/ACCESS.2023.3326750.

[14] H. Teymourlouei and V. E. Harris, "Preventing Data Breaches: Utilizing Log Analysis and Machine Learning for Insider Attack Detection," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Dec. 2022, pp. 1022–1027. doi: 10.1109/CSCI58124.2022.00181.

[15] M. F. Faiz, J. Arshad, M. Alazab, and A. Shalaginov, "Predicting likelihood of legitimate data loss in email DLP," *Future Generation Computer Systems*, vol. 110, pp. 744–757, Sep. 2020, doi: 10.1016/j.future.2019.11.004.

[16] R. Ranjan and S. S. Kumar, "User behaviour analysis using data analytics and machine learning to predict malicious user versus legitimate user," *High-Confidence Computing*, vol. 2, no. 1, p. 100034, Mar. 2022, doi: 10.1016/j.hcc.2021.100034.

[17] Muttaqin *et al.*, *Pengenalan Data Mining*. Yayasan Kita Menulis, 2023.

[18] I. Riantika, B. Sartono, and K. Anwar Notodiputro, "Effectiveness of SMOTE-ENN to Reduce Complexity in Classification Model," *Indonesian Journal of Statistics and Its Applications*, vol. 8, no. 1, pp. 70–82, Jun. 2024, doi: 10.29244/ijsa.v8i1p70-82.

[19] G. Sosa-Cabrera, S. Gómez-Guerrero, M. García-Torres, and C. E. Schaerer, "Feature selection: a perspective on inter-attribute cooperation," *Int J Data Sci Anal*, vol. 17, no. 2, pp. 139–151, Mar. 2024, doi: 10.1007/s41060-023-00439-z.

[20] Sheena p Shaji, Renju R, Julie Varghese, Lakshmi Sathyan, and Dhannya J, "Optimizing Hyperparameters: Techniques for Improving Machine Learning Models," *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 2, no. 12, pp. 3782–3787, Dec. 2024, doi: 10.47392/IRJAEM.2024.0561.

[21] M. Bhagat and Dr. Brijesh Bakariya, "A Comprehensive Review of Cross-Validation Techniques in Machine Learning," *International Journal on Science and Technology*, vol. 16, no. 1, Jan. 2025, doi: 10.71097/IJSAT.v16.i1.1305.

[22] A. Ferdita Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *Jurnal Infomedia*, vol. 7, no. 1, p. 39, Jun. 2022, doi: 10.30811/jim.v7i1.2959.

[23] T. F. Monaghan *et al.*, "Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value," *Medicina (B Aires)*, vol. 57, no. 5, p. 503, May 2021, doi: 10.3390/medicina57050503.

[24] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," *Expert Syst Appl*, vol. 193, p. 116429, May 2022, doi: 10.1016/j.eswa.2021.116429.

DOI: https://doi.org/10.38204/tematik.v12i1.2325