Jurnal Teknik Informatika (JUTIF)
Vol. 6, No. 1, February 2025, pp. 87-104
DOI: https://doi.org/10.52436/1.jutif.2025.6.1.4175
p-ISSN: 2723-3863
e-ISSN: 2723-3871

EFFECTIVENESS HYPERPARAMETER TUNING ON RANDOM FOREST, LINEAR DISCRIMINANT ANALYSIS, LOGISTIC REGRESSION AND NAÏVE BAYES ALGORITHMS FOR DETECTING DOS NETWORK ATTACKS

Inka Saputri[1], Primandani Arsi[2], Khairunnisak Nur Isnaini*[3]

[1]Information Technology, Computer Science Faculty, Universitas Amikom Purwokerto, Indonesia
[2,3]Informatics, Computer Science Faculty, Universitas Jenderal Soedirman, Indonesia
Email: [1]21sa3083@mhs.amikompurwokerto.ac.id, [2]ukhti.prima@amikompurwokerto.ac.id, [3]nisak@amikompurwokerto.ac.id

(Article received: December 4, 2024; Revision: January 7, 2025; published: February 20, 2025)

Abstract

Denial of Service (DoS) attacks are a major threat to network security, characterized by overwhelming system resources with illegitimate requests. Such attacks can disrupt critical services and cause substantial financial losses. However, there is still a need for a more efficient model to detect DoS attack with high accuracy. The aim of this research is to determine the impact of hyperparameter tuning on the four algorithms to identify the best algorithm for detecting DoS network attacks. The research method involves data preprocessing, feature selection, encoding, balancing using SMOTE (Synthetic Minority Over-Sampling Techinuque) and evaluation using confusion matrix. This research use the NSL-KDD dataset because it is relevant for DoS attack detection and flexible for testing various classification algorithms and utilizing hyperparameter tuning. This study evaluates the effectiveness hyperparameter tuning on several machine learning alghorithms namely Random Forest, Linear Discriminant Analysis (LDA), Logistic Regression and Naïve Bayes in detecting DoS attacks. Results indicate that Random Forest achieves highest accuracy (99,97%) and robust performance across all metrics, demonstrating superior generalization and precision. LDA, Logistic Regression and Naïve Bayes also performed well but fell short of Random Forest in handling complex patterns in the dataset. The utilization of hyperparameter tuning can improve the accuracy, consistency and efficiency of the algorithm so as to optimize the combination of various parameters in detecting DoS attacks. The findings provide valuable insights into selecting suitable algorithms for future implementations in cybersecurity systems.

Keywords: *Denial of Service, Hyperparameter Tuning, Intrusion Detection, Machine Learning, Network Security and NSL-KDD Dataset.*

PENGARUH HYPERPARAMETER TUNING PADA ALGORITMA RANDOM FOREST, LDA, LOGISTIC REGRESSION DAN NAIVE BAYES UNTUK MENDETEKSI SERANGAN JARINGAN DOS

Abstrak

Serangan Denial of Service (DoS) merupakan ancaman utama bagi keamanan jaringan, ditandai dengan melimpahnya sumber daya sistem dengan permintaan yang tidak sah. Serangan semacam itu dapat mengganggu layanan penting dan menyebabkan kerugian finansial yang besar. Namun, masih terdapat kebutuhan akan model yang lebih efisien dalam mendeteksi seranggan DoS dengan akurasi tinggi. Tujuan dari penelitian ini adalah untuk mengetahui dampak hyperparameter tuning pada keempat algoritma tersebut untuk mengidentifikasi algoritma terbaik dalam mendeteksi serangan jaringan DoS. Metode penelitian ini melibatkan prapemrosesan data, pemilihan fitur, pengkodean label, penyeimbangan menggunakan SMOTE (*Synthetic Minority Over-Sampling Techinuque*) serta evaluasi menggunakan confusion matrix. Penelitian ini menggunakan dataset NSL-KDD karena relevan untuk deteksi serangan DoS dan fleksibel untuk pengujian berbagai algoritma klasifikasi serta pemanfaatan hyperparameter tuning. Hasil penelitian ini mengevaluasi efektivitas hyperparameter tuning pada beberapa algoritma machine learning yaitu Random Forest, Linear Discriminant Analysis (LDA), Logistic Regression dan Naïve Bayes dalam mendeteksi serangan DoS. Hasilnya menunjukkan bahwa Random Forest mencapai akurasi tertinggi (99,97%) dan kinerja yang kuat di semua metrik, menunjukkan generalisasi dan presisi yang unggul. Linear Discriminant Analysis, Logistic Regression dan Naïve Bayes juga memiliki kinerja yang baik tetapi tidak sebaik Random Forest dalam menangani pola yang kompleks dalam dataset. Pemanfaatan hyperparameter tuning dapat meningkatkan akurasi, konsistensi dan efisiensi algoritma sehingga dapat

mengoptimalkan dari kombinasi berbagai parameter dalam mendeteksi serangan DoS. Temuan ini memberikan wawasan yang penting dalam memilih algoritma yang sesuai untuk implementasi di masa depan dalam sistem keamanan siber.

Kata kunci: *Denial of Service, Deteksi Intrusi, Hyperparameter Tuning, Keamanan Jaringan, Machine Learning, NSL-KDD.*

1. PENDAHULUAN

DoS atau Denial of Service adalah serangan siber yang bertujuan untuk melumpuhkan jaringan dengan mengirimkan sejumlah besar permintaan palsu. Serangan ini biasanya dilakukan dengan menggunakan botnet, yaitu jaringan komputer yang telah dikendalikan oleh penyerang. Botnet kemudian digunakan untuk mengirimkan permintaan palsu ke target, sehingga dapat mengganggu kinerja jaringan dan membuatnya tidak dapat berfungsi dengan baik[1].

Serangan DoS berupaya menguras sumber daya komputasi pada host atau jaringan, yang membuatnya tidak dapat diakses oleh pengguna yang sah atau bahkan mengakibatkan penurunan performa sistem komputer[2]. Contoh yang termasuk ke dalam kategori DoS adalah eksploitasi perangkat lunak (software exploit) dan Flooding Attacks[3]. Pada software exploit, penyerang menggunakan celah keamanan yang ada pada server untuk menghentikan layanan atau memperlambat performa server secara drastis. Sementara itu, Flooding Attacks menimbulkan gangguan dengan menguras sumber daya sistem, dimana penyerang mengirimkan permintaan besar yang tidak valid, menyebabkan server kelebihan beban dan tidak mampu menangani permintaan yang sah[4]. Serangan TCP SYN flooding memanfaatkan kelemahan three-way handshake TCP, ketika penyerang mengirim banyak pesan SYN palsu ke server, server merespon dengan SYN/ACK tetapi klien palsu tidak membalas, menyebabkan server kehabisan sumber daya koneksi setengah terbuka, sehingga tidak bisa menerima koneksi baru. Target serangan ini meliputi server web, FTP, email, dan layanan TCP lainnya[5].

Serangan DoS menyebabkan jaringan dan sistem tidak dapat diakses oleh pengguna yang sah, mengganggu layanan dan menghentikan aktivitas bisnis yang berdampak pada produktivitas serta pendapatan organisasi. Target utama serangan ini seperti web dan gateway default seringkali mengalami gangguan layanan luas dan hilangnya akses ke sumber daya penting[6]. Serangan DoS juga menimbulkan biaya signifikan terkait waktu henti, pemulihan dan kehilangan pelanggan Dampak psikologis pada pengguna juga mencakup hilangnya kepercayaan pelanggan, yang dapat mengakibatkan kerusakan reputasi jangka panjang. Serangan ini sering digunakan sebagai pengalih perhatian untuk serangan sekunder seperti pencurian data sehingga dapat memperburuk dampak keseluruhan [7].

Dalam beberapa tahun terakhir, frekuensi laporan mengenai serangan siber terutama serangan Denial of Service (DoS) semakin meningkat, dengan semakin banyak situs internet yang menjadi target serangan. Serangan DoS tidak hanya menyerang situs-situs kecil, tetapi juga situs penting seperti platform bisnis, layanan pemerintahan, hingga infrastruktur digital yang kritis[5]. Selama periode ini, serangan DoS berpotensi menimbulkan kerugian finansial dalam jumlah besar, mulai dari jutaan hingga miliaran dolar[8]. Dalam survei kejahatan siber dan keamanan yang dilaporkan oleh Crime Scene Investigation (CSI) dan Federal Bureau of Investigation (FBI), 42% peserta mengidentifikasi serangan DoS sebagai masalah utama. Kerugian finansial akibat serangan DoS menempati urutan kedua sebagai penyebab terbesar hilangnya pendapatan, setelah pencurian informasi yang bersifat kepemilikan[9].

Mengingat dampak signifikan yang ditimbulkan oleh serangan Denial of Service (DoS), implementasi sistem deteksi dini serangan siber menjadi sangat krusial. Melalui deteksi dini, tim keamanan siber dapat segera mengambil tindakan untuk memblokir serangan, mengisolasi sistem yang terinfeksi, dan memulihkan layanan yang terganggu. Salah satu pendekatan yang berpotensi adalah pemanfaatan teknologi machine learning. Sebagaimana penelitian yang dilakukan oleh Brunel Rolack Kikissagbe dkk, mereka menggunakan teknik dan objek yang sama yaitu penggunaan teknologi machine learning untuk mendeteksi serangan DoS[10]. Data sistem keamanan dimanfaatkan pada proses pembuatan model machine learning akan menghasilkan pola-pola yang terbentuk sehingga dapat mengidentifikasi serangan yang belum pernah terdeteksi sebelumnya. Selain itu, sistem juga dapat secara otomatis menyesuaikan strategi keamanan untuk menghadapi ancaman yang muncul[11]. Sehubungan dengan hal tersebut, pada penelitian ini akan dilakukan perbandingan beberapa algoritma machine learning yaitu Random Forest, Linear Discriminant Analysis, Logistic Regression dan Naïve Bayes untuk mendeteksi serangan jaringan DoS.

Algoritma Random Forest memiliki kemampuan generalisasi yang baik dan tahan terhadap overfitting. Melalui pemanfaatan prinsip ensemble learning, Random Forest mampu mengatasi kompleksitas data dan kemampuan untuk mengidentifikasi pola yang kompleks dengan cara menggabungkan hasil prediksi dari setiap pohon

sehingga dapat menjadi pilihan efektif untuk mendeteksi serangan DoS[12]. Algoritma Linear Discriminant Analysis (LDA) cocok untuk mengkategorikan data ke dalam kelompok yang sudah dikenal berdasarkan kombinasi fitur. LDA efektif dalam membedakan data serangan dan non-serangan, memanfaatkan beberapa fitur untuk meningkatkan akurasi klasifikasi [13]. Selanjutnya, algoritma Logistic Regression umum digunakan untuk menghitung probabilitas dalam memprediksi hasil berdasarkan variabel input. Metode ini menghasilkan output biner, yaitu benar dan salah. Dalam proses klasifikasi, Logistic Regression memberikan nilai pada setiap variabel prediktor yang mengukur kontribusi independen masing-masing prediktor terhadap perubahan variabel dependen[14]. Algoritma Naïve Bayes merupakan metode statistik yang juga umum digunakan untuk menyelesaikan masalah klasifikasi berdasarkan Teorema Bayes. Teknik ini dikenal karena kesederhanaan dan efisiensinya, terutama dalam menangani dataset ukuran besar. Model ini mengasumsikan bahwa fitur-fitur dalam data bersifat independen satu sama lain sehingga mempermudah proses klasifikasi[15].

Guna meningkatkan akurasi atau ketepatan dalam memprediksi ancaman serangan jaringan DoS, dibutuhkan metode tambahan yaitu hyperparameter tuning. Hyperparameter tuning bertujuan untuk menentukan nilai optimal yang melibatkan pencarian kombinasi dari berbagai parameter model machine learning[16]. Hyperparameter tuning diperlukan untuk meningkatkan kinerja model machine learning dalam mendeteksi serangan siber. Penelitian yang dilakukan oleh Amos Oyetero dkk, model Random Forest mencapai akurasi 100% tanpa hyperparameter tuning, sementara Neural Network memerlukan tuning untuk mengoptimalkan kinerjanya yang menunjukkan bahwa tiap algoritma mempunyai perbedaan respon terhadap hyperparameter tuning[17]**.** Tuning yang optimal dapat secara signifikan meningkatkan akurasi model, meskipun beberapa algoritma tetap menunjukkan kinerja yang baik dengan pengaturan bawaan [18]. Metode tuning seperti random search dan Bayesian optimization juga memiliki pengaruh besar terhadap efisiensi dan efektivitas proses penyetelan [19]. Namun, analisis sistematis terhadap publikasi machine learning mengungkapkan bahwa banyak pengaturan hyperparameter yang tidak dioptimalkan dengan baik, sehingga mengurangi reproduktivitas hasil penelitian [20]. Seperti yang dilakukan oleh Azar Abid Salih, dkk menyebutkan algoritma seperti Logistic Regression dan Naive Bayes membutuhkan hyperparameter tuning mendalam untuk mendapatkan hasil serupa[21]. Di sisi lain, fokus yang berlebihan pada hyperparameter tuning dapat menyebabkan *overfitting*, di mana model hanya unggul pada data pelatihan tetapi berkinerja buruk pada data baru. Oleh karena itu, penelitian lebih lanjut disarankan untuk mengidentifikasi metode penyetelan yang paling efektif untuk berbagai algoritma guna mencapai keseimbangan antara optimasi hyperparameter dan generalisasi model[19].

Hyperparameter tuning memerlukan waktu dan smber daya komputasi yang cukup besar, terutama dengan teknik seperti grid search atau random search yang mengharuskan evaluasi berbagai kombinasi parameter pada dataset yang besar. Hyperparameter tuning juga dapat menimbulkan risiko overfitting jika dilakukan berlebihan pada data validasi. Pemrosesan hyperparameter tuning tidak selalu menghasilkan peningkatan kinerja pada semua algoritma dan beberapa model bahkan tidak memperoleh manfaat yang signifikan dari tuning yang bisa mengakibatkan pemborosan waktu dan sumber daya [21][22]. Kurangnya pembahasan mendalam mengenai efek tuning pada beberapa algoritma sekaligus mengarah pada ketidakjelasan tentang bagaimana masing-masing algoritma merespon tuning, mengingat setiap algoritma memiliki karakteristik berbeda yang memengaruhi efektivitas tuning dalam meningkatkan kinerja model deteksi serangan siber.

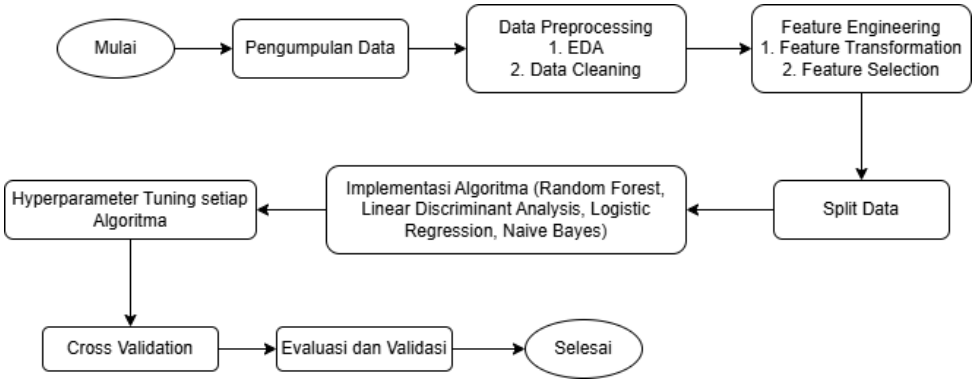
Penelitian sebelumnya oleh Mehdi Houichi dkk (2024) menggunakan berbagai algoritma, termasuk Random Forest yang mencapai akurasi 100%, KNN dengan 99,7% dan Logistic Regression dengan 89,4%[23]. Sementara itu, penelitian oleh Monika Vishwakarma (2023) menunjukkan bahwa algoritma Naïve Bayes mencapai akurasi 97% pada dataset NSL-KDD, 86,9% pada dataset UNSW-NB15 dan 98,59% pada dataset CIC-IDS2017[24]. Penelitian oleh T.Saranya dkk. (2020) menemukan bahwa Random Forest memiliki akurasi tertinggi sebesar 99,81% diikuti oleh LDA dengan 98,1%[25]. Berbeda dengan penelitian sebelumnya, penelitian ini akan berfokus pada deteksi serangan DoS dengan menggabungkan algoritma Random Forest, Linear Discriminant Analysis, Logistic Regression dan Naïve Bayes untuk mengevaluasi efektivitas masing-masing algoritma secara mendalam.

Penelitian ini berkontribusi dalam mengembangkan model adaptif berbasis machine learning untuk mendeteksi serangan DoS melalui analisis performa empat algoritma machine learning, yaitu Random Forest, Linear Discriminant Analysis(LDA), Logistic Regression dan Naïve Bayes dengan pendekatan berbasis hyperparameter tuning. Selain itu, penelitian ini membandingkan efektivitas algoritma-algoritma tersebut menggunakan dataset NSL-KDD, yang relevan dengan serangan DoS dan menawarkan pendekatan untuk mengurangi kesalahan deteksi serta meningkatkan efisiensi dalam pengolahan dataset kompleks. Secara keseluruhan, penelitian ini berkontribusi dalam merancang sistem deteksi serangan DoS yang lebih akurat, efisien, dan adaptif

melalui penggunaan model machine learning yang dioptimalkan. Fokus utama penelitian ini adalah mengevaluasi dampak dari pengoptimalan parameter model terhadap akurasi masing-masing algoritma dalam mendeteksi serangan DoS, yang merupakan ancaman siber signifikan dengan dampak finansial dan operasional yang besar. Melalui perbandingan ini, penelitian bertujuan untuk mengidentifikasi algoritma yang paling efektif dalam mendeteksi serangan jaringan DoS. Selain itu, hasil penelitian ini diharapkan dapat memberikan panduan praktis bagi pengembangan sistem keamanan adaptif yang lebih akurat dalam mendeteksi ancaman siber, sehingga mendukung upaya mitigasi serangan jaringan secara dini.

2. METODE PENELITIAN

Penelitian ini menjelaskan metode yang digunakan secara rinci, termasuk tahapan-tahapan atau langkah-langkah yang ditampilkan pada gambar 2.1.



Gambar 2.1 Metode Penelitian

Gambar 2.1 menampilkan mengenai tahapan-tahapan yang akan dilakukan yaitu Pengumpulan Data, Data Preprocessing, Feature Engineering, Split Data, Implementasi Algoritma, Hyperparameter Tuning, Cross Validation serta Evaluasi dan Validasi.

2.1. Pengumpulan Data

Dalam penelitian ini dilakukan pengumpulan data sekunder untuk memperoleh informasi penting bagi penelitian yang sedang dijalankan. Dataset yang digunakan adalah NSL-KDD yang dapat diakses secara publik melalui Kaggle. Dataset ini berisi mengenai berbagai jenis serangan seperti DoS, Probe, U2R, dan R2L, serta traffic jaringan yang normal. Secara keseluruhan, total entri pada dataset NSL-KDD yaitu 125.973 baris dengan 43 kolom[26].

2.2. Data Preprocessing

Data Preprocessing merupakan tahapan dalam pembuatan model machine learning yang bertujuan untuk membersihkan dan menyusun ulang data mentah agar siap digunakan dalam pelatihan model[23]. Tahapan data preprocessing meliputi EDA (Exploratory Data Analysis) dan Data Cleaning.

2.2.1. EDA (*Exploratory Data Analysis*)

Tahap Exploratory Data Analysis (EDA) bertujuan untuk memahami karakteristik dataset, mengidentifikasi pola dan distribusi data. Analisis statistik dan visualisasi pada fitur numerikal dan kategorikal juga dilakukan untuk mendeteksi outlier [22].

2.2.2. Data Cleaning

Setelah menyelesaikan tahap EDA, langkah selanjutnya adalah pembersihan data untuk memastikan kualitas data yang optimal sebelum tahap pemodelan. Proses pembersihan data ini meliputi penanganan missing value, duplikat data, outlier serta data yang tidak seimbang[27].

2.3. Feature Engineering

Feature Engineering bertujuan untuk mengubah data mentah menjadi format yang lebih sesuai untuk pemodelan prediktif sehingga dapat meningkatkan performa model machine learning[28][28]. Tahapan feature engineering meliputi feature encoding, imbalance label handling dan feature selection.

2.3.1. Feature Transformation

Feature transformation mencakup normalisasi dan encoding untuk mengubah data kategorik menjadi format yang sesuai pada pemodelan machine learning[29]. Teknik One-hot encoding digunakan untuk mengubah kategori menjadi kolom biner, sedangkan label encoding mengganti kategori dengan bilangan integer[30]. Selanjutnya dilakukan penyeimbangan data menggunakan metode SMOTE (*Synthetic Minority Over-Sampling Techinuqe*). Metode ini efektif menangani ketidakseimbangan kelas dengan menghasilkan sampel sintetis yang lebih representatif, mengurangi risiko overfitting[31]

[32]. Metode ini meningkatkan akurasi dan skor F1 model, seperti Random Forest, dibandingkan metode lain seperti ADASYN[33]. SMOTE juga fleksibel untuk berbagai dataset [34].

2.3.2. Feature Selection

Feature selection berperan dalam pengutamaan fitur yang paling relevan dan terkait dengan target dengan tujuan untuk menyederhanakan model dan mengurangi kompleksitasnya[35]. Feature selection dilakukan dengan menganalisis korelasi masing-masing fitur sehingga membantu memilih fitur yang memiliki korelasi kuat dan menghilangkan fitur dengan korelasi lemah sehingga mencegah overfitting atau redundansi[36].

2.4. Split Data

Split data membagi dataset menjadi data pelatihan, validasi, dan pengujian untuk memastikan model dapat mengeneralisasi dengan baik pada data baru[37][38]. Sementara itu, 20% data validation digunakan untuk menguji kinerja model, memberikan gambaran yang jelas mengenai seberapa baik model dapat menggeneralisasi ke data baru[39]. Pembagian data pelatihan menjadi data validasi memungkinkan penyesuaian hyperparameter dan pemilihan model tanpa mengorbankan data pengujian, sehingga hasil evaluasi model menjadi lebih akurat dan tidak bias[40]. Pembagian dataset menjadi train, validation dan test mencegah overfitting, memungkinkan tuning model dan pemilihan model terbaik berdasarkan kinerja pada data yang tidak digunakan saat pelatihan[41].

2.5. Implementasi Algoritma

Model algoritma machine learning yang digunakan pada penelitian ini yaitu algoritma Random Forest, Logistic Regression, Linear Discriminant Analysis dan Naïve Bayes.

2.5.1. Random Forest

Random Forest adalah algoritma klasifikasi berbasis pohon keputusan yang mengurangi overfitting dengan menggabungkan banyak pohon keputusan[42]. Algoritma Random Forest menggunakan teknik bagging dengan sampel acak dan random subsetting untuk memilih fitur secara acak, mengurangi korelasi antar pohon, dan meningkatkan variasi model[43]. Proses ini menghasilkan pohon keputusan independen, dengan prediksi digabungkan menggunakan mayoritas suara untuk klasifikasi atau rata-rata untuk regresi[44].

2.5.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) adalah algoritma pemrosesan linear yang bertujuan untuk mengklasifikasikan titik-titik dari kelas-kelas yang berbeda ke sebuah garis lurus. Tujuan utamanya adalah untuk memaksimalkan jarak antara titik-titik dari kelas yang sama sekaligus meminimalkan jarak antara titik-titik dari kelas yang berbeda dalam ruang yang diproyeksikan[45]. Dalam skenario sederhana yang melibatkan dua kelas, biasanya dilabeli sebagai C1 dan C2, fungsi diskriminan yang khas dilambangkan sebagai h1, digunakan untuk mengklasifikasikan titik data. Fungsi h1 untuk setiap titik data, LDA menggunakan fungsi diskriminan untuk memisahkan kelas data seperti yang dinyatakan dalam Persamaan 1.

$$h_1(x) = (g_1 - g_2)^t V^{-1} x \quad (1)$$

Pada Persamaan 2, g_1 dan g_2 adalah centroid kelas C1 dan C2, sementara V adalah matriks kovarian. Fungsi diskriminan h1 dihitung menggunakan data pelatihan untuk mengklasifikasikan data baru berdasarkan nilai x0, dengan keputusan C1 atau C2 tergantung rentang h1(x0), sebagaimana dirumuskan dalam Persamaan 2.

$$\begin{cases} if\ h_1(x_0) \geq 0\ then\ x_0 \in C_1 \\ else\ x_0 \in C_2 \end{cases} \quad (2)$$

Pendekatan ini menghasilkan klasifikasi efisien dan akurat dengan menyederhanakan masalah menjadi pemisahan linear antara dua kelas. [46]. LDA memproyeksikan sampel ke garis tertentu dan mengklasifikasikan sampel baru berdasarkan posisinya relatif terhadap proyeksi sampel pelatihan, menjadikannya alat efektif untuk klasifikasi[47].

2.5.3. Logistic Regression

Logistic Regression adalah metode klasifikasi biner yang memprediksi probabilitas antara 0 hingga 1, menggunakan fungsi logistik untuk mengubah hasil regresi linear menjadi probabilitas[45]. Persamaan 3 merepresentasikan model Logistic Regression secara umum ketika terdapat p variabel independen.

$$\boldsymbol{\pi(X) = E(Y|X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}} \quad (3)$$

π(X): Probabilitas prediksi bahwa variabel dependen Y bernilai 1 berdasarkan variabel independen X (0-1).

E(Y|X): Nilai harapan Y untuk X, setara dengan π(X).

β_0: Intercept yang menunjukkan log-odds hasil ketika semua variabel independen nol.

β_1, β_2, ..., β_p: Koefisien variabel independen X_1, X_2, ..., X_p, yang menunjukkan perubahan log-odds hasil untuk setiap kenaikan satu unit variabel, dengan variabel lain tetap [48].

Guna meningkatkan interpretasi, model pada Persamaan 3 ditransformasi menggunakan logit, menghasilkan Persamaan 4.

$$\boldsymbol{g(X) = \log\left(\frac{\pi(X)}{1-\pi(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p} \quad (4)$$

Parameter β dalam Logistic Regression diestimasi menggunakan Maximum Likelihood

Estimation (MLE) yang memaksimalkan fungsi likelihood untuk menjelaskan data secara optimal[47]. Proses pemodelan mencakup estimasi parameter, uji signifikansi keseluruhan dengan Likelihood Ratio Test, uji signifikansi individu dengan Uji Wald, pengecualian variabel tidak signifikan, dan evaluasi kesesuaian model[49].

2.5.4. Naïve Bayes

Naïve Bayes adalah metode klasifikasi berbasis Teorema Bayes yang efisien dan cepat, dengan asumsi fitur independen kondisional terhadap kelas, metode ini efektif, terutama dengan teknik estimasi kepadatan kernel[15]. Persamaan 5 merupakan aturan algoritma Naïve Bayes untuk melakukan keputusan klasifikasi.

$$y = \frac{argmax}{P\in(1,2,..,p)}\ p(C_p)\prod_{i=1}^{n} p(x_i|C_p) \quad (5)$$

y: Label kelas prediksi untuk data masukan, ditentukan dengan memaksimalkan probabilitas posterior.
C_p: Label kelas yang dievaluasi algoritma (1, 2, ..., p).

$P(C_p)$: Probabilitas prior kelas C_p sebelum mengamati fitur.

x_i: Fitur individual data dengan i dari 1 hingga n.

$P(x_i \mid C_p)$: Probabilitas fitur x_i muncul jika data termasuk kelas C_p [50][51].

Dalam model Naïve Bayes, $\boldsymbol{y}$ adalah label prediksi (kelas) untuk data tertentu. Proses prediksi ini didasarkan pada prinsip $\boldsymbol{argmax}$ yang berarti kelas $\boldsymbol{C_p}$ akan dipilih karena menghasilkan probabilitas tertinggi. Probabilitas ini dihitung menggunakan rumus $\boldsymbol{p(C_p)\prod_{i=1}^{n} p(x_i|C_p)}$, yang merupakan kombinasi dari probabilias prior $\boldsymbol{p(C_p)}$ dan probabilitas bersyarat dari setiap fitur $\boldsymbol{x_i}$ terhadap kelas $\boldsymbol{C_p}$. Secara sederhana, Naïve Bayes memilih kelas dengan nilai posterior tertinggi, mempertimbangkan semua fitur untuk memilih kelas dengan probabilitas terbesar[52].

2.6. Hyperparameter Tuning

Setelah implementasi algoritma machine learning, langkah berikutnya adalah melakukan hyperparameter tuning untuk menemukan kombinasi nilai hyperparameter yang optimal, sehingga meningkatkan akurasi model[36]. Teknik hyperparameter tuning yang digunakan pada penelitian ini yaitu Grid Search CV. Grid Search (GS) adalah teknik hyperparameter tuning yang mengevaluasi semua kombinasi nilai hyperparameter dalam grid menggunakan data validasi untuk menemukan konfigurasi terbaik[53]. Dalam Grid Search Cross Validation (CV), proses ini dilakukan secara sistematis dengan cross validation untuk memastikan eksplorasi menyeluruh terhadap ruang hyperparameter[27].

2.7. Cross Validation

Cross-validation membagi data menjadi beberapa bagian (k-fold) dan bergantian menggunakan tiap bagian sebagai data uji dan latih. [39]. Tujuannya adalah memberikan validasi akurasi yang optimal melalui rotasi antara data uji dan latih [41]. Dalam penelitian ini, setiap algoritma diuji dengan 10-fold cross-validation, membagi dataset menjadi sepuluh bagian untuk evaluasi kinerja yang lebih akurat.

2.8. Evaluasi dan Validasi

Setelah cross-validation, evaluasi dilakukan untuk memastikan model efektif pada data pelatihan, validasi, dan pengujian. Confusion matrix digunakan untuk menilai prediksi benar dan salah, sementara metrik seperti akurasi, presisi, recall, dan F1-Score dibandingkan antar algoritma untuk meningkatkan kinerja model.

2.8.1. Akurasi

Akurasi mengukur seberapa baik model mengklasifikasikan data dengan benar dihitung dari perbandingan prediksi tepat terhadap total prediksi[14]. Formula untuk menghitung akurasi terdapat pada persamaan 6.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

2.8.2 Presisi

Presisi mengukur ketepatan model dalam prediksi benar dari seluruh prediksi kelas tertentu, penting untuk meminimalkan False Positive[14]. Formula untuk menghitung presisi terdapat pada persamaan 7.

$$Precision = \frac{TP}{FP+TP} \quad (7)$$

2.8.3. Recall

Recall mengukur sejauh mana model mengenali dan mengklasifikasikan sampel positif dengan benar[14]. Formula untuk menghitung recall terdapat pada persamaan 8.

$$Recall = \frac{TP}{FN+TP} \quad (8)$$

2.8.4. F1-Score

F1-score merupakan nilai rata-rata harmonis yang dihitung dari kombinasi antara presisi dan recall[14]. Formula untuk menghitung F1-score terdapat pada persamaan 9.

$$F1 - Score = 2 * \frac{Precision+recall}{Precision+recall} \quad (9)$$

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Data

NSL-KDD adalah dataset yang dirancang untuk mengatasi kelemahan dataset KDD'99 dan

digunakan sebagai benchmark deteksi serangan jaringan. Dataset ini memiliki 43 atribut dan 125.973 records yang mencakup fitur seperti durasi koneksi, jenis protokol, dan jumlah byte yang ditransfer. Label target pada kolom "outcome" menunjukkan koneksi sebagai normal atau serangan.

3.2. Data Preprocessing

Setelah memuat data ke Google Colab, langkah berikutnya adalah preprocessing, dimulai dengan EDA untuk memahami pola dan mendeteksi masalah, lalu data cleaning untuk menangani missing value, duplikasi, dan inkonsistensi lainnya.

3.2.1. Exploratory Data Analysis (EDA)

Langkah awal sebelum eksplorasi data adalah memisahkan fitur berdasarkan tipe data, yaitu numerikal dan nominal, untuk memudahkan analisis. Fitur nominal mencakup atribut seperti 'protocol_type', 'service', 'flag', 'land', 'logged_in', 'is_host_login', 'is_guest_login'. Sementara itu, fitur numerikal meliputi 'duration', 'src_bytes', 'dst_bytes', 'wrong_fragment' dan lainnya. Setelah dilakukan pemisahan fitur berdasarkan tipe data, tahap selanjutnya adalah menganalisa fitur numerikal dan fitur nominal.

a. Analisa Fitur Numerikal

Analisis fitur numerik meliputi statistik deskriptif, distribusi data, dan kecondongan (skewness) serta outlier dideteksi menggunakan Z-score (threshold ±3). Dari 35 fitur, 26 memiliki distribusi sangat condong ke kiri, dengan data terkonsentrasi pada nilai tinggi. Distribusi ini membuat rata-rata lebih rendah dari median karena ekor panjang di sisi kiri.

Analisis distribusi fitur numerik menunjukkan 3 fitur berdistribusi moderately skewed to the left, 1 fitur highly skewed to the right dan 2 moderately skewed to the right. Tiga fitur yaitu num_outbound_cmds, dst_host_srv_count, dst_host_srv_diff_host_rate memiliki distribusi approximately symmetric. Analisis Z-score mendeteksi outlier pada 21 dari 43 fitur, dengan srv_diff_host_rate memiliki outlier terbanyak yaitu 8.144 dan dst_bytes paling sedikit 3, sementara 14 fitur tidak memiliki outlier.

b. Analisa Fitur Nominal

Analisis fitur nominal dilakukan dengan menampilkan grafik distribusi data dan informasi mengenai nilai yang paling sering muncul serta tingkat keseimbangan distribusi. Tingkat keseimbangan ditentukan berdasarkan proporsi nilai dengan frekuensi tertinggi menggunakan fungsi `check_imbalance` dan `value_counts(normalize=True).max()`. Jika proporsinya > 0,8, kolom dianggap "highly imbalanced" (sangat tidak seimbang); antara 0,6–0,8, "moderately imbalanced" (cukup tidak seimbang); dan < 0,6, "balanced" (seimbang). Keseimbangan kolom diidentifikasi berdasarkan distribusi frekuensi nilai.

3.2.2. Data Cleaning

a. Missing Value Handling

Setelah dilakukan pengecekan terhadap keberadaan nilai yang hilang pada setiap fitur dengan menggunakan fungsi isna().sum(), tidak ditemukan adanya missing value dalam dataset. Oleh karena itu, tidak diperlukan tindakan lanjutan untuk menangani missing value.

b. Duplicate Data Handling

Pemeriksaan data duplikat menggunakan `duplicated().sum()` menunjukkan tidak ada baris duplikat dalam dataset, sehingga tidak diperlukan penanganan tambahan.

c. Outlier Handling

Penanganan outlier dilakukan dengan menghapus 37.831 data yang menyimpang, mengurangi jumlah baris dari 125.973 menjadi 88.142. Langkah ini memastikan kualitas dataset dan mencegah pengaruh nilai ekstrem pada analisis.

3.3. Feature Engineering

3.3.1. Feature Transformation

Fitur target outcome pada dataset NSL-KDD memiliki 23 kategori yang dikelompokkan menjadi tiga yaitu dos (34.740 data), normal (51.831 data) dan non-dos (1.571 data). Label ini dikonversi ke format numerik yaitu dos (1), non-dos (2) dan normal (3). Fitur kategorik lainnya diubah menjadi numerik menggunakan One-Hot Encoding. Setelah semua fitur dikonversi, ketidakseimbangan data diatasi dengan metode SMOTE (Synthetic Minority Over-sampling Technique) untuk mencegah bias prediksi model.

Sebelum dilakukan penyeimbangan, kategori 'normal' memiliki jumlah data sebanyak 51.831, sedangkan kelas 'dos' mempunyai jumlah data 34.740, kemudian 'non-dos' memiliki jumlah data 1.571. Setelah penerapan SMOTE, semua kategori memiliki jumlah data yang sama yaitu 51.831 baris.

3.3.2. Feature Selection

Analisis korelasi menggunakan `corr()` pada `df_no_outliers_smote` mengidentifikasi fitur dengan korelasi absolut ≥ 0.1 terhadap target outcome. Dari 121 fitur, 30 fitur memiliki korelasi signifikan baik positif maupun negatif dan dipilih untuk pemrosesan lebih lanjut.

3.4. Split Data

3.4.1. Split Dataset (Train & Test)

Setelah feature engineering, dataset dipisah menjadi variabel X (fitur) dan y (target outcome). Data kemudian dibagi menjadi training (X_train, y_train) dan testing (X_test, y_test) dengan proporsi 80:20 menggunakan parameter `test_size=0.2`. Data pelatihan digunakan untuk melatih model, sedangkan data pengujian untuk evaluasi kinerja.

3.4.2. Split Train (Train & Validation)

Data pelatihan (X_train, y_train) dibagi menjadi training (80%) dan validation (20%). Hasilnya, training set memiliki 99.515 sampel, validation set 24.879 sampel, dan test set 31.099

sampel, masing-masing dengan 30 fitur dan target yang sama. Selanjutnya, standarisasi dilakukan untuk menyamakan skala fitur (mean 0, standar deviasi 1).

3.5. Pembuatan Model

3.5.1. Pembuatan Model Random Forest

Model Random Forest dilatih menggunakan X_train dan y_train, menghasilkan akurasi 100% pada data pelatihan. Berdasarkan confusion matrix, semua sampel kelas (1, 2, 3) diklasifikasi dengan benar tanpa kesalahan, didukung nilai presisi, recall dan F1-score sempurna (1.00) untuk semua kelas. Prediksi pada data validasi (y_pred_valid) menghasilkan akurasi 99.9759%, dengan sedikit kesalahan klasifikasi (1-2 sampel). Presisi, recall dan F1-score untuk semua kelas tetap sempurna (1.00). Akurasi model Random Forest pada data pelatihan dan data validasi dapat dilihat pada gambar 3.1.

```
Akurasi pada data training: 100.0000%
Confusion Matrix - Training:
[[33172     0     0]
 [    0 33172     0]
 [    0     0 33171]]

Classification Report - Training:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     33172
           2       1.00      1.00      1.00     33172
           3       1.00      1.00      1.00     33171
```

```
Akurasi pada data validasi: 99.9759%
Confusion Matrix - Validation:
[[8292    0    1]
 [   0 8291    2]
 [   2    1 8290]]

Classification Report - Validation:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      8293
           2       1.00      1.00      1.00      8293
           3       1.00      1.00      1.00      8293
```

Gambar 3.1 Akurasi Model Random Forest Pada Data Train Dan Data Validasi

3.5.2. Pembuatan Model Linear Dsicriminant Analysis

Model LDA mencapai akurasi 97.6998% pada data pelatihan. Evaluasi melalui confusion matrix menunjukkan presisi, recall dan F1-score tinggi, dengan F1-score mendekati 0.98 untuk semua kelas. Kemudian hasil evaluasi pada menunjukkan bahwa model tetap memiliki performa yang kuat dengan akurasi 97,8978. Model menunjukkan presisi yang bervariasi yaitu pada kelas 1 (dos) menunjukkan presisi 98,87%, kelas 2 (non-dos) 96,88% serta kelas 3 (normal) menunjukkan presisi 97,97% dengan recall antara 96,41% hingga 98,98%. Akurasi model Linear Discriminant Analysis pada data pelatihan dan data validasi dapat dilihat pada gambar 3.2.

```
Akurasi pada data training: 97.6998%
Confusion Matrix - Training:
[[32539     9   624]
 [  227 32806   139]
 [  159  1131 31881]]

Classification Report - Training:
              precision    recall  f1-score   support

           1       0.99      0.98      0.98     33172
           2       0.97      0.99      0.98     33172
           3       0.98      0.96      0.97     33171
```

```
Akurasi pada data training: 97.6998%
Confusion Matrix - Training:
[[32539     9   624]
 [  227 32806   139]
 [  159  1131 31881]]

Classification Report - Training:
              precision    recall  f1-score   support

           1       0.99      0.98      0.98     33172
           2       0.97      0.99      0.98     33172
           3       0.98      0.96      0.97     33171
```

Gambar 3.2 Akurasi Model Linear Discriminant Analysis Pada Data Train Dan Data Validasi

3.5.3. Pembuatan Model Logistic Regression

Hasil pelatihan model menunjukkan akurasi 99.4714%, dengan distribusi kesalahan yang minimal menurut confusion matrix. Classification report menunjukkan presisi, recall dan F1-score di atas 99% untuk semua kelas, menandakan kemampuan model dalam membedakan kelas dengan sangat baik.

Evaluasi model pada data validasi menunjukkan akurasi 99.4373%, hampir setara dengan data pelatihan. Confusion matrix menunjukkan kesalahan rendah di semua kelas. Laporan klasifikasi mencatat presisi 99.60% (kelas dos), 99.26% (kelas non-dos), dan 99.45% (kelas normal). Akurasi model Logistic Regression pada data pelatihan dan data validasi dapat dilihat pada gambar 3.3.

```
Akurasi pada data training: 99.4714%
Confusion Matrix - Training:
[[33019     9   144]
 [    4 33101    67]
 [  111   191 32869]]

Classification Report - Training:
              precision    recall  f1-score   support

           1     0.9965    0.9954    0.9960     33172
           2     0.9940    0.9979    0.9959     33172
           3     0.9936    0.9909    0.9923     33171
```

```
Akurasi pada data validasi: 99.4373%
Confusion Matrix - Validasi:
[[8256    6   31]
 [   2 8277   14]
 [  31   56 8206]]

Classification Report - Validasi:
              precision    recall  f1-score   support

           1     0.9960    0.9955    0.9958      8293
           2     0.9926    0.9981    0.9953      8293
           3     0.9945    0.9895    0.9920      8293
```

Gambar 3.3 Akurasi Model Logistic Regression Pada Data Train Dan Data Validasi

3.5.4. Pembuatan Model Naïve Bayes

Hasil pelatihan model pada data pelatihan menunjukkan akurasi 96.9522%. Confusion matrix memperlihatkan distribusi prediksi benar dan salah di ketiga kelas. Laporan klasifikasi mencatat presisi dan recall tinggi: kelas pertama (99% presisi, 98% recall), kelas kedua (93% presisi, 99% recall), kelas ketiga (99% presisi, 94% recall). F1-score di atas 96% menunjukkan kinerja model yang baik dengan kesalahan minimal. Akurasi model Naive Bayes pada data pelatihan dan data validasi dapat dilihat pada gambar 3.4.

Evaluasi pada data validasi menunjukkan akurasi 96.9934%, hampir setara dengan data pelatihan, menunjukkan kinerja yang konsisten. Confusion matrix menunjukkan beberapa kesalahan klasifikasi, namun distribusi prediksi tetap baik di semua kelas. Laporan klasifikasi mencatat presisi 93,31%–99,21% dan recall 94,07%–98,73%, dengan F1-score tinggi (95,95%–98,69%), menunjukkan keseimbangan yang baik antara presisi dan recall.

```
Akurasi pada data training: 96.9522%
Confusion Matrix - Training:
[[32482   515   175]
 [  225 32767   180]
 [   43  1895 31233]]

Classification Report - Training:
              precision    recall  f1-score   support

           1       0.99      0.98      0.99     33172
           2       0.93      0.99      0.96     33172
           3       0.99      0.94      0.96     33171
```

```
Akurasi pada data validasi: 96.9934%
Confusion Matrix - Validasi:
[[8142  105   46]
 [  55 8188   50]
 [  10  482 7801]]

Classification Report - Validasi:
              precision    recall  f1-score   support

           1     0.9921    0.9818    0.9869      8293
           2     0.9331    0.9873    0.9595      8293
           3     0.9878    0.9407    0.9637      8293
```

Gambar 3.4 Akurasi Model Naive Bayes Pada Data Train Dan Data Validasi

3.6. Hyperparameter Tuning

3.6.1. Hyperparameter Tuning Model Random Forest

Setelah pembuatan model Random Forest, langkah selanjutnya adalah implementasi hyperparameter tuning untuk meningkatkan akurasi serta pencarian hyperparameter yang optimal pada model. Hyperparameter tuning pada model Random Forest dilakukan menggunakan Bayesian Optimization dengan library skopt dan metode BayesSearchCV. Hasil optimasi menunjukkan kombinasi parameter terbaik yaitu max_depth 37, max_features 'sqrt', min_samples_leaf 1, min_samples_split 4, dan n_estimators 85, dengan nilai akurasi data validasi terbaik sebesar 1,0 atau sempurna. Konfigurasi ini memberikan kinerja terbaik, memungkinkan model menangkap pola data secara efektif dan menghindari overfitting.

3.6.2. Hyperparameter Tuning Model Linear Discriminant Analysis

Hyperparameter tuning pada model Linear Discriminant Analysis (LDA) dilakukan menggunakan Grid Search CV dengan library sklearn (GridSearchCV). Hasil optimasi menunjukkan kombinasi parameter terbaik: 'solver' diatur ke 'svd' (Singular Value Decomposition) untuk menghitung pemisahan antar kelas, 'store_covariance' diatur ke True untuk menyimpan matriks kovarians selama pelatihan, dan 'tol' terbaik adalah 0,0001 yang menentukan batas toleransi konvergensi algoritma LDA, nilai akurasi terbaik pada data validasi yaitu sebesar 97,8978% yang menunjukkan bahwa hyperparameter tuning pada model ini tidak menghasilkan peningkatan.

3.6.3. Hyperparameter Tuning Model Logistic Regression

Pada algoritma Logistic Regression, hyperparameter tuning dilakukan menggunakan Bayesian Optimization. Hasil optimasi menunjukkan kombinasi parameter terbaik, yaitu C dengan nilai 100,0 yang mengontrol kekuatan regularisasi model. Nilai C yang lebih besar mengurangi efek regularisasi, memungkinkan model lebih fleksibel. Jenis regularisasi yang dipilih adalah 'l2', bentuk regulasi umum untuk menghindari overfitting dengan menambahkan penalti pada bobot model. Nilai akurasi terbaik pada data validasi yaitu

99,48845% menunjukkan peningkatan performa model.

3.6.4. Hyperparameter Tuning Model Naïve Bayes

Pada algoritma Naïve Bayes, hyperparameter tuning yang digunakan yaitu Bayesian Optimization. Hasilnya menunjukkan parameter terbaik yang ditemukan yaitu var-smoothing sebesar $\mathbf{1,82257885981521 \times 10^{-10}}$ yang mengontrol tingkat penambahan varians pada data untuk menghindari pembagian dengan nilai sangat kecil selama perhitungan probabilitas. Nilai var_smoothing yang sangat kecil ini menunjukkan bahwa model Naïve Bayes mengoptimalkan kinerjanya dengan sedikit menambahkan smoothing, sehingga model lebih sensitif terhadap data pelatihan asli dan lebih sedikit melakukan regularisasi. Nilai akurasi terbaik pada data validasi yaitu 97,0084% menunjukkan peningkatan performa model.

3.7. Cross Validation

3.7.1. Cross Validation Model Random Forest

Cross validation dilakukan dengan cv=10 menunjukkan model memiliki performa sangat tinggi pada setiap fold, dengan akurasi sempurna pada fold ketiga dan kedelapan. Rata-rata skor cross-validation sebesar 0,999847858 menunjukkan bahwa model Random Forest memiliki performa yang sangat baik dan stabil dalam memprediksi setiap kelas.

3.7.2. Cross Validation Model Linear Discriminant Analysis

Berdasarkan perlakuan yang sama dengan model Random Forest, hasil cross validation model LDA menunjukkan skor tertinggi mencapai 0,9783963 dan skor terendah sebesar 0,97477894. Skor rata-rata cross validation sebesar 0,9769281406 menunjukkan bahwa model memiliki performa yang stabil dalam memprediksi kelas data.

3.7.3. Cross Validation Model Logistic Regression

Berdasarkan hasil cross-validation, model Logistic Regression menunjukkan nilai akurasi tertinggi pada skor cross-validation mencapai 0,99557833. Rata-rata skor akurasi sebesar 0,994573 menunjukkan performa keseluruhan model.

3.7.4. Cross Validation Model Naïve Bayes

Berdasarkan perlakuan yang sama dengan algoritma sebelumnya, hasil cross validation model Naïve Bayes menunjukkan akurasi tertinggi sebesar 0,97125917 pada fold ke-9. Rata-rata akurasi keseluruhan dari 10 fold adalah 0,969572 yang menunjukkan bahwa model Naïve Bayes yang digunakan memiliki performa yang konsisten dan cukup baik dalam mengenali pola data pelatihan.

3.8. Evaluasi dan Validasi

3.8.1. Pengujian Algoritma Random Forest

Evaluasi performa model pada data pengujian menggunakan fungsi `evaluate_test` menghitung akurasi, presisi, recall, dan F1-score, serta menampilkan classification report dan confusion matrix yang divisualisasikan untuk menunjukkan distribusi prediksi.

Model mencapai akurasi 0,9997, dengan presisi, recall, dan F1-score masing-masing 0,9997, menunjukkan hampir 100% klasifikasi yang benar. Ini menandakan kemampuan model yang sangat baik dalam mendeteksi dan memprediksi data, dengan keseimbangan yang sempurna antara presisi dan recall. Classification Report menunjukkan hasil hampir sempurna untuk setiap kelas. Gambar 3.12 merupakan visualisasi confusion matrix model Random Forest menggunakan heatmap.

Berdasarkan confusion matrix, model menunjukkan performa yang sangat baik pada semua kelas. Pada kelas 1, terdapat 10.362 True Positive, 4 False Negative, 2 False Positive, dan 20.727 True Negative. Pada kelas 2, True Positive sebanyak 10.365, 1 False Negative, dan 20.731 True Negative. Pada kelas 3, True Positive mencapai 10.364, 3 False Negative, 4 False Positive, dan 20.725 True Negative. Hasil ini mengindikasikan bahwa model memiliki prediksi yang sangat akurat dengan sedikit kesalahan pada setiap kelas.

3.8.2. Pengujian Algoritma LDA

Pengujian model dilakukan pada data uji, dengan evaluasi menggunakan classification report dan confusion matrix yang divisualisasikan dengan heatmap. Hasil evaluasi menunjukkan akurasi 97,68%, presisi 97,69%, recall 97,68%, dan F1-score 97,68%, dengan kinerja yang sangat baik dalam memprediksi ketiga kelas.

Confusion matrix menunjukkan hasil prediksi untuk tiga kelas. Pada kelas 1, terdapat 10.172 True Positive (TP), 194 False Negative (FN) yang terdiri dari 9 data yang diprediksi sebagai kelas 2 dan 185 data sebagai kelas 3, serta 134 False Positive (FP), dengan 71 data kelas 2 dan 63 data kelas 3. True Negative (TN) untuk kelas 1 mencapai 20.833. Pada kelas 2, TP tercatat sebanyak 10.244, dengan 122 FN, di mana 71 data diprediksi sebagai kelas 1 dan 341 data sebagai kelas 3, serta TN berjumlah 20.705. Pada kelas 3, terdapat 9.963 TP, 404 FN (63 data diprediksi sebagai kelas 1 dan 341 data sebagai kelas 2), serta 236 FP (185 data kelas 1 dan 51 data kelas 2). TN untuk kelas 3 mencapai 20.532.

3.8.3. Pengujian Algoritma Logistic Regression

Hasil pengujian model menunjukkan akurasi 0,9942, dengan presisi dan recall masing-masing sebesar 0,9942, menunjukkan model sangat akurat dalam mengidentifikasi kelas dan mendeteksi sebagian besar data aktual. F1-score yang mencapai 0,9942 menandakan keseimbangan optimal antara presisi dan recall.

Classification Report menunjukkan bahwa model memiliki kinerja hampir sempurna pada setiap kelas. Kelas 1 memiliki 10.314 prediksi benar dengan sedikit kesalahan. Kelas 2 juga menunjukkan hasil yang sangat baik dengan 10.341 prediksi benar dan minimal kesalahan. Kelas 3 memiliki 10.265 prediksi benar, meskipun ada beberapa kesalahan

prediksi. Confusion Matrix memperlihatkan distribusi True Positive, False Positive dan False Negative. Misalnya, kelas 1 memiliki 47 False Negative (salah diklasifikasikan sebagai kelas 3) dan 5 False Positive (salah diklasifikasikan sebagai kelas 2). True Negative untuk kelas 1 adalah 20.797, untuk kelas 2 adalah 20.662, dan untuk kelas 3 adalah 20.660.

3.8.4. Pengujian Algoritma Naïve Bayes

Pengujian model Naïve Bayes menghasilkan akurasi 97,08%, dengan presisi rata-rata 97,19%, recall 97,08%, dan F1-Score 97,09%, menunjukkan performa yang baik dan seimbang. Berdasarkan classification report, kelas 1 memiliki presisi 0,99, recall 0,98, dan F1-Score 0,99. Kelas 2 mencatat presisi 0,93, recall 0,99, dan F1-Score 0,96, sementara kelas 3 mencapai presisi 0,99, recall 0,94, dan F1-Score 0,97, meskipun ada beberapa kesalahan prediksi.

Confusion Matrix menunjukkan distribusi prediksi model, dengan nilai diagonal utama 10.157, 10.268, dan 9.765 sebagai True Positive. Pada kelas 1, terdapat 146 data salah diprediksi sebagai kelas 2 dan 63 sebagai kelas 3, dengan True Negative 20.748. Kelas 2 memiliki True Negative 19.983, dan kelas 3 mencapai 20.637. Meskipun ada kesalahan pada kelas 3 (583 data salah diklasifikasikan sebagai kelas 2), model menunjukkan performa yang baik dengan nilai tinggi pada diagonal.

Setelah pengujian masing-masing model selesai, dilakukan perbandingan metrik evaluasi seperti akurasi, presisi, recall, dan F1-Score menggunakan tiga set data yaitu data pelatihan, validasi, dan pengujian. Tabel 2 menunjukkan perbandingan evaluasi menggunakan data pelatihan.

Tabel 2. Perbandingan Evaluasi Menggunakan Data Pelatihan

Model Algoritma	Metrik Evaluasi Menggunakan Data Pelatihan			
	Akurasi	Presisi	Recall	F1-Score
Random Forest	100%	100%	100%	100%
Linear Discriminant Analysis	97,69%	98%	97,67%	97,67%
Logistic Regression	99,42%	99,42%	99,42%	99,42%
Naïve Bayes	96,95%	97%	97%	97%

Evaluasi model pada data pelatihan yang terdapat pada Tabel 2 menunjukkan bahwa Random Forest memiliki performa terbaik dengan akurasi, presisi, recall, dan F1-Score mencapai 100%, menandakan deteksi serangan DoS yang sempurna. Logistic Regression mencatatkan 99,42%, diikuti Linear Discriminant Analysis (97,69%), dan Naïve Bayes (96,95%) dengan metrik evaluasi 97%. Evaluasi ini membuktikan Random Forest sebagai model yang paling andal. Tabel 3 menunjukkan perbandingan evaluasi masing-masing model menggunakan data validasi.

Tabel 3. Perbandingan Evaluasi Menggunakan Data Validasi

Model Algoritma	Metrik Evaluasi Menggunakan Data Validasi			
	Akurasi	Presisi	Recall	F1-Score
Random Forest	99,97%	99,98%	99,97%	99,97%
Linear Discriminant Analysis	97,89%	97,91%	97,90%	97,90%
Logistic Regression	99,43%	99,44%	99,44%	99,44%
Naïve Bayes	96,99%	97,10%	96,99%	97%

Tabel 3 merupakan perbandingan metrik evaluasi menggunakan data validasi untuk setiap algoritma. Evaluasi menggunakan data validasi menunjukkan bahwa Random Forest unggul dengan akurasi, presisi, recall, dan F1-score masing-masing 99,97%, menandakan deteksi DoS yang hampir sempurna. Logistic Regression mencatat 99,43%, diikuti Linear Discriminant Analysis dengan 97,89%, dan Naïve Bayes dengan 96,99% serta metrik lainnya sekitar 97%. Evaluasi ini menunjukkan Random Forest menunjukkan keseimbangan terbaik antara presisi dan recall, serta kemampuan meminimalkan deteksi positif palsu dan mendeteksi semua serangan.

Tabel 4. Perbandingan Evaluasi Menggunakan Data Pengujian

Model Algoritma	Metrik Evaluasi Menggunakan Data Pengujian			
	Akurasi	Presisi	Recall	F1-Score
Random Forest	99,97%	99,97%	99,97%	99,97%
Linear Discriminant Analysis	97,68%	97,69%	97,68%	97,68%
Logistic Regression	99,42%	99,42%	99,42%	99,42%
Naïve Bayes	97,07%	97,19%	97,07%	97,08%

Tabel 4 merupakan perbandingan metrik evaluasi menggunakan data pengujian untuk masing-masing algoritma. Evaluasi menggunakan data pengujian menunjukkan bahwa Random Forest memiliki performa terbaik dengan akurasi, presisi, recall, dan F1-score masing-masing 99,97%, menunjukkan kemampuannya mendeteksi serangan DoS secara konsisten. Logistic Regression berada di posisi kedua dengan akurasi 99,42%, diikuti Linear Discriminant Analysis dengan 97,68%, dan Naïve Bayes dengan akurasi 97,07% serta metrik lainnya di kisaran 97%. Evaluasi ini mengukur kinerja model pada data yang belum pernah dilihat selama pelatihan dan validasi.

Hasil penelitian ini berkontribusi dalam memperluas pemahaman mengenai penerapan machine learning untuk mendeteksi serangan DoS dengan meningkatkan kemampuan deteksi dini. Dengan demikian, penanganan lebih lanjut dapat dilakukan secara efektif untuk mengatasi ancaman yang berdampak pada kerahasiaan, integritas dan ketersediaan data sebagai komponen utama CIA Triangle. Model machine learning memiliki kemampuan untuk menganalisis pola lalu lintas jaringan, mengenali anomali, dan beradaptasi terhadap perubahan taktik serangan, sehingga mampu memperkuat keamanan jaringan secara keseluruhan.

4. DISKUSI

Berdasarkan evaluasi pada data pelatihan, data validasi dan data pengujian, Random Forest menunjukkan performa terbaik dalam mendeteksi serangan DoS dengan akurasi, presisi, recall, dan F1-Score sempurna sebesar 100% pada data pelatihan. Setelah hyperparameter tuning, Random Forest mempertahankan keunggulannya dengan akurasi validasi hingga 99,97% dan hasil terbaik mencapai 100%, menunjukkan kemampuan deteksi yang sangat andal bahkan pada data baru. Logistic Regression menempati posisi kedua dengan akurasi validasi terbaik 99,49% setelah tuning, meningkat dari nilai awal 99,43%, dan akurasi pelatihan sebesar 99,42%. Linear Discriminant Analysis (LDA) berada di posisi ketiga dengan akurasi pelatihan 97,69% dan validasi 97,89%, namun hyperparameter tuning tidak memberikan adanya peningkatan signifikan. Naïve Bayes, meskipun di posisi terakhir, mencatat peningkatan kecil setelah tuning dengan akurasi validasi terbaik 97,01% dari nilai awal 96,99%. Evaluasi data pengujian mengonfirmasi bahwa Random Forest tetap unggul dengan akurasi 99,97%, diikuti Logistic Regression 99,42%, LDA 97,68%, dan Naïve Bayes 97,07%. Random Forest secara konsisten menunjukkan keseimbangan presisi dan recall yang optimal, meminimalkan kesalahan positif palsu dan mempertahankan performa tinggi di semua subset data. Berdasarkan hasil yang diperoleh, kelebihan dan kekurangan algoritma Random Forest, Linear Disrcriminant Analysis , Logistic Regression dan Naive Bayes terdapat pada tabel 4.1.

Tabel 4.1 Kelebihan Dan Kekurangan Setiap Algoritma

Algoritma Machine Learning	Kelebihan	Kekurangan
Random Forest	Akurasi tinggi karena mekanisme ensemble learning sehingga tahan overfitting, konsisten menjaga keseimbangan presisi dan recall sehingga meminimalkan false positive dan false negative, fleksibel terhadap dataset kompleks.	Membutuhkan sumber daya komputasi yang besar terutama pada dataset besar seperti NSL-KDD, sulit diinterpretasikan karena struktur modelnya yang kompleks.
Linear Discriminant Analysis	Sangat efisien karena menggunakan distribusi data untuk membuat keputusan sehingga proses komputasinya ringan, memiliki performa stabil, bekerja optimal pada dataset dengan distribusi normal dan hubungan linear antar variabel prediktor.	Tidak mampu menangani dataset dengan pola non-linear sehingga fleksibilitasnya terbatas, performa model dapat menurun jika asumsi normalitas atau homogenitas varians antar kelas tidak terpenuhi.
Logistic Regression	Efisien secara komputasi, bekerja optimal pada dataset dengan hubungan linear antar fitur, mudah diinterpretasikan	Memiliki keterbatasan dalam menangani data dengan pola non-linear, sensitif terhadap multikolinearitasi dimana korelasi tinggi antar fitur dapat memengaruhi stabilitas model
Naïve Bayes	Sangat cepat dalam proses training dan prediksi sehingga ideal diterapkan pada dataset besar, fleksibel menangani fitur diskrit maupun kontinu, mudah diimplementasikan pada dataset dengan banyak fitur.	Akurasi model dapat menurun karena asumsi independensi antar fitur seringkali tidak realistis pada dataset kompleks, memiliki keterbatasan dalam menangkap hubungan non-linear antar variabel.

Dalam lima tahun terakhir, berbagai teknik telah dieksplorasi oleh peneliti untuk mengoptimalkan sistem deteksi keamanan jaringan. Penelitian mengenai pemanfaatan algoritma machine learning untuk mendeteksi serangan, khususnya serangan DoS, menunjukkan bahwa sebagian besar studi cenderung tidak menggunakan metode hyperparameter tuning untuk meningkatkan akurasi

model atau mencari kombinasi hyperparameter yang optimal. Hal ini disebabkan oleh kebutuhan sumber daya komputasi yang jauh lebih besar saat menerapkan hyperparameter tuning, terutama jika dataset yang digunakan bersifat kompleks dan memiliki volume data yang sangat besar. Proses tuning memerlukan waktu dan tenaga komputasi yang signifikan karena melibatkan pengujian berbagai kombinasi parameter secara berulang untuk menemukan konfigurasi terbaik. Pada kondisi tertentu, hasil tuning juga bergantung pada dataset yang digunakan, sehingga model yang dioptimalkan pada satu dataset belum tentu memberikan kinerja serupa pada dataset lain, yang dapat mengurangi generalisasi model. Kompleksitas tambahan yang dihadirkan oleh proses tuning ini sering kali tidak sebanding dengan peningkatan kinerja yang dihasilkan, terutama untuk kasus penggunaan yang tidak memerlukan tingkat akurasi yang sangat tinggi. Sehingga banyak peneliti lebih memilih pendekatan sederhana tanpa tuning, terutama untuk menghindari beban komputasi yang berat dan menjaga efisiensi waktu penelitian.

Hasil penelitian menunjukkan bahwa hyperparameter tuning secara signifikan meningkatkan performa model, terutama pada Random Forest dan Logistic Regression, dengan akurasi tinggi yang konsisten di berbagai subset data. Temuan ini sejalan dengan literatur yang menyoroti kemampuan adaptif Random Forest terhadap tuning, meskipun prosesnya memerlukan sumber daya komputasi besar, terutama pada dataset kompleks seperti NSL-KDD. Minimnya peningkatan pada LDA dan Naïve Bayes juga konsisten dengan studi yang menyebutkan parameter terbatas pada algoritma ini membuat dampak tuning cenderung marginal. Penelitian ini menegaskan bahwa meskipun tuning efektif, biaya komputasi dan keterbatasan generalisasi perlu dipertimbangkan, mendukung pandangan literatur bahwa penerapan tuning harus disesuaikan dengan kebutuhan penelitian. Meskipun demikian, hasil penelitian ini relevan dengan tren AI di bidang keamanan siber, yang berfokus pada peningkatan kinerja model deteksi ancaman melalui optimasi seperti hyperparameter tuning. Pendekatan ini mencerminkan tren modern dalam pengembangan model AI yang lebih akurat dan adaptif, sebagaimana dibutuhkan untuk deteksi serangan real-time. Penelitian ini mendukung pengembangan teknologi AI yang lebih cerdas dan efisien untuk menjawab kompleksitas ancaman siber modern.

Hasil penelitian ini dapat diterapkan secara nyata dalam dunia keamanan siber, seperti pada sistem deteksi intrusi (IDS) di perusahaan untuk memonitor aktivitas jaringan secara real-time menggunakan Random Forest yang dioptimalkan, atau Logistic Regression sebagai solusi ringan bagi organisasi dengan sumber daya komputasi terbatas. Dalam pusat operasi keamanan (SOC), Random Forest dapat memberikan analisis ancaman yang presisi, sedangkan ISP dan infrastruktur publik dapat memanfaatkan model berbasis machine learning untuk melindungi jaringan besar dari serangan DoS melalui sistem berbasis cloud. Selain itu, algoritma seperti Naïve Bayes atau LDA yang efisien secara komputasi cocok untuk keamanan perangkat IoT dengan sumber daya terbatas. Namun, penelitian ini memiliki beberapa keterbatasan, seperti generalisasi model yang bergantung pada dataset NSL-KDD, biaya komputasi tinggi untuk hyperparameter tuning, keterbatasan algoritma tertentu seperti Logistic Regression, LDA, dan Naïve Bayes dalam menangani hubungan non-linear, serta belum adanya pengujian pada data real-time yang dapat memengaruhi performa model dalam kondisi sebenarnya. Penggunaan dataset terstandar juga membatasi representasi ancaman jaringan modern, dan interpretabilitas model kompleks seperti Random Forest menjadi tantangan di industri yang membutuhkan transparansi keputusan, seperti di sektor keuangan atau pemerintahan.

5. KESIMPULAN

Berdasarkan evaluasi yang telah dilakukan, Random Forest menunjukkan performa terbaik dalam mendeteksi serangan DoS dengan akurasi, presisi, recall, dan F1-Score 100% pada data pelatihan serta 99,97% pada data validasi dan pengujian, mencerminkan konsistensi dan deteksi yang sangat baik. Logistic Regression stabil di atas 99%, sedangkan LDA dan Naïve Bayes mencapai akurasi sekitar 97-98%. Pemanfaatan hyperparameter tuning meningkatkan akurasi pada algoritma Random Forest, Logistic Regression dan Naïve Bayes, namun tidak signifikan pada LDA yang sudah optimal. Random Forest terbukti andal dalam mengenali pola dan beradaptasi dengan ancaman, menjadikannya pilihan tepat untuk melindungi jaringan, termasuk dalam sistem IoT dan SDN. Namun, tantangan di masa depan akan tetap ada, termasuk kebutuhan dataset yang mempunyai kualitas tinggi, kemampuan untuk terus beradaptasi terhadap metode serangan baru, pengelolaan beban komputasi, serta pengurangan kesalahan deteksi (false positives) tanpa mengurangi tingkat akurasi deteksi. Mengatasi tantangan tersebut melalui inovasi yang terfokus akan menjadi kunci untuk lebih meningkatkan sistem deteksi DoS menggunakan machine learning sehingga tetap tangguh dan efektif dalam menghadapi ancaman siber yang terus berkembang.

Dampak penting dari penelitian ini pada bidang keamanan siber adalah pemanfaatan algoritma machine learning untuk meningkatkan deteksi serangan DoS dengan optimal. Random Forest, dengan kinerja terbaik di antara algoritma lainnya, menunjukkan potensi besar dalam meminimalkan dampak serangan dan meningkatkan efisiensi dalam melindungi jaringan. Hasil penelitian ini juga

menyoroti pentingnya hyperparameter tuning untuk meningkatkan kinerja model deteksi, terutama dalam menghadapi ancaman yang berkembang dengan cepat. Saran untuk penelitian mendatang adalah penggunaan dataset yang lebih beragam dan mencerminkan ancaman nyata yang lebih kompleks, seperti serangan multi-vektor atau serangan DoS dengan variansi lebih besar. Hal ini penting untuk menguji keandalan model dalam lingkungan yang dinamis. Selain itu, pengujian model pada data real-time akan memberikan wawasan lebih lanjut tentang bagaimana kinerja model dalam situasi serangan yang sebenarnya, yang bisa berbeda dari evaluasi berbasis dataset statis. Penelitian lebih lanjut juga dapat mengeksplorasi teknik pengoptimalan yang lebih efisien, seperti AutoML, untuk mengatasi tantangan sumber daya komputasi yang tinggi dalam hyperparameter tuning.

Berdasarkan hasil penelitian ini dapat dilakukan implementasi Random Forest dalam sistem deteksi intrusi (IDS) untuk meningkatkan keamanan jaringan. Organisasi dapat memanfaatkan kemampuan Random Forest dalam mendeteksi serangan DoS dengan akurasi tinggi, namun perlu mempertimbangkan biaya komputasi yang besar. Di sisi lain, algoritma seperti Logistic Regression dapat digunakan untuk aplikasi dengan sumber daya terbatas, sedangkan Naïve Bayes dan LDA bisa dipilih untuk sistem yang membutuhkan komputasi lebih ringan meskipun dengan kinerja yang sedikit lebih rendah.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Universitas Amikom Purwokerto atas dukungan dana dan fasilitas yang diberikan sehingga penelitian ini dapat dilaksanakan dengan baik. Dukungan yang diberikan memiliki peran yang sangat penting dalam kelancaran proses penelitian dan penyusunan artikel ini. Semoga kontribusi ini dapat memberikan manfaat yang luas bagi pengembangan ilmu pengetahuan dan kemajuan institusi. Terima kasih atas kepercayaan dan kesempatan yang telah diberikan.

DAFTAR PUSTAKA

[1] A. A. Ojugo and R. E. Yoro, "Forging a deep learning neural network intrusion detection framework to curb the distributed denial of service attack," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1498–1509, 2021, doi: 10.11591/ijece.v11i2.pp1498-1509.

[2] P. Radhakrishnan *et al.*, "DoS attack detection and hill climbing based optimal forwarder selection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 36, no. 2, pp. 882–891, 2024, doi: 10.11591/ijeecs.v36.i2.pp882-891.

[3] L. Narayanan, S. .R, G. Julie, Y. Robinson, and V. Shanmuganathan, "Machine Learning Based Detection and a Novel EC-BRTT Algorithm Based Prevention of DoS Attacks in Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 127, pp. 1–25, Feb. 2021, doi: 10.1007/s11277-021-08277-7.

[4] M. Bogdanoski, T. Shuminoski, and A. Risteski, "Analysis of the SYN Flood DoS Attack," *Int. J. Comput. Netw. Inf. Secur.*, vol. 5, no. 8, pp. 15–11, 2013, doi: 10.5815/ijcnis.2013.08.01.

[5] M. M. Rasheed, A. K. Faieq, and A. A. Hashim, "Development of a new system to detect denial of service attack using machine learning classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1068–1072, 2021, doi: 10.11591/ijeecs.v23.i2.pp1068-1072.

[6] N. Tripathi and B. Mehtre, "DoS and DDoS Attacks : Impact , Analysis and Countermeasures," *Adv. Comput. Netw. Secur.*, no. December 2013, pp. 1–6, 2014.

[7] G. Kumar, "Denial of service attacks–an updated perspective," *Syst. Sci. Control Eng.*, vol. 4, no. 1, pp. 285–294, 2016, doi: 10.1080/21642583.2016.1241193.

[8] R. B. Blaˇ, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods," *Proc. IEEE …*, no. June, pp. 1–7, 2001, [Online]. Available: http://www.professores.unirg.edu.br/marcelo/coordenacao/mar/doutorado/ufrj/DoSDectionPaper.pdf

[9] O. Boyar, M. E. Özen, and B. Metin, "Detection of Denial-of-Service Attacks with SNMP/RMON," *INES 2018 - IEEE 22nd Int. Conf. Intell. Eng. Syst. Proc.*, no. March, pp. 000437–000440, 2018, doi: 10.1109/INES.2018.8523851.

[10] Y. Xie, "Machine learning-based DDoS detection for IoT networks," *Appl. Comput. Eng.*, vol. 29, no. 1, pp. 99–107, 2023, doi: 10.54254/2755-2721/29/20230972.

[11] S. Situmorang and Yahfizham, "Analisis Kinerja Algoritma Machine Learning Dalam Deteksi Anomali Jaringan," *J. Mat. dan Ilmu Pengetah. Alam*, vol. 1, no. 4, pp. 258–269, 2023, [Online]. Available: https://doi.org/10.59581/konstanta.v1i4.1722

[12] S. Wali, I. Khan, I. A. Khan, and S. Member, "Explainable AI and Random Forest Based Reliable Intrusion Detection system," *Authorea Prepr.*, pp. 1–9, 2023, [Online]. Available: https://www.authorea.com/doi/full/10.36227/techrxiv.17169080.v1?commit=efd2c0478c

7643c3f5acbea1e1e16753d0a2c2e6

[13] S. Wang *et al.*, "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques," *Eng. Sci. Technol. an Int. J.*, vol. 35, no. November, 2022, p. 101176, 2022, doi: 10.1016/j.jestch.2022.101176.

[14] W. F. Urmi *et al.*, "A stacked ensemble approach to detect cyber attacks based on feature selection techniques," *Int. J. Cogn. Comput. Eng.*, vol. 5, no. January, pp. 316–331, 2024, doi: 10.1016/j.ijcce.2024.07.005.

[15] Y. Shang, "Prevention and detection of DDOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning," *Meas. Sensors*, vol. 31, no. July 2023, p. 100991, 2024, doi: 10.1016/j.measen.2023.100991.

[16] D. R. Wijaya, N. F. Syarwan, M. A. Nugraha, D. Ananda, T. Fahrudin, and R. Handayani, "Seafood Quality Detection Using Electronic Nose and Machine Learning Algorithms With Hyperparameter Optimization," *IEEE Access*, vol. 11, pp. 62484–62495, 2023, doi: 10.1109/ACCESS.2023.3286980.

[17] A. Oyetoro, J. Mart, U. Amah, J. M. Orcid, and U. A. Orcid, "Using Machine Learning Techniques Random Forest and Neural Network to Detect Cyber Attacks," *Sci. Prepr.*, no. April 2023, 2023, doi: 10.14293/PR2199.000059.v1.

[18] M. Sipper, "High Per Parameter: A Large-Scale Study of Hyperparameter Tuning for Machine Learning Algorithms," *Algorithms*, vol. 15, no. 9, p. 315, 2022, doi: 10.3390/a15090315.

[19] F. Arden and C. Safitri, "Hyperparameter Tuning Algorithm Comparison with Machine Learning Algorithms," *Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng.*, no. Desember 2022, pp. 183–188, Dec. 2022, doi: 10.1109/ICITISEE57756.2022.10057630.

[20] S. Simon, N. Kolyada, C. Akiki, M. Potthast, B. Stein, and N. Siegmund, "Exploring Hyperparameter Usage and Tuning in Machine Learning Research," *2023 IEEE/ACM 2nd Int. Conf. AI Eng. – Softw. Eng. AI*, pp. 68–79, May 2023, doi: 10.1109/CAIN58948.2023.00016.

[21] A. A. Salih and M. B. Abdulrazaq, "Cybernet Model: A New Deep Learning Model for Cyber DDoS Attacks Detection and Recognition," *Comput. Mater. Contin.*, vol. 78, no. 1, pp. 1275–1295, 2024, doi: 10.32604/cmc.2023.046101.

[22] I. Zada *et al.*, "Fine-Tuning Cyber Security Defenses: Evaluating Supervised Machine Learning Classifiers for Windows Malware Detection," *Comput. Mater. Contin.*, vol. 80, no. 2, pp. 2917–2939, 2024, doi: 10.32604/cmc.2024.052835.

[23] M. Houichi, F. Jaidi, and A. Bouhoula, "Cyber Security within Smart Cities: A Comprehensive Study and a Novel Intrusion Detection-Based Approach," *Comput. Mater. Contin.*, vol. 81, no. 1, pp. 393–441, 2024, doi: 10.32604/cmc.2024.054007.

[24] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decis. Anal. J.*, vol. 7, no. April, p. 100233, 2023, doi: 10.1016/j.dajour.2023.100233.

[25] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1251–1260, 2020, doi: 10.1016/j.procs.2020.04.133.

[26] M. H. Zaib, "NSL-KDD." Accessed: Sep. 10, 2024. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd/data?select=KDDTest-21.arff

[27] Asif, Bibi, Arif, and Mukheimer, "Mejora de la predicción de enfermedades cardíacas mediante técnicas de aprendizaje por conjuntos con optimización de hiperparámetros," *Algorithms*, vol. 16, no. 6, pp. 1–17, 2023, doi: 10.3390/a16060308.

[28] A. Lailiyah, V. R. S. Nastiti, E. D. Wahyuni, and C. S. K. Aditya, "Klasifikasi Tingkat Kemampuan Adaptasi Siswa dalam Pembelajaran Online Menggunakan Decision Tree," *Techno.Com*, vol. 23, no. 1, pp. 11–19, 2024, doi: 10.62411/tc.v23i1.9739.

[29] C. Starbuck, *The Fundamentals of People Analytics With Applications in R*. 2023. doi: 10.1007/978-3-031-28674-2.

[30] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 4, pp. 533–544, 2022, doi: 10.22266/ijies2022.0831.48.

[31] R. Kaur and N. Gupta, *Comprehending SMOTE Adaptations to Alleviate Imbalance in Intrusion Detection Systems*. 2023. doi: 10.1109/ICESC57686.2023.10193257.

[32] A. Tesfahun and L. Bhaskari, *Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction*. 2013. doi: 10.1109/CUBE.2013.31.

[33] K. Swarnalatha, N. Narisetty, G. R. Kancherla, and B. Bobba, "Analyzing Resampling Techniques for Addressing the

Class Imbalance in NIDS using SVM with Random Forest Feature Selection," *Int. J. Exp. Res. Rev.*, vol. 43, no. September 2024, pp. 42–55, 2024, doi: 10.52756/ijerr.2024.v43spl.004.

[34] Y.-H. Seo, Jae-Hyun and Kim, "Machine Learning Approach to Optimize SMOTE Ratio in Class Imbalance Dataset for Intrusion Detection," *Comput. Intell. Neurosci.*, vol. 2018, no. November 2018, pp. 1–11, 2018, doi: 0.1155/2018/9704672.

[35] P. Araujo *et al.*, "Impact of Feature Selection Methods on the Classification of DDoS Attacks using XGBoost," *J. Commun. Inf. Syst.*, vol. 36, no. 1, pp. 200–214, 2021, doi: 10.14209/jcis.2021.22.

[36] M. A. Hossain and M. S. Islam, "Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity," *Meas. Sensors*, vol. 32, no. January, p. 101037, 2024, doi: 10.1016/j.measen.2024.101037.

[37] K. L. Goh, A. Goto, and Y. Lu, "LGB-Stack: Stacked Generalization with LightGBM for Highly Accurate Predictions of Polymer Bandgap," *ACS Omega*, vol. 7, no. 34, pp. 29787–29793, 2022, doi: 10.1021/acsomega.2c02554.

[38] M. Purba *et al.*, "Effect of Random Splitting and Cross Validation for Indonesian Opinion Mining using Machine Learning Approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, pp. 145–151, 2022, doi: 10.14569/IJACSA.2022.0130917.

[39] M. Riskiyadi, "Detecting future financial statement fraud using a machine learning model in Indonesia: a comparative study," *Asian Rev. Account.*, vol. 32, no. 3, pp. 394–422, 2024, doi: 10.1108/ARA-02-2023-0062.

[40] J. Simm *et al.*, "Splitting chemical structure data sets for federated privacy-preserving machine learning," *J. Cheminform.*, vol. 13, no. 1, pp. 1–14, 2021, doi: 10.1186/s13321-021-00576-2.

[41] V. G. Prabhu, K. Taaffe, and R. Pirrallo, "A multi-layered LSTM for predicting physician stress during an ED shift," *Proc. 2020 IISE Annu. Conf.*, pp. 1223–1228, 2020.

[42] M. Agus Syamsul Arifin, A. Anto Tri Susilo, A. Taqwa Martadinata, and B. Santoso, "KLIK: Kajian Ilmiah Informatika dan Komputer Deteksi Aktifitas Malware pada Internet of Things menggunakan Algoritma Decision Tree dan Random Forest," *Media Online)*, vol. 4, no. 6, pp. 3073–3079, 2024, doi: 10.30865/klik.v4i6.1903.

[43] O. Adiputra and E. Setiawan, "Klasifikasi Malicious URL Menggunakan Algoritma Improved Random Forest dan Random Forest Berbasis Web," *J. Sains dan Inform.*, vol. 9, no. 1, pp. 8–14, 2023, doi: 10.22216/jsi.v9i1.1378.

[44] M. Azhari, Z. Situmorang, and R. Rosnelly, "Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 640, 2021, doi: 10.30865/mib.v5i2.2937.

[45] V. Vajrobol, B. B. Gupta, and A. Gaurav, "Mutual information based logistic regression for phishing URL detection," *Cyber Secur. Appl.*, vol. 2, no. January, p. 100044, 2024, doi: 10.1016/j.csa.2024.100044.

[46] O. Mjahed, S. El Hadaj, E. M. El Guarmah, and S. Mjahed, "Improved Supervised and Unsupervised Metaheuristic-Based Approaches to Detect Intrusion in Various Datasets," *C. - Comput. Model. Eng. Sci.*, vol. 137, no. 1, pp. 265–298, 2023, doi: 10.32604/cmes.2023.027581.

[47] A. Shafiq, A. B. Çolak, T. N. Sindhu, S. A. Lone, and T. A. Abushal, "Modeling and survival exploration of breast carcinoma: A statistical, maximum likelihood estimation, and artificial neural network perspective," *Artif. Intell. Life Sci.*, vol. 4, no. July, p. 100082, 2023, doi: 10.1016/j.ailsci.2023.100082.

[48] Q. Ma, "Recent applications and perspectives of logistic regression modelling in healthcare," *Theor. Nat. Sci.*, vol. 36, no. 1, pp. 185–190, 2024, doi: 10.54254/2753-8818/36/20240614.

[49] S. D. Permai and K. Herdianto, "Prediction of Health Insurance Claims Using Logistic Regression and XGBoost Methods," *Procedia Comput. Sci.*, vol. 227, pp. 1012–1019, 2023, doi: 10.1016/j.procs.2023.10.610.

[50] R. Kumar, B. Goswami, S. Mhatre, and S. Agrawal, "Naive Bayes in Focus: A Thorough Examination of its Algorithmic Foundations and Use Cases," *Int. J. Innov. Sci. Res. Technol.*, no. June 2024, pp. 2078–2081, Jun. 2024, doi: 10.38124/ijisrt/IJISRT24MAY1438.

[51] M. Schonlau, "The Naive Bayes Classifier," in *inbook*, 2023, pp. 143–160. doi: 10.1007/978-3-031-33390-3_8.

[52] A. K. M. A. Habib, M. K. Hasan, R. Hassan, S. Islam, R. Thakkar, and N. Vo, "Distributed denial-of-service attack detection for smart grid wide area measurement system: A hybrid machine learning technique," *Energy Reports*, vol. 9,

no. December 2022, pp. 638–646, 2023, doi: 10.1016/j.egyr.2023.05.087.

[53] Y. N. Fuadah, M. A. Pramudito, and K. M. Lim, "An Optimal Approach for Heart Sound Classification Using Grid Search in Hyperparameter Optimization of Machine Learning," *Bioengineering*, vol. 10, no. 1, p. 45, 2023, doi: 10.3390/bioengineering10010045.