Analisis perbandingan algoritma machine learning untuk deteksi serangan DDoS pada jaringan IoT

Muchammad Basroil Billah, Mohammad Idhom*, Hendra Maulana

Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur
Email: 22081010260@student.upnjatim.ac.id, idhom@upnjatim.ac.id, hendra.maulana.if@upnjatim.ac.id

Abstrak

Serangan *Distributed Denial of Service* (DDoS) merupakan ancaman serius bagi keamanan jaringan yang dapat melumpuhkan layanan dan menyebabkan kerugian besar. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja enam algoritma *machine learning* dalam mendeteksi serangan DDoS, yaitu *Logistic Regression*, *Decision Tree*, *Random Forest*, *Gradient Boosting*, *XGBoost*, dan *K-Nearest Neighbors* (KNN). Dataset yang digunakan adalah kombinasi CICDDoS2019 untuk data serangan dan CICIDS2017 untuk data lalu lintas normal, dengan total 1.000.000 sampel yang seimbang (50% serangan dan 50% normal) serta 78 fitur jaringan. *Preprocessing* dilakukan melalui pembersihan data, penanganan *missing values*, dan normalisasi menggunakan *StandardScaler*. Pembagian data dilakukan dengan rasio 80:20 untuk *training* dan *testing* dengan *stratified sampling*. Hasil eksperimen menunjukkan bahwa *XGBoost* mencapai performa terbaik dengan *F1-Score* 99,99%, AUC 0,999999, dan waktu *training* tercepat (5,34 detik), diikuti oleh *Random Forest* dengan *F1-Score* 99,99% dan AUC 0,999984. Validasi menggunakan *5-fold cross-validation* mengkonfirmasi stabilitas model tanpa *overfitting*. Temuan ini menunjukkan bahwa algoritma *ensemble* berbasis *boosting*, khususnya *XGBoost*, merupakan pendekatan yang paling efektif dan efisien untuk sistem deteksi serangan DDoS.

Kata Kunci: DDoS; machine learning; XGBoost; random forest; deteksi serangan

Comparative analysis of machine learning algorithms for DDoS attack detection

Abstract

Distributed Denial of Service (DDoS) attacks pose a serious threat to computer network security, capable of disrupting services and causing significant losses. This study aims to analyze and compare the performance of six machine learning algorithms in detecting DDoS attacks: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, and K-Nearest Neighbors (KNN). The dataset used is a combination of CICDDoS2019 for attack data and CICIDS2017 for normal traffic data, with a total of 1,000,000 balanced samples (50% attack and 50% normal) and 78 network features. Preprocessing was conducted through data cleaning, missing value handling, and normalization using StandardScaler. Data splitting was performed with an 80:20 ratio for training and testing using stratified sampling. Experimental results show that XGBoost achieved the best performance with an F1-Score of 99.99%, AUC of 0.999999, and the fastest training time (5.34 seconds), followed by Random Forest with an F1-Score of 99.99% and AUC of 0.999984. Validation using 5-fold cross-validation confirmed model stability without overfitting. These findings demonstrate that ensemble-based boosting algorithms, particularly XGBoost, represent the most effective and efficient approach for DDoS attack detection systems.

Keywords: *DDoS; machine learning; XGBoost; random forest; attack detection*

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi yang pesat telah menjadikan jaringan komputer sebagai infrastruktur kritis dalam berbagai sektor kehidupan, mulai dari perbankan, pemerintahan, pendidikan, hingga layanan kesehatan (Bhattacharyya & Kalita, 2016). Namun, seiring meningkatnya ketergantungan terhadap layanan berbasis jaringan, ancaman keamanan siber juga mengalami eskalasi yang signifikan (Yan et al., 2016). Salah satu ancaman paling destruktif adalah serangan *Distributed Denial of Service* (DDoS), yang bertujuan untuk melumpuhkan layanan jaringan dengan membanjiri

target menggunakan lalu lintas palsu dalam volume besar secara terkoordinasi dari banyak sumber (Yan et al., 2016).

Berdasarkan laporan *Cloudflare* (2024), serangan DDoS mengalami peningkatan sebesar 117% secara *year-over-year*, dengan volume serangan yang semakin besar dan teknik yang semakin canggih. Dampak ekonomi dari serangan DDoS sangat signifikan, dengan estimasi kerugian rata-rata mencapai 218.000 USD per insiden untuk perusahaan skala menengah (Kaspersky, 2023). Metode deteksi konvensional yang berbasis *signature* dan *threshold* terbukti tidak memadai dalam menghadapi variasi serangan DDoS modern yang terus berevolusi (Dong et al., 2020).

Pendekatan *machine learning* telah menunjukkan potensi besar sebagai solusi deteksi DDoS yang lebih adaptif dan akurat (Buczak & Guven, 2016). Berbagai penelitian sebelumnya telah mengeksplorasi penggunaan algoritma *machine learning* untuk deteksi DDoS. Sahoo et al. (2020) menggunakan *Random Forest* pada dataset CICDDoS2019 dan mencapai akurasi 99,8%. Sementara itu, Hussain et al. (2021) membandingkan *Decision Tree* dan SVM pada dataset CICIDS2017 dengan hasil akurasi masing-masing 99,6% dan 97,3%. Li et al. (2022) menerapkan *XGBoost* untuk deteksi DDoS pada lingkungan IoT dan melaporkan *F1-Score* sebesar 99,5%. Namun, studi komparatif yang komprehensif dengan menggunakan kombinasi kedua dataset benchmark dan mengevaluasi *multiple* algoritma secara bersamaan masih terbatas.

Penelitian ini bertujuan untuk melakukan analisis perbandingan komprehensif terhadap enam algoritma *machine learning*, yaitu *Logistic Regression*, *Decision Tree*, *Random Forest*, *Gradient Boosting*, *XGBoost*, dan *K-Nearest Neighbors* (KNN) dalam mendeteksi serangan DDoS. Kebaruan penelitian ini terletak pada penggunaan dataset gabungan CICDDoS2019 dan CICIDS2017 dengan total 1.000.000 sampel, evaluasi menggunakan *multiple metrics* (*Accuracy*, *Precision*, *Recall*, *F1-Score*, AUC), serta analisis efisiensi komputasi masing-masing algoritma. Hasil penelitian ini diharapkan dapat memberikan rekomendasi algoritma terbaik untuk implementasi sistem deteksi DDoS yang efektif dan efisien.

2. Metode

2.1. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental kuantitatif dengan metode komparatif untuk menganalisis kinerja enam algoritma *machine learning* dalam mendeteksi serangan DDoS. Alur penelitian terdiri dari lima tahapan utama, yaitu pengumpulan dataset, *preprocessing data*, pembagian data, pelatihan model, dan evaluasi performa. Implementasi dilakukan menggunakan bahasa pemrograman *Python* 3.12 (Raschka et al., 2020) dan *XGBoost* (Chen & Guestrin, 2016). Seluruh eksperimen dijalankan pada lingkungan lokal.

2.2. Dataset

Penelitian ini menggunakan kombinasi dua dataset *benchmark* yang bersumber dari *Canadian Institute for Cybersecurity* (CIC). Dataset pertama adalah CICDDoS2019 (Sharafaldin et al., 2019) yang digunakan sebagai sumber data serangan DDoS. Dataset ini merupakan salah satu dataset *benchmark* terkini yang mencakup berbagai jenis serangan DDoS refleksi dalam lingkungan yang realistis (Sharafaldin et al., 2019). Dari dataset ini diambil sepuluh jenis serangan refleksi, yaitu DrDoS_DNS, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NTP, DrDoS_NetBIOS, DrDoS_SNMP, DrDoS_SSDP, DrDoS_UDP, Syn, dan UDPLag. Satu file (TFTP.csv) tidak disertakan karena mengandung inkonsistensi data. Setiap file di-sampling sebanyak 50.000 sampel secara acak sehingga total data serangan yang diperoleh adalah 500.000 sampel.

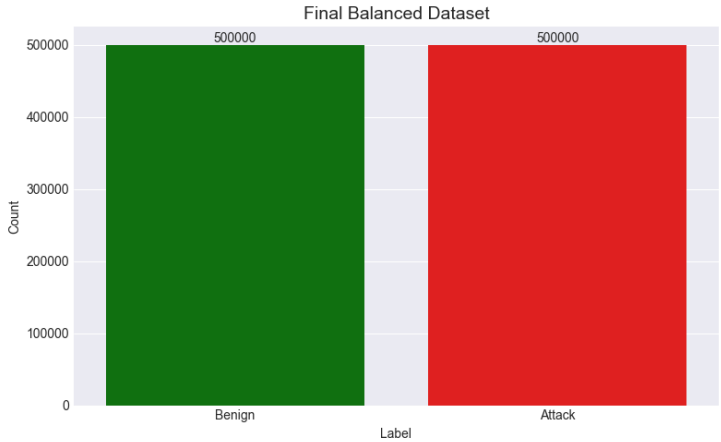
UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK | Canadian Institute for Cybersecurity

Attack Type	Flow Count
Benign	56,863
DDoS_DNS	5,071,011
DDoS_LDAP	2,179,930
DDoS_MSSQL	4,522,492
DDoS_NetBIOS	4,093,279
DDoS_NTP	1,202,642
DDoS_SNMP	5,159,870
DDoS_SSDP	2,610,611
DDoS_SYN	1,582,289
DDoS_TFTP	20,082,580
DDoS_UDP	3,134,645
DDoS_UDP-Lag	366,461
DDoS_WebDDoS	439

Gambar 1. Dataset Serangan CIC-DDoS2019

Dataset kedua adalah CICIDS2017 (Sharafaldin et al., 2018), khususnya file *Monday-WorkingHours* yang berisi rekaman lalu lintas jaringan normal (*benign*). Dari file tersebut diperoleh 529.918 sampel berlabel benign. Pemilihan file *Monday-WorkingHours* didasarkan pada karakteristiknya yang hanya mengandung lalu lintas normal tanpa campuran serangan (Panigrahi & Borah, 2018), sehingga cocok digunakan sebagai representasi kelas negatif.

Kedua dataset kemudian diselaraskan berdasarkan fitur yang saling beririsan, menghasilkan 78 fitur jaringan yang menjadi variabel independen dalam penelitian ini. Untuk mengatasi ketidakseimbangan kelas, dilakukan downsampling pada kelas benign dari 529.918 menjadi 500.000 sampel (Leevy et al., 2018), sehingga diperoleh dataset akhir dengan total 1.000.000 sampel yang seimbang secara sempurna (rasio 1:1 antara kelas serangan dan normal).



Gambar 2. Balancing Data

2.3. Preprocessing Data

Tahap preprocessing dilakukan melalui tiga langkah utama (García et al., 2016). Langkah pertama adalah inspeksi data untuk mengidentifikasi nilai yang hilang (*missing values*) dan nilai tak hingga (*infinity values*). Hasil inspeksi menunjukkan bahwa fitur *Flow Bytes/s* memiliki 11.371 *missing values* (1,14%) serta 7.666 *infinity values*, dan fitur *Flow Packets/s* memiliki 19.037 *infinity values*.

Langkah kedua adalah pembersihan data. Seluruh nilai infinity pada kedua fitur tersebut terlebih dahulu dikonversi menjadi NaN, kemudian seluruh NaN diisi menggunakan nilai median dari masing-masing fitur. Pemilihan median sebagai metode imputasi didasarkan pada sifatnya yang lebih *robust* terhadap *outlier* dibandingkan *mean* (Emmanuel et al., 2021), mengingat distribusi data jaringan yang cenderung memiliki skewness tinggi.

Langkah ketiga adalah normalisasi fitur menggunakan *StandardScaler* (Raschka et al., 2020), yang mentransformasi setiap fitur agar memiliki rata-rata mendekati nol dan standar deviasi satu. Proses *fitting* dilakukan hanya pada data *training* untuk mencegah *data leakage* (Kapoor & Narayanan, 2023), kemudian hasil transformasi diterapkan pada data *training* dan *testing* secara terpisah. Selain itu, label kategorikal (*Attack* & *Benign*) dikonversi menjadi nilai numerik menggunakan *LabelEncoder*, dengan *mapping Attack* = 0 dan *Benign* = 1.

2.4. Pembagian Data

Dataset yang telah melalui tahap preprocessing dibagi menjadi dua bagian menggunakan metode *holdout* dengan rasio 80:20 (Raschka, 2018). Dari total 1.000.000 sampel, sebanyak 800.000 sampel dialokasikan sebagai data training dan 200.000 sampel sebagai data testing. Pembagian dilakukan menggunakan fungsi *train_test_split* (Raschka et al., 2020) dengan parameter *stratify* untuk memastikan proporsi kelas pada data *training* dan *testing* tetap seimbang, yaitu masing-masing 50% *Attack* dan 50% Benign. Nilai random_state ditetapkan sebesar 42 untuk menjamin reprodusibilitas hasil eksperimen.

2.5. Algoritma Machine Learning

Penelitian ini mengevaluasi enam algoritma *machine learning* yang merepresentasikan pendekatan klasifikasi yang berbeda-beda. Seluruh model dilatih menggunakan parameter *default* untuk memastikan perbandingan yang adil tanpa bias dari proses *hyperparameter tuning*. Pemilihan keenam algoritma ini didasarkan pada kebutuhan untuk membandingkan performa antar kategori pendekatan, yaitu model linear berupa *Logistic Regression* (Christodoulou et al., 2019), model berbasis pohon tunggal berupa *Decision Tree* (Charbuty & Abdulazeez, 2021), model *ensemble bagging* berupa *Random Forest* (Biau & Scornet, 2016), model *ensemble boosting* berupa *Gradient Boosting* (Bentéjac et al., 2021) dan *XGBoost* (Chen & Guestrin, 2016), serta model berbasis jarak berupa KNN (Zhang, 2017). Konfigurasi masing-masing algoritma ditunjukkan pada Tabel 1.

Tabel 1. Algoritma Machine Learning

Algoritma	Kategori	Parameter Utama
Logistic Regression	Linear	max_iter = 1000
Decision Tree	Tree-based	default (criterion = gini)
Random Forest	Ensemble (Bagging)	n_estimators = 100, n_jobs = -1
Gradient Boosting	Ensemble (Boosting)	n_estimators = 100
XGBoost	Ensemble (Boosting)	n_estimators = 100, eval_metric = logloss
K-Nearest Neighbors (KNN)	Instance-based	n_neighbors = 5, n_jobs = -1

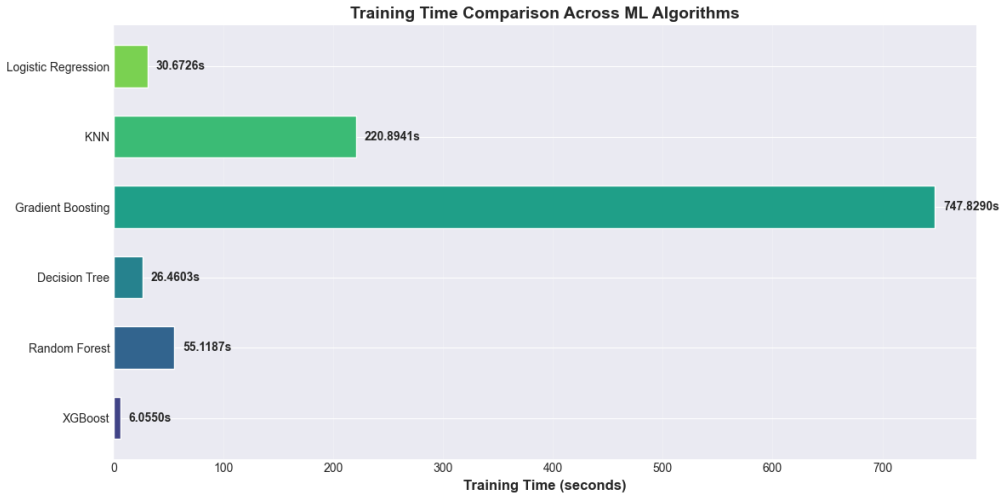
2.6. Metrik Evaluasi

Performa setiap algoritma dievaluasi menggunakan lima metrik utama yang umum digunakan dalam klasifikasi biner (Hossin & Sulaiman, 2015), yaitu *Accuracy, Precision, Recall, F1-Score*, dan AUC (Area Under ROC Curve). Metrik-metrik ini dipilih karena memberikan perspektif evaluasi yang komprehensif, di mana *Accuracy* mengukur proporsi prediksi benar secara keseluruhan, *Precision* mengukur ketepatan prediksi positif, *Recall* mengukur kemampuan mendeteksi seluruh data positif, *F1-Score* memberikan rata-rata harmonis antara *Precision* dan *Recall* (Chicco & Jurman, 2020), serta AUC mengukur kemampuan diskriminasi model pada berbagai *threshold* klasifikasi (Halimu et al., 2019).

3. Hasil dan Pembahasan

3.1. Hasil Perbandingan Algoritma

Hasil pelatihan dan pengujian terhadap enam algoritma *machine learning* ditunjukkan pada Gambar 3. Seluruh model dievaluasi menggunakan data *testing* sebanyak 200.000 sampel yang tidak digunakan dalam proses pelatihan.



Gambar 3. Training Time 6 Algoritma Machine Learning

Tabel 2. Hasil Perbandingan Performa Model Machine Learning

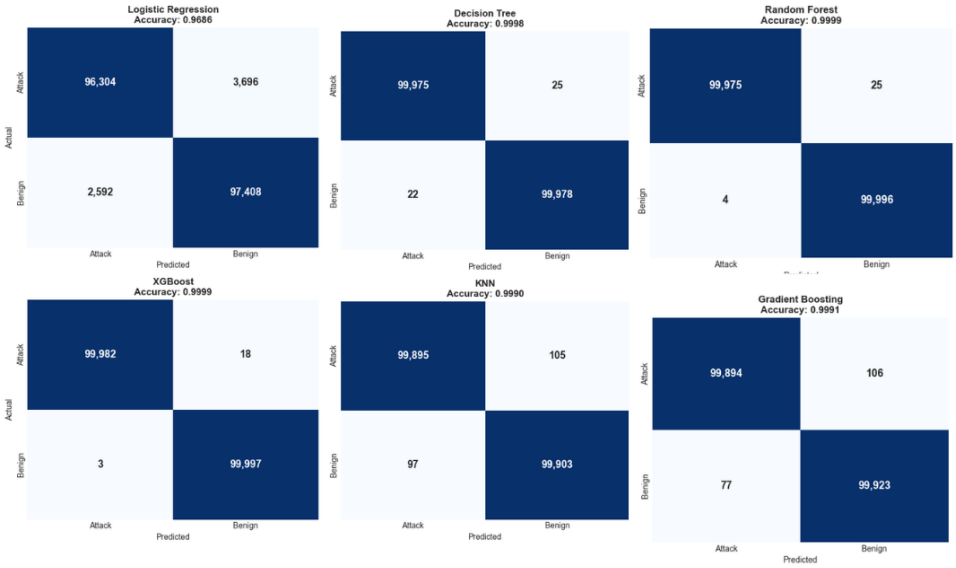
Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*	AUC	Training Time (s)
XGBoost	0,9999	0,9998	1,0000	0,9999	0,9999	5,34
Random Forest	0,9999	0,9998	1,0000	0,9999	0,9999	48,70
Decision Tree	0,9998	0,9998	0,9998	0,9998	0,9998	29,43
Gradient Boosting	0,9991	0,9989	0,9992	0,9991	0,9999	717,98
KNN	0,9990	0,9990	0,9990	0,9990	0,9997	230,92
Logistic Regression	0,9686	0,9634	0,9741	0,9687	0,9974	27,16

Berdasarkan Tabel 3, *XGBoost* dan *Random Forest* menempati peringkat teratas dengan *F1-Score* yang identik yaitu 0,9999. Namun, *XGBoost* menunjukkan keunggulan signifikan dari sisi efisiensi komputasi dengan waktu pelatihan hanya 5,34 detik, sekitar sembilan kali lebih cepat dibandingkan *Random Forest* (48,70 detik). Keunggulan kecepatan *XGBoost* ini sejalan dengan temuan Chen & Guestrin (2016) yang menunjukkan bahwa arsitektur *XGBoost* dirancang secara khusus untuk efisiensi komputasi melalui teknik *sparsity-aware* algorithm dan *cache-aware access*. *Decision Tree* menempati posisi ketiga dengan *F1-Score* 0,9998 dan waktu pelatihan 29,43 detik.

Gradient Boosting dan KNN memperoleh performa yang sedikit lebih rendah dengan *F1-Score* masing-masing 0,9991 dan 0,9990, namun keduanya memerlukan waktu pelatihan yang jauh lebih lama. *Logistic Regression* sebagai satu-satunya model linear menunjukkan performa terendah dengan *F1-Score* 0,9687, yang mengindikasikan bahwa batas keputusan linear kurang mampu memisahkan pola lalu lintas serangan dan normal secara optimal (Christodoulou et al., 2019). Hasil ini konsisten dengan penelitian Sahoo et al. (2020) yang juga menemukan bahwa algoritma ensemble berbasis pohon mengungguli model linear dalam deteksi DDoS.

3.2. Analisis *Confusion Matrix* & *Error Rate*

Untuk memahami lebih detail karakteristik kesalahan prediksi masing-masing model, dilakukan analisis *confusion matrix* terhadap seluruh algoritma (Halimu et al., 2019). Hasil analisis ditunjukkan pada Gambar 4.



Gambar 4. *Confusion Matrix* 6 Algoritma *Machine Learning*

Tabel 3. Analisis Error Rate Seluruh Algoritma

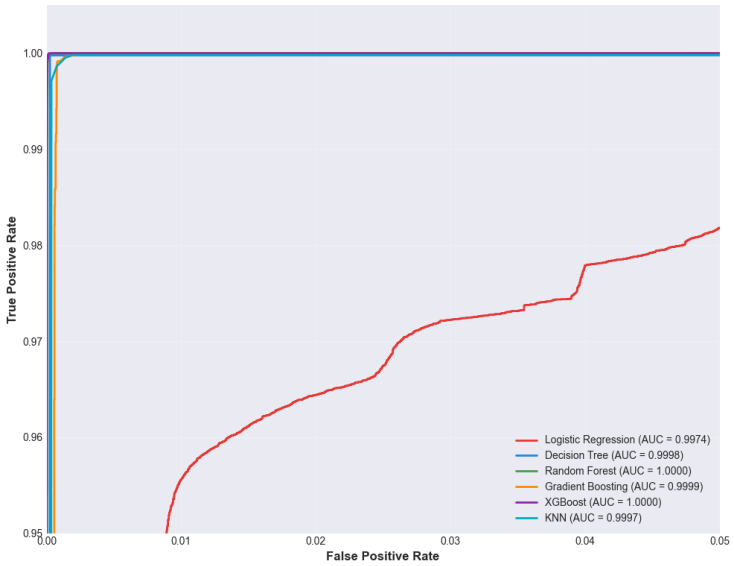
Model	*TP*	*TN*	*FP*	*FN*	Error Rate (%)
XGBoost	99.982	99.997	3	18	0,0105
Random Forest	99.975	99.996	4	25	0,0145
Decision Tree	99.978	99.975	25	22	0,0235
Gradient Boosting	99.894	99.923	77	106	0,0915
KNN	99.903	99.895	105	97	0,1010
Logistic Regression	96.304	97.408	2.592	3.696	3,1440

Dari Tabel 4 terlihat bahwa *XGBoost* menghasilkan jumlah kesalahan paling sedikit, yaitu hanya 3 False Positive dan 18 *False Negative* dari total 200.000 prediksi. Angka *False Positive* yang sangat rendah menunjukkan bahwa *XGBoost* hampir tidak pernah salah mengklasifikasikan lalu lintas normal sebagai serangan, sehingga meminimalkan potensi gangguan layanan akibat alarm palsu (Doriguzzi-Corin et al., 2020). *Random Forest* menunjukkan karakteristik serupa dengan *4 False Positive* dan 25 *False Negative*, yang juga tergolong sangat rendah. Temuan ini sejalan dengan Biau & Scornet (2016) yang menjelaskan bahwa mekanisme *bagging* pada *Random Forest* mampu mengurangi *variance* prediksi secara efektif.

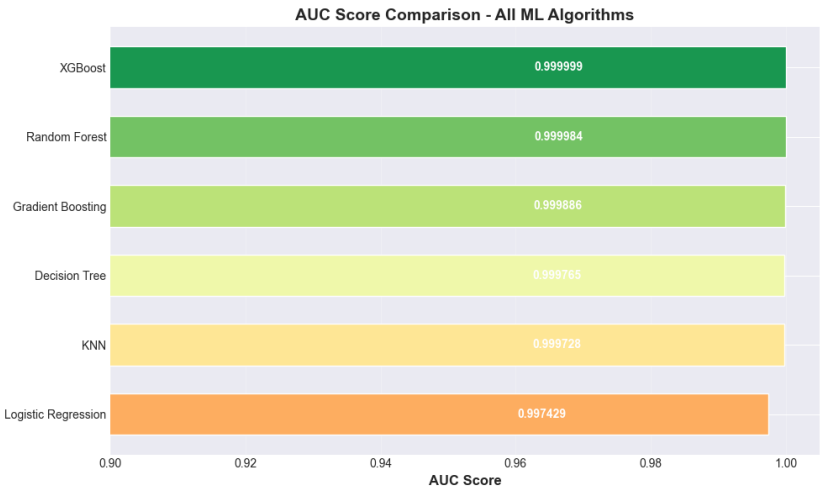
Perbedaan paling mencolok terlihat pada Logistic Regression yang mencatat 2.592 *False Positive* dan 3.696 *False Negative*. Tingginya angka *False Negative* pada model ini mengindikasikan bahwa hampir 3,7% serangan DDoS tidak terdeteksi, yang dalam konteks keamanan jaringan merupakan risiko yang serius karena serangan yang lolos dapat menyebabkan gangguan layanan secara langsung (Yan et al., 2016). Sebagaimana dikemukakan oleh Hussain et al. (2021), keterbatasan model linear dalam menangkap pola non-linear pada data jaringan merupakan faktor utama rendahnya performa *Logistic Regression*.

3.3. Analisis *ROC Curve* & AUC

Evaluasi lebih lanjut dilakukan menggunakan *ROC Curve* dan nilai AUC untuk mengukur kemampuan diskriminasi masing-masing model pada berbagai threshold klasifikasi. Hasil perbandingan nilai AUC ditunjukkan pada Tabel 5.



Gambar 5. *ROC Curve*



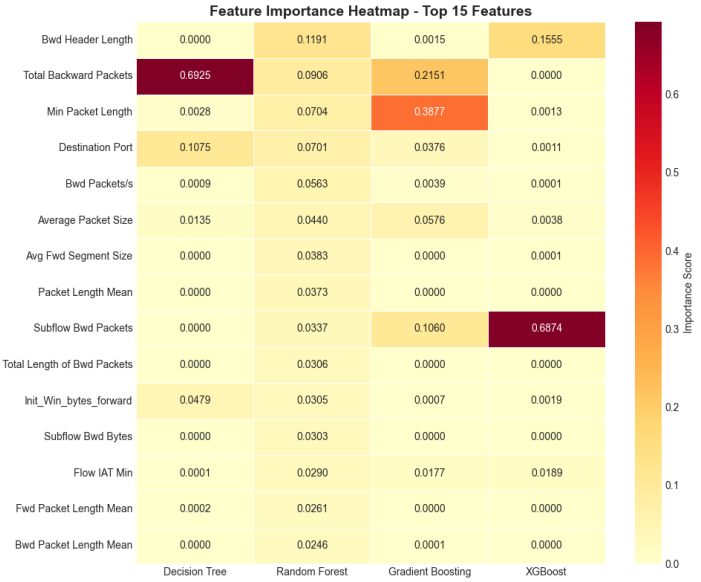
Gambar 6. *AUC Score*

XGBoost mencapai nilai *AUC* tertinggi yaitu 0,999999 yang mendekati sempurna, menunjukkan bahwa model ini mampu membedakan antara lalu lintas serangan dan normal dengan tingkat kepercayaan yang sangat tinggi pada hampir seluruh *threshold*. *Random Forest* dan *Gradient Boosting* menunjukkan nilai *AUC* yang sangat kompetitif, masing-masing 0,999984 dan 0,999982. Meskipun *Gradient Boosting* memiliki *AUC* sedikit lebih tinggi dibandingkan *Random Forest*, performa *F1-Score* dan *error rate*-nya justru lebih rendah sebagaimana ditunjukkan pada Tabel 3 dan 4, yang mengindikasikan bahwa *AUC* yang tinggi tidak selalu berkorelasi langsung dengan performa klasifikasi pada *threshold default*.

Logistic Regression kembali menempati posisi terendah dengan *AUC* 0,994553. Meskipun angka ini tergolong tinggi secara absolut, selisihnya cukup signifikan dibandingkan model lain, yang memperkuat temuan bahwa pendekatan linear memiliki keterbatasan dalam menangkap kompleksitas pola serangan *Distributed Denial of Service (DDoS)*.

3.4. Analisis Feature Importance

Analisis *feature importance* dilakukan pada empat model berbasis pohon keputusan, yaitu *Decision Tree*, *Random Forest*, *Gradient Boosting*, dan *XGBoost*. Tabel 6 menunjukkan lima belas fitur teratas berdasarkan skor *importance* dari model *Random Forest* sebagai referensi utama.



Dari Tabel 6 terlihat bahwa setiap model memiliki preferensi fitur yang berbeda. *Random Forest* menunjukkan distribusi *importance* yang paling merata di antara keempat model, dengan *Bwd Header Length* (0,1191) sebagai fitur tertinggi. Sebaliknya, *Decision Tree* sangat bergantung pada satu fitur dominan yaitu *Total Backward Packets* (0,6925), yang menunjukkan kecenderungan model pohon tunggal untuk terfokus pada fitur pemisah terkuat. *Gradient Boosting* menempatkan *Min Packet Length* (0,3877) sebagai fitur paling berpengaruh, sementara *XGBoost* sangat bergantung pada *Subflow Bwd Packets* (0,6874).

Meskipun terdapat perbedaan prioritas antar model, fitur-fitur yang berkaitan dengan karakteristik *backward traffic* secara konsisten muncul di seluruh model. Pola ini relevan dengan mekanisme serangan *Distributed Denial of Service (DDoS)* refleksi dalam dataset *CICDDoS2019*, di mana penyerang mengirim *request* berukuran kecil ke server reflektor yang kemudian menghasilkan *response* berukuran besar ke target, sehingga menciptakan anomali yang signifikan pada fitur-fitur *backward*.

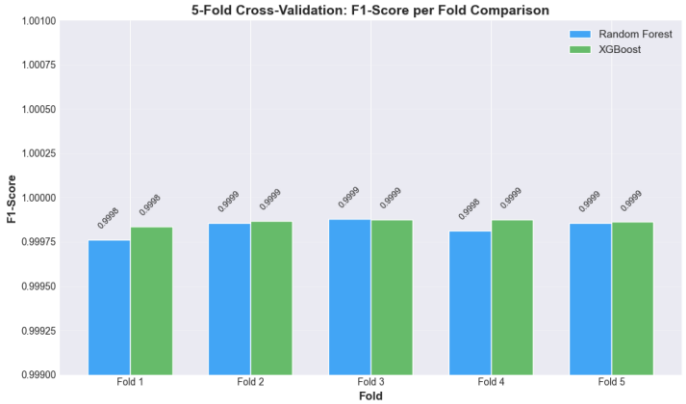
3.5. Validasi Model

Untuk memastikan reliabilitas hasil eksperimen, dilakukan empat tahap validasi terhadap dua model dengan performa terbaik, yaitu *Random Forest* dan *XGBoost*. Tahap pertama adalah pemeriksaan *overfitting* melalui perbandingan performa pada data training dan testing. Hasil perbandingan ditunjukkan pada Tabel 7.

Tabel 4. Perbandingan Performa Training dan Testing

Model	*Metrik*	*Training*	*Testing*	*Selisih*
Random Forest	Accuracy	1,0000	0,9999	0,0001
Random Forest	F1-Score	1,0000	0,9999	0,0001
XGBoost	Accuracy	1,0000	0,9999	0,0001
XGBoost	F1-Score	1,0000	0,9999	0,0001

Selisih antara performa *training* dan *testing* pada kedua model tidak melebihi 0,0002, yang berada jauh di bawah ambang batas 2%. Hasil ini mengindikasikan bahwa kedua model tidak mengalami *overfitting* dan mampu melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Tahap kedua adalah validasi silang menggunakan *5-fold cross-validation* pada data *training*. Hasil validasi ditunjukkan pada Tabel 8.



Gambar 7. Hasil 5-Fold Cross-Validation (F1-Score)

Nilai standar deviasi sebesar 0,0000 menunjukkan bahwa model memiliki stabilitas yang sangat tinggi di seluruh fold, tanpa adanya variasi performa yang berarti. Konsistensi ini mengonfirmasi bahwa hasil evaluasi tidak bergantung pada pembagian data tertentu.

4. Kesimpulan

Penelitian ini telah melakukan analisis perbandingan terhadap enam algoritma *machine learning* untuk deteksi serangan *Distributed Denial of Service (DDoS)* menggunakan dataset gabungan *CICDDoS2019* dan *CICIDS2017* dengan total 1.000.000 sampel seimbang. Hasil eksperimen menunjukkan bahwa *XGBoost* mencapai performa terbaik secara keseluruhan dengan *F1-Score* 0,9999, *AUC* 0,999999, dan waktu pelatihan tercepat yaitu 6,06 detik, diikuti oleh *Random Forest* dengan *F1-Score* 0,9999, *AUC* 0,999984, namun memerlukan waktu pelatihan 55,12 detik. *Logistic Regression* sebagai model linear menunjukkan performa terendah dengan *F1-Score* 0,9687, yang mengindikasikan bahwa pendekatan *non-linear* lebih sesuai untuk menangkap kompleksitas pola serangan *DDoS*.

Analisis *feature importance* mengungkapkan bahwa fitur-fitur yang berkaitan dengan *backward traffic* seperti *Bwd Header Length*, *Total Backward Packets*, dan *Subflow Bwd Packets* merupakan indikator paling dominan dalam membedakan lalu lintas serangan dan normal, yang konsisten dengan karakteristik serangan *DDoS* refleksi. Validasi melalui *5-fold cross-validation* mengonfirmasi stabilitas kedua model terbaik tanpa indikasi *overfitting* maupun *data leakage*.

Berdasarkan temuan tersebut, algoritma *ensemble* berbasis *boosting* khususnya *XGBoost* merupakan pendekatan yang paling efektif dan efisien untuk sistem deteksi serangan *DDoS*, dengan keunggulan pada akurasi prediksi yang sangat tinggi dan waktu komputasi yang minimal. Penelitian selanjutnya dapat diarahkan pada implementasi model pada perangkat *edge* seperti *Raspberry Pi* untuk deteksi *DDoS* secara *real-time* pada lingkungan *Internet of Things (IoT)*, serta eksplorasi teknik *feature selection* untuk mengurangi dimensi fitur tanpa mengorbankan performa

5. Ucapan terimakasih

Penulis mengucapkan terima kasih kepada Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur atas dukungan fasilitas dan bimbingan dalam pelaksanaan penelitian ini. Ucapan terima kasih juga ditujukan kepada *Canadian Institute for Cybersecurity* (CIC) yang menyediakan dataset CICDDoS2019 dan CICIDS2017 sebagai dasar dalam penelitian ini.

Daftar Pustaka

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Berrar, D. (2019). Cross-validation. In S. Ranganathan et al. (Eds.), *Encyclopedia of Bioinformatics and Computational Biology* (Vol. 1, pp. 542–545). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

Bhattacharyya, D. K., & Kalita, J. K. (2016). DDoS attacks: Evolution, detection, prevention, reaction, and tolerance. CRC Press.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, 25(2), 197–227. https://doi.org/10.1007/s11749-016-0481-7

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(1), 20–28. https://doi.org/10.38094/jastt20165

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Christodoulou, E., et al. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of Clinical Epidemiology*, 110, 12–22. https://doi.org/10.1016/j.jclinepi.2019.02.004

Cloudflare. (2024). DDoS threat report for 2024 Q1. Cloudflare.

Dong, S., Abbas, K., & Jain, R. (2020). A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. *IEEE Access*, 8, 80813–80828. https://doi.org/10.1109/ACCESS.2019.2936774

Doriguzzi-Corin, R., et al. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Transactions on Network and Service Management*, 17(2), 876–889. https://doi.org/10.1109/TNSM.2020.2971776

Emmanuel, T., et al. (2021). A survey on missing data in machine learning. *Journal of Big Data*, 8(1), 140. https://doi.org/10.1186/s40537-021-00516-9

García, S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, 98, 1–29. https://doi.org/10.1016/j.knosys.2015.12.006

Halimu, C., et al. (2019). Empirical comparison of AUC and MCC for evaluating machine learning algorithms on imbalanced datasets. *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, 1–6. https://doi.org/10.1145/3310986.3311023

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201

Hussain, F., et al. (2021). IoT DoS and DDoS attack detection using ResNet. *IEEE INMIC 2021 Proceedings*, 1–6. https://doi.org/10.1109/INMIC53986.2021.9642085

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Leevy, J. L., et al. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1), 42. https://doi.org/10.1186/s40537-018-0151-6

Li, Y., Liu, Q., & Sun, L. (2022). DDoS attack detection for IoT using XGBoost. *Computers & Security*, 118, 102731. https://doi.org/10.1016/j.cose.2022.102731

Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. *International Journal of Engineering and Technology*, 7(3.24), 479–482.

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in Python. *Information*, 11(4), 193. https://doi.org/10.3390/info11040193

Raschka, S. (2018). Model evaluation and selection in machine learning. *arXiv preprint arXiv:1811.12808*.

Sahoo, K. S., et al. (2020). An evolutionary SVM model for DDOS attack detection in SDN. *IEEE Access*, 8, 132502–132513. https://doi.org/10.1109/ACCESS.2020.3009733

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset. *ICISSP Proceedings*, 108–116.

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic DDoS attack dataset and taxonomy. *ICCST Proceedings*, 1–8.

Yan, Q., et al. (2016). SDN and DDoS attacks in cloud computing environments. *IEEE Communications Surveys & Tutorials*, 18(1), 602–622.

Zhang, Z. (2017). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, 5(11), 230.

Somani, G., et al. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. In: *Proceedings of the 11th International Conference on Cloud Computing (CLOUD)*. IEEE.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd ed. O'Reilly Media.

Alqahtani, S. A. (2020). *Machine Learning Approaches for DDoS Attack Detection in IoT Networks*. Master Thesis. King Saud University.

Canadian Institute for Cybersecurity. (2019). *CICDDoS2019 Dataset Documentation*. https://www.unb.ca/cic/datasets/ddos-2019.html