ANALISIS EFEKTIVITAS INTRUSION DETECTION SYSTEM (IDS) UNTUK SERANGAN DDOS MENGGUNAKAN KOMPARASI *RANDOM FOREST* DAN *DECISION TREE*

Muhamad Arief Rachmatullah, Amar Subagja Firdaus, Naufal Afaf Ekayana, Fieren Al-hilal Saepul Bahri, Muhfiz Zauzi
Informatika, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten
Syeikh Nawawi Al-Bantani No.30 Curug, Kota Serang, Indonesia
muhariefrac@gmail.com

ABSTRAK

Serangan Distributed Denial of Service (DDoS) merupakan salah satu ancaman utama terhadap ketersediaan layanan jaringan yang menuntut sistem deteksi intrusi (Intrusion Detection System / IDS) yang akurat dan efisien. Permasalahan dalam pengembangan IDS berbasis machine learning terletak pada trade-off antara tingkat akurasi deteksi dan efisiensi komputasi, di mana algoritma sederhana cenderung kurang akurat, sementara metode kompleks membutuhkan sumber daya yang besar. Penelitian ini bertujuan untuk menganalisis dan membandingkan efektivitas algoritma Random Forest dan Decision Tree dalam mendeteksi serangan DDoS pada sistem IDS. Metode penelitian dilakukan dengan menggunakan dataset CIC-IDS2017, khususnya skenario Friday-WorkingHours-Afternoon-DDoS, melalui tahapan pra-pemrosesan data, pembagian data latih dan data uji dengan rasio 80:20, serta implementasi model klasifikasi menggunakan pustaka Scikit-learn. Evaluasi kinerja dilakukan berdasarkan metrik akurasi, precision, recall, F1-score, analisis Confusion Matrix, serta waktu pelatihan sebagai indikator efisiensi komputasi. Hasil penelitian menunjukkan bahwa kedua algoritma mampu mencapai tingkat akurasi yang sangat tinggi, yaitu 99,99%. Namun, dari sisi efisiensi, Decision Tree memiliki waktu pelatihan yang jauh lebih cepat dibandingkan Random Forest, sehingga lebih sesuai untuk implementasi IDS yang membutuhkan respons deteksi secara real-time pada sistem dengan keterbatasan sumber daya.

Kata kunci : *DDoS, Intrusion Detection System, Random Forest, Decision Tree, CIC-IDS2017.*

1. PENDAHULUAN

Keamanan siber merupakan aspek krusial dalam infrastruktur teknologi informasi seiring dengan meningkatnya ketergantungan layanan digital terhadap jaringan komputer. Salah satu ancaman utama yang berdampak langsung pada aspek ketersediaan (availability) adalah serangan Distributed Denial of Service (DDoS), yang bekerja dengan membanjiri sumber daya jaringan menggunakan lalu lintas berbahaya sehingga layanan menjadi tidak dapat diakses oleh pengguna yang sah [1]. Seiring berkembangnya pola serangan DDoS yang semakin kompleks dan masif, mekanisme pertahanan jaringan konvensional menjadi kurang efektif, sehingga dibutuhkan sistem deteksi intrusi (Intrusion Detection System / IDS) yang mampu melakukan deteksi secara otomatis dan akurat dalam waktu nyata.

Berbagai pendekatan telah dikembangkan untuk meningkatkan kinerja IDS, khususnya dengan memanfaatkan algoritma Machine Learning. Namun, permasalahan utama dalam pengembangan IDS berbasis pembelajaran mesin terletak pada trade-off antara tingkat akurasi dan efisiensi komputasi [2]. Beberapa algoritma sederhana seperti Naïve Bayes terbukti memiliki akurasi rendah dalam mendeteksi serangan DDoS modern [3], sementara pendekatan berbasis Deep Learning mampu menghasilkan akurasi yang sangat tinggi tetapi memerlukan sumber daya komputasi besar serta waktu pelatihan yang lama [4]. Kondisi ini menjadi kendala tersendiri bagi implementasi IDS pada sistem dengan keterbatasan sumber daya atau yang membutuhkan respons deteksi secara real-time.

Berdasarkan permasalahan tersebut, diperlukan evaluasi terhadap algoritma klasifikasi yang mampu memberikan performa deteksi tinggi sekaligus efisien secara komputasi. Algoritma berbasis pohon keputusan, seperti Random Forest dan Decision Tree, telah banyak digunakan dalam deteksi intrusi jaringan karena kemampuannya dalam menangani data berdimensi tinggi serta pola serangan yang kompleks. Penelitian sebelumnya menunjukkan bahwa Decision Tree memiliki keunggulan dari sisi kecepatan eksekusi, sedangkan Random Forest unggul dalam stabilitas dan ketahanan terhadap overfitting [5]. Namun demikian, perbandingan kinerja kedua algoritma tersebut masih perlu divalidasi menggunakan dataset yang merepresentasikan lalu lintas jaringan modern.

Oleh karena itu, tujuan penelitian ini adalah untuk menganalisis dan membandingkan efektivitas algoritma Random Forest dan Decision Tree dalam mendeteksi serangan DDoS pada sistem IDS, dengan menggunakan dataset CIC-IDS2017 yang dirancang untuk merepresentasikan skenario serangan jaringan terkini [6]. Evaluasi dilakukan dengan meninjau tingkat akurasi, analisis kesalahan klasifikasi menggunakan Confusion Matrix, serta efisiensi waktu pelatihan sebagai indikator kelayakan implementasi pada sistem real-time.

Untuk menyelesaikan permasalahan tersebut, penelitian ini dilakukan melalui beberapa tahapan, yaitu pra-pemrosesan dataset CIC-IDS2017,

pembagian data menjadi data latih dan data uji, pembangunan model klasifikasi menggunakan algoritma Random Forest dan Decision Tree, serta evaluasi kinerja kedua model berdasarkan metrik akurasi, Confusion Matrix, dan waktu pelatihan. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi algoritma yang paling sesuai untuk diterapkan pada sistem IDS dengan mempertimbangkan keseimbangan antara akurasi dan efisiensi komputasi.

2. TINJAUAN PUSTAKA

Deteksi serangan siber telah menjadi topik riset yang intensif dalam satu dekade terakhir. Khraisat dkk. [7] dalam survei komprehensifnya menyatakan bahwa evolusi serangan DDoS menuntut adanya sistem deteksi (IDS) yang tidak hanya akurat tetapi juga adaptif terhadap pola serangan baru. Metode tradisional berbasis tanda tangan (*signature-based*) dinilai semakin kurang efektif, sehingga pendekatan berbasis *Machine Learning* menjadi standar baru dalam pengembangan IDS.

Beberapa penelitian berupaya menerapkan algoritma klasifikasi untuk mengenali serangan ini. Mas'ud dkk. [3] menguji algoritma *Naïve Bayes* pada protokol *Zigbee* untuk deteksi DDoS. Meskipun komputasinya ringan, hasil eksperimen mereka menunjukkan akurasi yang rendah (45%) dan tingkat *False Negative* yang tinggi. Hal ini dikuatkan oleh temuan Kurniabudi dkk. [8] yang menyatakan bahwa algoritma sederhana sering kali gagal menangkap fitur kompleks dari serangan modern, dan menyarankan penggunaan *Ensemble Learning* seperti *Random Forest* yang terbukti meningkatkan akurasi secara signifikan pada kasus serupa.

Pendekatan yang lebih mutakhir diusulkan oleh Liu dkk. [4] yang menggabungkan *K-Means*, *Random Forest*, dan *Deep Learning* (CNN-LSTM) pada dataset CIC-IDS2017. Metode hibrida ini mencapai akurasi 99.91%, melampaui metode tradisional. Namun, M. Haggag dkk. [9] mencatat bahwa penggunaan metode *Deep Learning* sering kali terkendala oleh waktu pelatihan yang lama dan kebutuhan perangkat keras berkinerja tinggi, yang mungkin tidak ideal untuk implementasi *real-time* di semua lingkungan jaringan. Isu efisiensi ini kembali ditekankan dalam studi terbaru tahun 2025 oleh Ibrahim dkk. [10], yang menunjukkan bahwa untuk mitigasi serangan waktu nyata, kecepatan inferensi menjadi faktor krusial yang sering kali lebih penting daripada peningkatan akurasi marjinal.

Sebagai solusi penyeimbang, Ramadhan dan Ashari [5] membandingkan *Random Forest* dan *Decision Tree* pada dataset UNSW-NB15. Hasilnya menunjukkan bahwa *Decision Tree* mampu memberikan performa deteksi yang kompetitif dengan waktu eksekusi yang jauh lebih cepat dibandingkan *Random Forest*. Senada dengan itu, Devarakonda dkk. [11] yang membandingkan empat pengklasifikasi pada dataset NSL-KDD juga menemukan bahwa algoritma berbasis pohon (*tree-based*) menawarkan keseimbangan terbaik antara presisi dan efisiensi. Temuan ini diperkuat oleh riset Abiramasundari dan Ramaswamy [12] pada tahun 2025 yang mengonfirmasi bahwa pada dataset CIC-IDS2017, algoritma berbasis pohon tetap mendominasi dengan akurasi di atas 98%, mengungguli algoritma berbasis jarak seperti KNN. Bahkan, Ibrahim dkk. [10] membuktikan bahwa *Decision Tree* mampu mencapai kecepatan inferensi 0.004 detik, menjadikannya kandidat kuat untuk sistem pertahanan otomatis.

Dataset CIC-IDS2017 yang digunakan dalam penelitian ini, dikembangkan oleh Sharafaldin dkk. [6], dipilih karena merepresentasikan profil lalu lintas jaringan modern yang mencakup serangan DDoS terkini, mengatasi keterbatasan *dataset* lama seperti KDD99 yang sudah tidak relevan dengan tren serangan saat ini.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental kuantitatif untuk mengevaluasi kinerja algoritma *Random Forest* dan *Decision Tree* dalam mendeteksi serangan *Distributed Denial of Service* (DDoS) pada sistem *Intrusion Detection System* (IDS). Pendekatan ini dipilih karena banyak digunakan dalam penelitian deteksi intrusi berbasis *machine learning* untuk mengukur performa klasifikasi lalu lintas jaringan [7].

3.1. *Dataset*

Data yang digunakan dalam penelitian ini bersumber dari *dataset* **CIC-IDS2017** yang dipublikasikan oleh *Canadian Institute for Cybersecurity* (CIC) [6]. *Dataset* ini dipilih karena merepresentasikan simulasi serangan siber modern dengan profil lalu lintas jaringan yang realistis, mencakup protokol HTTP, HTTPS, FTP, SSH, dan email. Fokus penelitian ini adalah pada file Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv, yang secara spesifik berisi rekaman lalu lintas serangan DDoS beserta lalu lintas normal (*Benign*) yang terjadi pada jam kerja. *Dataset* ini terdiri dari lebih dari 200.000 baris data dengan 78 fitur yang diekstraksi menggunakan *CICFlowMeter*.

3.2. Pra-pemrosesan Data (*Data Preprocessing*)

Tahap pra-pemrosesan data dilakukan untuk meningkatkan kualitas dataset sebelum proses pelatihan model. Data yang mengandung nilai kosong (*missing values*) dan nilai tak hingga (*infinite values*) dihapus karena dapat menyebabkan kesalahan komputasi pada proses pembelajaran model, sebagaimana dilakukan pada penelitian deteksi DDoS sebelumnya [8]. Selanjutnya, atribut label dikonversi dari format kategorikal menjadi numerik menggunakan *label encoding*, dengan kelas 0 merepresentasikan lalu lintas normal (*Benign*) dan kelas 1 merepresentasikan serangan DDoS.

Dataset dibagi menjadi dua bagian, yaitu data latih dan data uji, dengan rasio 80:20. Pembagian ini bertujuan untuk menguji kemampuan generalisasi model terhadap data yang tidak digunakan selama proses pelatihan [5]. Proses pembagian data dilakukan secara acak dengan parameter random state bernilai 42 untuk menjaga konsistensi dan reprodusibilitas hasil eksperimen.

3.3. Algoritma Klasifikasi

Penelitian ini mengimplementasikan dua algoritma klasifikasi, yaitu Random Forest dan Decision Tree, menggunakan bahasa pemrograman Python dengan pustaka *Scikit-learn*. Random Forest digunakan sebagai metode *ensemble learning* berbasis pohon keputusan yang dikenal memiliki stabilitas tinggi dan ketahanan terhadap overfitting pada data berdimensi tinggi [8]. Sementara itu, Decision Tree digunakan sebagai model pohon tunggal untuk mengevaluasi efisiensi komputasi dan kecepatan pelatihan, sebagaimana direkomendasikan pada penelitian sebelumnya [5]. Seluruh algoritma dijalankan menggunakan parameter bawaan (*default parameters*) untuk memastikan perbandingan kinerja yang objektif.

3.4. Evaluasi

Evaluasi kinerja model dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score untuk mengukur performa klasifikasi secara kuantitatif. Selain itu, *Confusion* Matrix digunakan untuk menganalisis kesalahan klasifikasi yang meliputi *True Positive*, *True Negative*, *False Positive*, dan *False Negative* [13]. Efisiensi komputasi dievaluasi berdasarkan waktu pelatihan (training time) masing-masing algoritma, yang diukur dalam satuan detik untuk menilai kelayakan implementasi IDS pada sistem yang membutuhkan respons deteksi secara cepat.

3.5. Lingkungan Eksperimen

Seluruh eksperimen dijalankan pada lingkungan komputasi berbasis CPU tanpa akselerasi GPU, dengan tujuan mensimulasikan kondisi implementasi IDS pada sistem dengan keterbatasan sumber daya. Pendekatan ini sejalan dengan kebutuhan sistem IDS real-time yang menuntut efisiensi komputasi tanpa bergantung pada infrastruktur perangkat keras berkinerja tinggi [10].

4. HASIL DAN PEMBAHASAN

Bab ini menguraikan hasil eksperimen deteksi serangan DDoS menggunakan dataset CIC-IDS2017. Evaluasi sistem dilakukan secara komprehensif mencakup tiga aspek utama: performa deteksi berdasarkan metrik klasifikasi, analisis kesalahan prediksi (*error analysis*) menggunakan *Confusion Matrix*, dan efisiensi komputasi (*computational efficiency*).

4.1. Evaluasi Kinerja Klasifikasi (*Accuracy, Precision, Recall*)

Berdasarkan pengujian pada data uji (*testing set*), kedua algoritma menunjukkan kinerja yang sangat superior. Tabel 1 merangkum performa deteksi berdasarkan metrik Akurasi, Presisi (*Precision*), *Recall*, dan *F1-Score*.

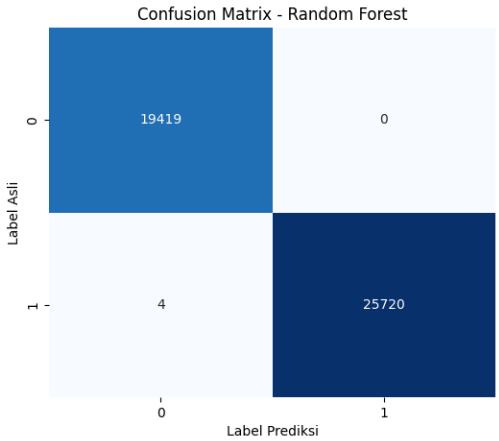
Tabel 1. Performa klasifikasi algoritma

Algoritma	Akurasi	Precision	Recall	F1-Score
Random Forest	99.99%	100%	99.99%	99.99%
Decision Tree	99.99%	99.99%	99.99%	99.99%

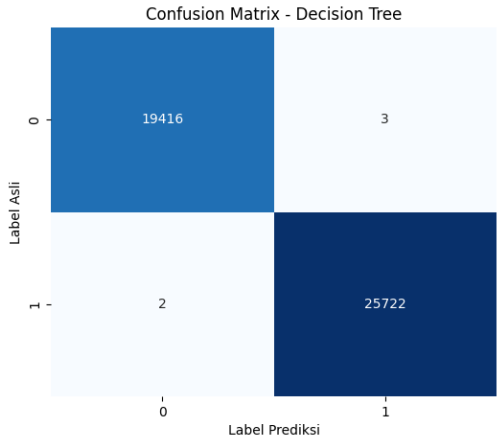
Hasil ini menunjukkan peningkatan yang signifikan dibandingkan penelitian terdahulu oleh Mas'ud dkk. [3] yang menggunakan Naïve Bayes pada kasus serupa namun hanya mencapai akurasi 45%. Tingginya nilai *F1-Score* pada kedua model (mendekati 1.0) mengindikasikan bahwa *Random Forest* dan *Decision Tree* mampu menangani keseimbangan antara presisi dan recall dengan sangat baik, berbeda dengan temuan Karatas dkk. [13] yang mencatat penurunan performa pada algoritma tunggal saat menghadapi data serangan yang tidak seimbang. Hal ini mengonfirmasi bahwa algoritma berbasis pohon (*tree-based*) jauh lebih efektif dalam mengekstraksi fitur kompleks pada serangan DDoS modern dibandingkan algoritma probabilistik sederhana maupun linear.

4.2. Analisis *Confusion Matrix*

Visualisasi *Confusion Matrix* pada Gambar 1 dan Gambar 2 memperlihatkan detail distribusi kesalahan prediksi yang menjadi penentu keandalan sistem keamanan.



Gambar 1. *Confusion Matrix algoritma Random Forest*



Gambar 2. *Confusion Matrix algoritma Decision Tree*

Analisis mendalam terhadap *Confusion Matrix* menunjukkan karakteristik keamanan spesifik:

4.3. *Keandalan Random Forest (Zero False Positive)*

Model ini menunjukkan stabilitas luar biasa dengan 0 *False Positive*. Artinya, tidak ada lalu lintas normal yang salah diklasifikasikan sebagai serangan. Temuan ini sejalan dengan penelitian Kurniabudi dkk. [8] yang menyatakan bahwa metode *ensemble* pada *Random Forest* mampu mengurangi varians error. Dalam operasional jaringan, *False Positive* yang rendah sangat krusial untuk mencegah pemblokiran akses terhadap pengguna yang sah (*legitimate users*), yang dapat mengganggu *Quality of Service* (QoS) [14].

4.4. *Sensitivitas Decision Tree*

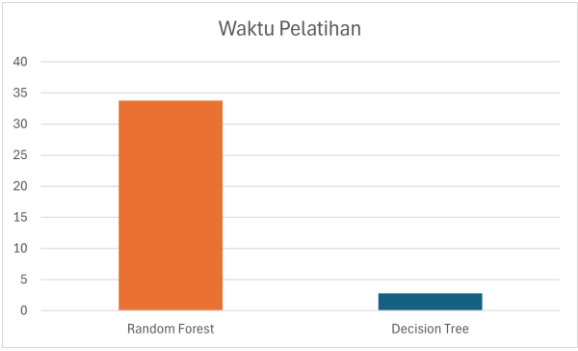
Meskipun memiliki akurasi total yang sama tingginya, model *Decision Tree* menghasilkan 3 False *Positive*. Namun, model ini sedikit lebih sensitif dalam mendeteksi serangan dengan hanya 2 *False Negative*, lebih rendah dibandingkan *Random Forest* yang meloloskan 4 paket serangan. Hasil ini konsisten dengan temuan Ramadhan dan Ashari [5] pada dataset UNSW-NB15, di mana *Decision Tree* seringkali menunjukkan agresivitas deteksi yang lebih tinggi dalam klasifikasi biner.

4.5. Analisis Efisiensi Waktu (*Computational Efficiency*)

Dalam konteks keamanan siber *real-time*, akurasi tinggi saja tidak cukup; kecepatan deteksi adalah faktor krusial untuk memitigasi serangan sebelum server lumpuh. Tabel 2 menyajikan perbandingan waktu pelatihan (*training time*).

Tabel 2. Perbandingan waktu komputasi

Algoritma	Waktu Pelatihan (detik)	Rasio Kecepatan
Random Forest	33.75	1x (*Baseline*)
Decision Tree	2.80	12x Lebih Cepat



Gambar 3. Perbandingan waktu pelatihan (detik)

Data menunjukkan kesenjangan efisiensi yang ekstrem. *Decision Tree* mampu menyelesaikan proses pelatihan hanya dalam 2.80 detik, sedangkan *Random Forest* membutuhkan waktu 33.75 detik. Dengan kata lain, *Decision Tree* bekerja 12 kali lebih cepat dibandingkan *Random Forest*. Efisiensi ini menjadi keunggulan utama *Decision Tree*, sebagaimana didukung oleh teori kompleksitas algoritma di mana struktur pohon tunggal jauh lebih ringan secara komputasi dibandingkan membangun ratusan pohon dalam metode *ensemble* [15].

4.6. Komparasi dengan Penelitian Terdahulu

Hasil penelitian ini memberikan perspektif baru jika dibandingkan dengan metode lain dalam literatur terkini. Akurasi 99.99% yang dicapai menyamai performa metode *Hybrid Deep Learning* (K-Means + CNN-LSTM) yang diusulkan oleh Liu dkk. [4] (99.91%). Namun, pendekatan dalam penelitian ini jauh lebih efisien. Metode *Deep Learning* umumnya membutuhkan infrastruktur komputasi berkinerja tinggi seperti GPU atau *Apache Spark* [9], sedangkan *Decision Tree* dalam penelitian ini terbukti ringan dan dapat dijalankan pada CPU standar.

Selain itu, jika dibandingkan dengan algoritma *Support Vector Machine* (SVM) yang diuji oleh Al-Yaseen dkk. [16], algoritma berbasis pohon terbukti lebih unggul dalam kecepatan pelatihan. SVM memiliki kompleksitas waktu kubik yang membuatnya lambat pada dataset besar seperti CIC-IDS2017, sedangkan *Decision Tree* menawarkan kompleksitas logaritmik yang jauh lebih skalabel. Hal ini juga memperkuat temuan Jan dkk. [17] yang merekomendasikan algoritma *lightweight* untuk deteksi anomali pada lingkungan *Internet of Things* (IoT).

4.7. Implikasi Praktis

Berdasarkan analisis *trade-off* antara akurasi dan efisiensi, pemilihan algoritma harus disesuaikan dengan kebutuhan infrastruktur:

4.8. Untuk Lingkungan IoT dan *Edge Computing*

Decision Tree adalah pilihan terbaik. Kecepatannya yang tinggi (2.80 detik) memungkinkan sistem untuk melakukan adaptasi dan pelatihan ulang (*retraining*) secara periodik terhadap pola serangan

baru tanpa membebani daya komputasi perangkat yang terbatas. Hal ini menjawab tantangan adaptabilitas IDS yang ditekankan oleh Khraisat dkk. [7].

4.9. Untuk Lingkungan Perbankan dan Infrastruktur Kritis

Random Forest lebih direkomendasikan. Meskipun lebih lambat, jaminan 0 *False Positive* memastikan bahwa tidak ada transaksi sah yang terganggu, menjaga aspek *Availability* dan kepercayaan pengguna secara maksimal [11].

Secara keseluruhan, penelitian ini membuktikan bahwa untuk dataset CIC-IDS2017, algoritma *Decision Tree* menawarkan keseimbangan terbaik (*best trade-off*) sebagai solusi IDS yang responsif dan akurat.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil eksperimen, dapat disimpulkan bahwa algoritma *Decision Tree* dan *Random Forest* sama-sama mampu mencapai akurasi deteksi yang sangat tinggi, yaitu 99.99%, pada dataset CIC-IDS2017 untuk skenario serangan DDoS. Namun, *Decision Tree* terbukti jauh lebih efisien dengan waktu pelatihan 2.80 detik, atau 12 kali lebih cepat dibandingkan *Random Forest* yang membutuhkan 33.75 detik, sehingga *Decision Tree* lebih direkomendasikan untuk implementasi sistem deteksi intrusi (IDS) yang membutuhkan respons waktu nyata (*real-time*), meskipun *Random Forest* memiliki keunggulan stabilitas dengan tingkat *False Positive* nol. Untuk penelitian selanjutnya, disarankan agar memperluas cakupan pengujian pada variasi serangan lain seperti *PortScan* atau *Web Attack*, menerapkan teknik seleksi fitur guna mereduksi dimensi data untuk mempercepat komputasi *Random Forest*, serta mengimplementasikan model yang telah dilatih ke dalam lingkungan jaringan nyata menggunakan perangkat lunak IDS berbasis *open-source* untuk validasi performa di lapangan.

DAFTAR PUSTAKA

[1] A. R. Zain, I. M. Malik Matin, and K. Kautsar, "Analisis Implementasi Modsecurity dan Reverse Proxy Untuk Pencegahan Serangan Keamanan DDoS pada Web Server," 2023.

[2] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021, doi: 10.1109/ACCESS.2021.3109081.

[3] I. Mas'ud, K. Kusrini, and A. B. Prasetio, "Distributed Denial of Service (DDOS) Attack Detection on Zigbee Protocol Using Naive Bayes Algorithm," *International Journal of Artificial Intelligence Research*, vol. 5, no. 2, Jun. 2021, doi: 10.29099/ijair.v5i2.214.

[4] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[5] R. F. Ramadhan and W. M. Ashari, "Performance Comparison of Random Forest and Decision Tree Algorithms for Anomaly Detection in Networks," 2024.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SciTePress, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[8] Kurniabudi, A. Haris, and A. Rahim, "Seleksi Fitur dengan Information Gain untuk Meningkatkan Deteksi Serangan DDoS Menggunakan Random Forest," 2020.

[9] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," *IEEE Access*, vol. 8, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[10] A. J. Ibrahim, S. R. Répás, and N. Bektaş, "Feature-Optimized Machine Learning Approaches for Enhanced DDoS Attack Detection and Mitigation," *Computers*, vol. 14, no. 11, Nov. 2025, doi: 10.3390/computers14110472.

[11] A. Devarakonda, N. Sharma, P. Saha, and S. Ramya, "Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2022. doi: 10.1088/1742-6596/2161/1/012043.

[12] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDOS) attack detection using supervised machine learning algorithms," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-024-84879-y.

[13] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.

[14] S. S. Sefati, B. Arasteh, S. Halunga, and O. Fratu, "A comprehensive survey of cybersecurity techniques based on quality of service (QoS) on the Internet of Things (IoT)," *Cluster Comput*, vol. 28, no. 12, Nov. 2025, doi: 10.1007/s10586-025-05449-z.

[15] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine," *Electronics (Switzerland)*, vol. 9, no. 1, Jan. 2020, doi: 10.3390/electronics9010173.

[16] X. Zhao and X. Nie, "Splitting choice and computational complexity analysis of decision trees," *Entropy*, vol. 23, no. 10, Oct. 2021, doi: 10.3390/e23101241.

[17] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019, doi: 10.1109/ACCESS.2019.2907965