Edusaintek: Jurnal Pendidikan, Sains dan Teknologi
Volume 12 Issue 4 2025 Pages 1874 – 1893
p-ISSN: 1858-005X e-ISSN: 2655-3392 DOI: https://doi.org/10.47668/edusaintek.v12i3.1930
website: https://journalstkippgrisitubondo.ac.id/index.php/EDUSAINTEK

KLASIFIKASI SERANGAN *DISTRIBUTED DENIAL OF SERVICE* MENGGUNAKAN *ENSEMBLE STACKING*

Juan Parez Mangku Alamsyah[1*], Fayruz Rahma[2], Arrie Kurniawardhani[3]
[1,2,3]Universitas Islam Indonesia, Indonesia
*Corresponding author: juan.alamsyah@students.uii.ac.id

Abstract: Distributed Denial of Service (DDoS) attack is a type of cyberattack that aims to make network services or resources inaccessible to legitimate users by massively overwhelming network traffic. The increasingly complex and diverse patterns of DDoS attacks require detection systems that are not only reliable but also adaptive to various types of attacks. Most previous studies have been limited to binary classification, making them less effective in addressing the challenges of more diverse attack classification. This research aims to develop a machine learning-based Intrusion Detection System (IDS) using an ensemble learning approach for multiclass DDoS attack classification. The model was built using a stacking approach, with K-Nearest Neighbors, Decision Tree, Naive Bayes, and Support Vector Machine as base learners, and Logistic Regression as the meta learner. The CIC-DDoS2019 dataset was used as the data source for training and testing the model. The evaluation results show that the ensemble stacking model achieved the best performance with an accuracy of 78.8%, an F1-score of 78.4%, and the highest AUC value of 0.982. These results demonstrate that the ensemble approach can improve the performance and accuracy of DDoS attack detection systems in multiclass classification scenarios compared to individual models.

Keywords: Distributed Denial Of Service, Ensemble Learning, Intrusion Detection Systems, Machine Learning, Stacking

Abstrak: Serangan Distributed Denial of Service (DDoS) merupakan jenis serangan siber yang bertujuan untuk membuat layanan atau sumber daya jaringan tidak dapat diakses oleh pengguna yang sah dengan membanjiri lalu lintas jaringan secara masif. Pola serangan DDoS yang semakin kompleks dan bervariasi menuntut adanya sistem deteksi yang tidak hanya andal, tetapi juga adaptif terhadap berbagai jenis serangan. Sebagian besar penelitian sebelumnya masih terbatas pada klasifikasi biner sehingga kurang efektif dalam menghadapi tantangan klasifikasi serangan yang lebih beragam. Penelitian ini bertujuan untuk mengembangkan model Intrusion Detection System (IDS) berbasis machine learning dengan pendekatan ensemble learning untuk klasifikasi multiclass serangan DDoS. Model ini dibangun menggunakan pendekatan stacking, dengan K-Nearest Neighbors, Decision Tree, Naive Bayes, dan Support Vector Machine sebagai base learners, serta Logistic Regression sebagai meta learner. Dataset CIC-DDoS2019 digunakan sebagai sumber data untuk proses pelatihan dan pengujian model. Hasil evaluasi menunjukkan bahwa model ensemble stacking memberikan kinerja terbaik dengan accuracy sebesar 78,8%, F1-score sebesar 78,4%, dan nilai AUC tertinggi sebesar 0,982. Dengan demikian, pendekatan ensemble learning terbukti mampu meningkatkan kinerja dan keakuratan sistem deteksi serangan DDoS dalam skenario klasifikasi multiclass dibandingkan model individual.

Kata kunci: Distributed Denial Of Service, Ensemble Learning, Intrusion Detection Systems, Machine Learning, Stacking

PENDAHULUAN

Distributed Denial of Service (DDoS) merupakan salah satu serangan siber yang terjadi ketika banyak komputer secara bersamaan menghasilkan lalu lintas internet yang berlebihan untuk membanjiri sistem target sehingga menyebabkan sistem tersebut melambat secara drastis atau menjadi tidak responsif sama sekali (Butt et al., 2024). Sumber utama serangan DDoS adalah botnet, yaitu kumpulan perangkat yang dikompromikan (Adedeji, Abu-Mahfouz, & Kurien, 2023). Setelah botnet terbentuk, penyerang dapat mengendalikan serangan dengan mengirimkan perintah jarak jauh ke setiap bot, yang kemudian secara bersamaan mengirimkan permintaan ke server korban sehingga berpotensi membanjiri jaringan dan mengganggu lalu lintas yang sah (Adedeji et al., 2023).

Berdasarkan data Cloudflare ("Targeted by 20.5 million DDoS attacks, up 358% year-over-year: Cloudflare's 2025 Q1 DDoS Threat Report"), pada kuartal pertama tahun 2025, Cloudflare berhasil memblokir 20,5 juta serangan DDoS, yang mencerminkan peningkatan sebesar 198% dibandingkan kuartal sebelumnya. Sementara itu, menurut laporan NETSCOUT ("NETSCOUT DDoS Threat Intelligence Report - Latest Cyber Threat Intelligence Report"), jumlah serangan DDoS pada paruh kedua secara global mencapai 8,9 juta, yang mencerminkan kenaikan sebesar 12,75% dibandingkan total 7,9 juta serangan yang terjadi pada paruh pertama tahun 2024. Data dan laporan tersebut menunjukkan bahwa serangan DDoS terus mengalami eskalasi baik dari sisi frekuensi maupun intensitas sehingga dibutuhkan pendekatan mitigasi yang lebih adaptif dan cerdas.

Dalam merespon permasalahan tersebut, dikembangkan sebuah model Intrusion Detection System (IDS) berbasis *machine learning*. Model ini dirancang untuk membedakan antara lalu lintas jaringan normal dengan berbagai jenis serangan secara lebih efektif. Studi-studi terdahulu telah menunjukkan efektivitas model *machine learning* dalam membedakan antara lalu lintas jaringan yang normal dengan serangan DDoS (Aamir & Ali Zaidi, 2021; Bhayo et al., 2023; Munir, Ardiansyah, Santoso, Mustopa, & Mulyatun, 2022). Namun, sebagian besar studi tersebut hanya menggunakan algoritma individual dan menerapkan klasifikasi biner sehingga terbatas hanya pada pengenalan dua kelas, yaitu lalu lintas normal (*benign traffic*) dan lalu lintas berbahaya (*malicious traffic*). Dengan didasarkan pada pemahaman bahwa setiap model machine

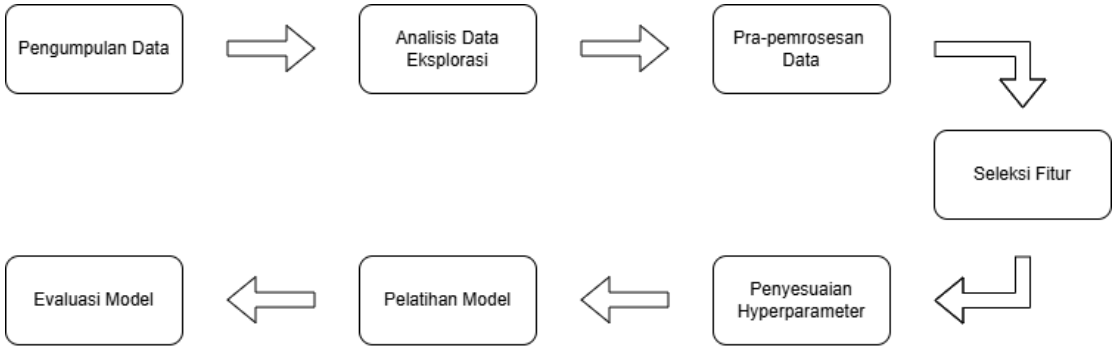
learning memiliki keterbatasan dan potensi kesalahan, penggunaan algoritma individual sering kali tidak cukup untuk menangani permasalahan deteksi serangan yang lebih kompleks (Alashhab et al., 2024; Mienye & Sun, 2022).

Untuk menjawab tantangan tersebut, pendekatan *ensemble learning* diadopsi sebagai solusi yang lebih adaptif dan komprehensif. *Ensemble learning* merupakan suatu metodologi dalam *machine learning* yang menggunakan berbagai metode pembelajaran, yang disebut sebagai *weak learners* atau *base models*, untuk meningkatkan kemampuan prediksi dari masing-masing algoritma pembelajaran yang digunakan (Comito & Pizzuti, 2022). *Ensemble learning* didasarkan pada pemahaman bahwa setiap model *machine learning* memiliki keterbatasan dan potensi kesalahan sehingga pendekatan ini dikembangkan untuk meningkatkan kinerja prediksi dengan memanfaatkan kekuatan dari berbagai model dasar guna saling mengompensasi kelemahan masing-masing model individual (Alashhab et al., 2024; Mienye & Sun, 2022).

Dalam penelitian ini, dikembangkan sebuah model *ensemble learning* untuk mendeteksi serangan DDoS pada skenario *multiclass classification*. Model ini menggunakan K-Nearest Neighbors, Naive Bayes, Decision Tree, dan Support Vector Machine sebagai *base learners*, sementara Logistic Regression digunakan sebagai *meta learner*. Berbagai kombinasi algoritma tersebut diuji untuk membentuk konfigurasi *ensemble learning* yang optimal, kemudian kinerjanya akan dibandingkan dengan model individual. Evaluasi dilakukan menggunakan metrik *accuracy, precision, recall, f1-score*, dan ROC-AUC untuk menilai sejauh mana pendekatan *ensemble learning* mampu meningkatkan kinerja prediksi dan ketahanan model dalam mengklasifikasikan berbagai jenis serangan DDoS dibandingkan dengan penggunaan model individual.

METODE

Rangkaian tahapan dalam pengembangan model *ensemble learning* yang diusulkan untuk mengidentifikasi serangan DDoS pada skenario *multiclass classification* ditunjukkan pada Gambar 1. Tahapan ini dirancang secara sistematis, mencakup tahapan mulai dari pengumpulan data, analisis data eksploratif, praproses data, pemilihan fitur, penyetelan *hyperparameter*, pelatihan model, hingga evaluasi akhir dari performa model.



Gambar 1. Tahapan Penelitian

Pengumpulan Data

Data dalam penelitian ini diperoleh dari situs resmi milik University of New Brunswick, melalui pusat riset Canadian Institute for Cybersecurity (CIC), yang menyediakan dataset CIC-DDoS2019 ("DDoS evaluation dataset (CIC-DDoS2019) | Datasets | Research | Canadian Institute for Cybersecurity | UNB"). Dataset ini dipilih karena secara khusus dirancang untuk mendukung penelitian dalam bidang deteksi serangan Distributed Denial of Service (DDoS), yang sesuai dengan tujuan utama penelitian ini, yaitu mengidentifikasi dan mengklasifikasikan berbagai jenis serangan DDoS.

Dalam penelitian ini, data yang digunakan hanya mengambil dari bagian *training day*, yang mencakup 12 jenis serangan DDoS, antara lain DrDoS_LDAP, DrDoS_UDP, DrDoS_NetBIOS, DrDoS_MSSQL, DrDoS_DNS, DrDoS_SSDP, DrDoS_SNMP, DrDoS_NTP, SYN, TFTP, UDP-lag, dan WebDDoS. Namun demikian, WebDDoS tidak disertakan dalam proses pemodelan karena volume lalu lintasnya sangat rendah jika dibandingkan dengan kelas atau label serangan lainnya sehingga dianggap tidak representatif untuk pelatihan model.

Analisis Data Eksplorasi

Analisis data eksplorasi dilakukan untuk memahami karakteristik dataset CIC-DDoS2019 serta mengidentifikasi pola, distribusi kelas, dan potensi masalah seperti *missing value*, nilai tak terdefinisi atau data yang terduplikasi. Dataset CIC-DDoS2019 yang digunakan mencakup 1 kelas lalu lintas normal (*benign*) dan 11 jenis serangan DDoS, yaitu DrDoS_LDAP, DrDoS_UDP, DrDoS_NetBIOS, DrDoS_MSSQL, DrDoS_DNS, DrDoS_SSDP, DrDoS_SNMP, DrDoS_NTP, SYN, TFTP, dan UDP-lag.

Selain itu, dilakukan analisis statistik deskriptif dan pemeriksaan kualitas data, di mana ditemukan adanya *missing value*, nilai tak terdefinisi, dan data yang terduplikasi.

Masalah ini akan ditangani pada tahap pra-pemrosesan data untuk memastikan data yang digunakan bersih, konsisten, dan siap untuk pelatihan model.

Pra-pemrosesan Data

Pra-pemrosesan data dilakukan untuk memastikan kualitas dan konsistensi data sebelum digunakan dalam proses pelatihan model klasifikasi. Tahapan ini sangat penting karena kualitas data memiliki pengaruh besar terhadap hasil dan kinerja model yang dibangun (Paranjape, Katta, & Ohlenforst, 2022). Proses ini mencakup pembersihan data dengan menghapus baris yang mengandung nilai kosong (*missing values*), data yang terduplikasi, dan nilai tak terdefinisi seperti *infinite* atau -*infinite*. Selain itu, fitur-fitur yang dianggap tidak relevan seperti Flow ID, Source IP, Destination IP, Source Port, Destination Port, Timestamp, SimillarHTTP, dan Fwd Header Length.1 dihilangkan dari dataset. Fitur-fitur tersebut tidak berkontribusi langsung terhadap karakteristik pola lalu lintas jaringan dan dapat menyebabkan bias jika digunakan dalam pelatihan. Pada penelitian (Moustafa, 2021) menyarankan penghapusan fitur alamat IP dan *port* untuk mencegah model bergantung pada informasi yang bersifat spesifik, serta agar model dapat mengenali pola serangan secara lebih umum dan menyeluruh.

Data yang berupa kategori dikonversi menjadi bentuk numerik menggunakan teknik *LabelEncoder* agar dapat dikenali oleh algoritma pembelajaran mesin. Teknik *LabelEncoder* dipilih karena efisien dan tidak menambah jumlah fitur secara signifikan. Proses standardisasi data juga diterapkan menggunakan *StandardScaler*. Teknik *StandardScaler* mengubah setiap fitur agar memiliki nilai rata-rata (*mean*) 0 dan standar *deviasi* 1 sehingga seluruh fitur berada pada skala yang sebanding dan memberikan kontribusi yang proporsional dalam proses pembelajaran model. Proses standardisasi data hanya dilakukan pada fitur yang aslinya sudah berbentuk angka, bukan fitur yang berupa kategori.

Seleksi Fitur

Seleksi fitur dilakukan untuk mengidentifikasi sekumpulan fitur yang paling berpengaruh dalam membangun model statistik, seperti untuk tugas klasifikasi atau regresi sehingga fitur-fitur yang tidak relevan maupun yang bersifat redundan dapat dihilangkan (Macedo, Valadas, Carrasquinha, Oliveira, & Pacheco, 2022). Seleksi fitur yang digunakan dalam penelitian ini adalah Mutual Information. Mutual Information digunakan untuk mengukur tingkat korelasi antara dua variabel (Cheng, Sun, Yao, Xu, &

Cao, 2022). Semakin besar nilai Mutual Information, semakin tinggi juga tingkat ketergantungan antara kedua variabel tersebut (Rafie, Moradi, & Ghaderzadeh, 2023). Variabel yang memiliki ketergantungan tinggi terhadap variabel target akan dipertahankan (Zhou, Wang, & Zhu, 2022).

Nilai Mutual Information antara dua variabel dapat dihitung menggunakan persamaan berikut:

$$I(X_i; Y) = H(X_i) - H(X_i|Y) \qquad (1)$$

Di mana, H(Xi) adalah entropi dari fitur Xi, dan H($Xi|y$) adalah entropi kondisional dari fitur Xi terhadap variabel target y.

Penyesuaian Hyperparameter

Hyperparameter adalah parameter eksternal yang ditentukan sebelum pelatihan model dan berperan dalam mengendalikan jalannya proses pembelajaran (Ogunsanya, Isichei, & Desai, 2023). Penyesuaian *hyperparameter* merupakan langkah penting karena dapat meningkatkan kinerja dari model. Dalam penelitian ini, metode *GridSearchCV* digunakan untuk mengeksplorasi kombinasi *hyperparameter* secara sistematis. *GridSearchCV* bekerja dengan mengevaluasi semua kombinasi yang mungkin dari nilai *hyperparameter* yang telah ditentukan sehingga memberikan peluang untuk memperoleh hasil yang optimal (Ogunsanya et al., 2023).

Algoritma Machine Learning

Di antara berbagai algoritma *supervised learning*, dipilih lima algoritma dasar dan umum digunakan dalam klasifikasi pemodelan *machine learning*, yaitu K-Nearest Neighbors, Naive Bayes, Logistic Regression, Decision Tree, dan Support Vector Machine

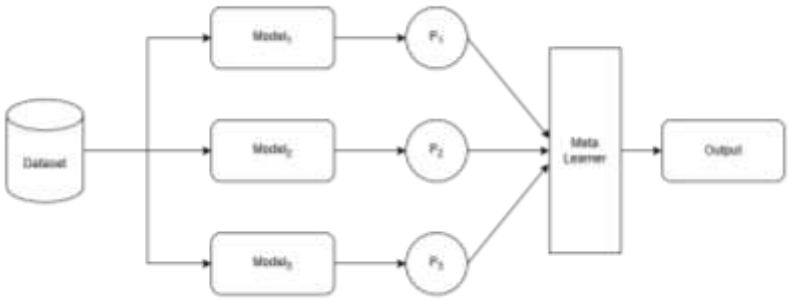
1. K-Nearest Neighbors (KNN): mengklasifikasi berdasarkan mayoritas kelas dari sejumlah tetangga terdekat, di mana jumlah tetangga ditentukan oleh nilai k (Fathima, Devi, & Faizaanuddin, 2023).
2. Naive Bayes (NB): algoritma klasifikasi statistik dan probabilistik yang mengasumsikan bahwa setiap fitur bersifat saling bebas secara mutlak sehingga keberadaan atau ketiadaan suatu fitur pada suatu kelas tidak dipengaruhi oleh fitur lainnya (Deepa, Sudar, & Deepalakshmi, 2019).
3. Logistic Regression (LR): algoritma yang memodelkan probabilitas keluaran biner (1 atau 0) untuk mengklasifikasikan observasi ke dalam salah satu dari dua kelompok

yang berbeda (Fathima et al., 2023). Logistic Regression dapat dikembangkan menjadi Multinomial Logistic Regression, yang dapat digunakan untuk menyelesaikan permasalahan klasifikasi *multiclass* dengan memanfaatkan fungsi *softmax* (Kamble & Dale, 2022).

4. Decision Tree (DT): membangun model berbentuk pohon keputusan untuk memprediksi nilai target dengan mempelajari aturan-aturan keputusan sederhana dari fitur data (Mihoub, Fredj, Cheikhrouhou, Derhab, & Krichen, 2022).
5. Support Vector Machine (SVM): memetakan data ke ruang fitur berdimensi lebih tinggi menggunakan fungsi kernel, lalu mencari *hyperplane* optimal untuk klasifikasi secara efektif (Liu et al., 2024).

Ensemble Learning

Pendekatan *ensemble learning* yang digunakan dalam penelitian ini adalah *stacking*. *Stacking* menggabungkan beberapa model dasar (*base models*) untuk meningkatkan akurasi dan ketangguhan (*robustness*) dari hasil prediksi (Arya, Kumar, & Ahmad, 2023). Salah satu keunggulan *stacking* adalah kemampuannya untuk menggunakan model-model yang heterogen sebagai *base learner*, di mana *output* dari beberapa model dasar (*level-0 learners*) digunakan sebagai *input* bagi *meta-learner* (*level-1 learner*) untuk menghasilkan prediksi akhir yang lebih akurat (Lazzarini, Tianfield, & Charissis, 2023). Kekurangan dari *ensemble stacking* yaitu biaya komputasi dapat menjadi tinggi, terutama apabila model-model dasar yang digunakan memiliki kompleksitas tinggi atau data yang digunakan memiliki dimensi yang besar (Arya et al., 2023). Ilustrasi arsitektur *ensemble learning stacking* secara umum ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur *Ensemble Learning Stacking*

Dalam penelitian ini, algoritma yang digunakan sebagai *base learner* adalah K-Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machine (SVM), dan Decision Tree (DT). Sementara itu, algoritma yang digunakan sebagai *meta-learner* adalah Logistic Regression. Pemilihan Logistic Regression sebagai *meta-learner* didasarkan pada kemampuannya menghasilkan model yang sederhana, mudah diinterpretasikan, serta mampu memberikan output dalam bentuk probabilitas yang informatif (Taha & Malebary, 2022).

Evaluasi Model

Untuk mengukur performa model yang dikembangkan, digunakan beberapa metrik evaluasi seperti accuracy, precision, recall, f1-score, confusion matrix, Receiver Operating Characteristic (ROC), dan Area Under the Curve (AUC). Penggunaan berbagai metrik ini bertujuan untuk memberikan gambaran yang komprehensif mengenai kinerja model, baik dari segi ketepatan prediksi maupun kemampuan dalam membedakan antar kelas.

Dasar dari perhitungan metrik-metrik tersebut mengacu pada empat komponen utama dalam hasil klasifikasi, yaitu:

a) *True Positive* (TP): Jumlah data yang benar-benar termasuk dalam suatu kelas dan berhasil diprediksi sebagai kelas tersebut oleh model.
b) *True Negative* (TN): Jumlah data yang berasal dari kelas lain dan berhasil diprediksi bukan sebagai kelas tersebut.
c) *False Positive* (FP): Jumlah data yang berasal dari kelas lain, tetapi salah diprediksi sebagai kelas tersebut oleh model.
d) *False Negative* (FN): Jumlah data yang termasuk dalam suatu kelas, tetapi salah diprediksi sebagai kelas lain oleh model.

1. *Accuracy*, *Precision*, *Recall*, dan *F1-Score*

a) *Accuracy*: Mengukur seberapa banyak prediksi model yang benar dibandingkan dengan seluruh data uji.

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \qquad (2)$$

b) *Precision*: Menunjukkan seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dibuat.

$$P = \frac{TP}{TP+FP} \qquad (3)$$

c) *Recall*: Menunjukkan seberapa banyak data positif yang berhasil dikenali dengan benar oleh model.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

d) *F1-score*: Rata-rata harmonis dari *precision* dan *recall*, yang digunakan untuk menilai keseimbangan antara keduanya.

$$F1\ Score = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (5)$$

2. *Confusion matrix:* performa model dalam bentuk tabel yang menunjukkan jumlah prediksi benar dan salah untuk masing-masing kelas, membantu dalam mengidentifikasi jenis kesalahan klasifikasi yang paling sering terjadi.
3. *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC):
   a) *Receiver Operating Characteristic* (ROC): Representasi visual dari performa model pada berbagai *threshold*.
   b) *Area Under the Curve* (AUC): nilai numerik dari luas di bawah kurva ROC. Semakin mendekati 1, semakin baik performa model dalam membedakan antara kelas positif dan negatif.

Pendekatan ROC-AUC yang digunakan adalah *macro average One-vs-Rest (OvR)*, di mana setiap kelas diperlakukan sebagai kelas positif dan sisanya sebagai negatif, kemudian nilai AUC dihitung untuk setiap kelas dan dirata-ratakan secara makro. Pendekatan ini memungkinkan evaluasi yang adil pada distribusi data yang seimbang, sehingga performa model dapat diukur secara komprehensif pada seluruh kelas.

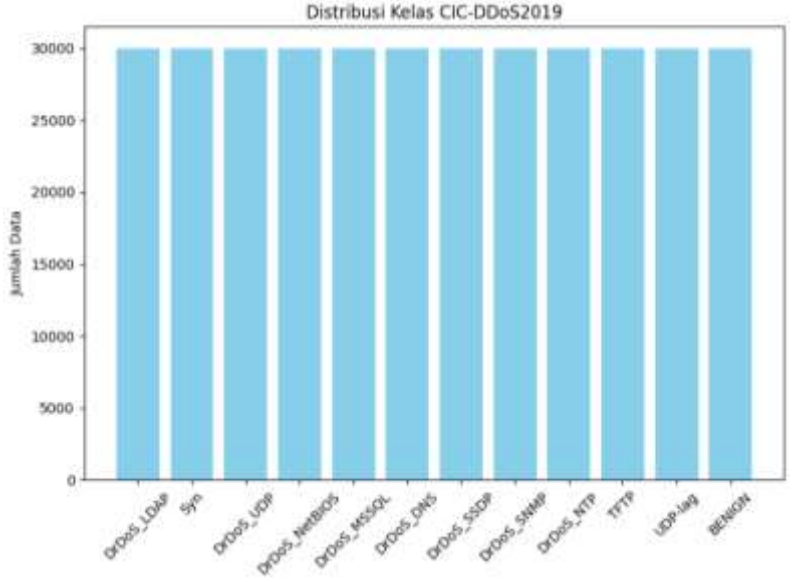
HASIL DAN PEMBAHASAN

Hasil

Pra-pemrosesan Data

Data yang digunakan dalam penelitian ini berasal dari dataset CIC-DDoS2019 yang dikembangkan oleh Canadian Institute for Cybersecurity. Secara keseluruhan, dataset ini memiliki total sebanyak 50.063.112 entri data. Setelah dilakukan proses pembersihan, yaitu penghapusan data yang mengandung *missing value*, nilai tidak valid (seperti *infinite* atau *-infinite*), serta data yang terduplikasi, dilakukan *random sampling* untuk memperoleh distribusi kelas yang seimbang. Hasil akhir dari proses ini adalah sebanyak 360.000 data yang digunakan dalam tahap pemodelan, baik pelatihan maupun pengujian,

dengan masing-masing kelas terdiri atas 30.000 data. Pemilihan jumlah data ini didasarkan pada pertimbangan keterbatasan sumber daya komputasi sehingga diperlukan kompromi antara ukuran data dan efisiensi pemrosesan. Distribusi data untuk masing-masing kelas ditampilkan pada Gambar 3, yang mencakup satu kelas normal (*benign*) dan sebelas jenis serangan DDoS.



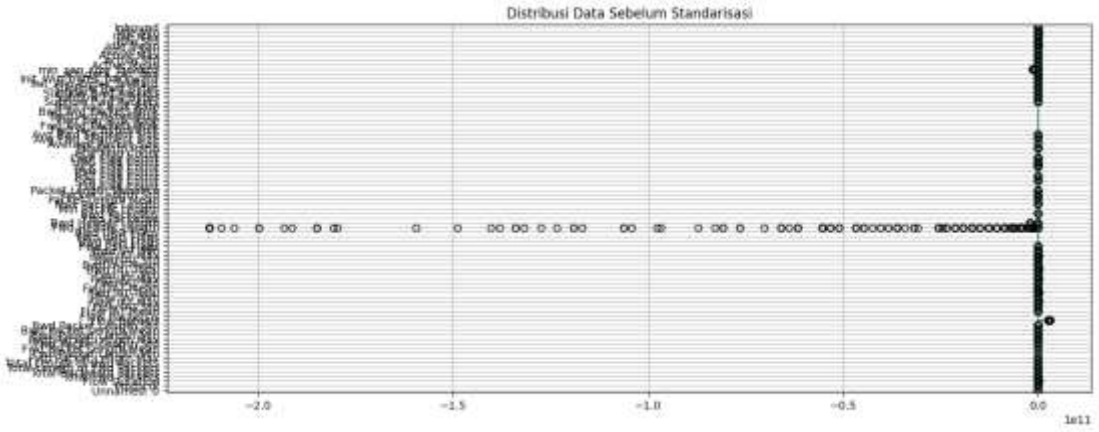
Gambar 3. Distribusi Kelas CIC-DDoS2019

Dilakukan proses *encoding* terhadap label kelas pada dataset CIC-DDoS2019 menggunakan metode *LabelEncoder*. Proses ini bertujuan untuk mengubah label berbentuk teks menjadi nilai numerik agar dapat diproses oleh algoritma *machine learning*. Pemetaan antara label asli dan nilai numerik hasil *encoding* disajikan pada Tabel 1.

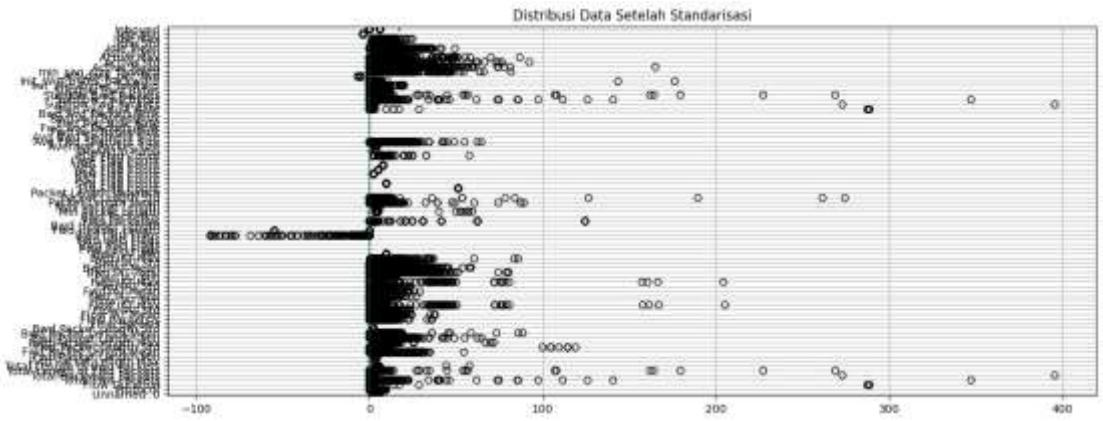
Tabel 1. Pemetaan Label Asli ke Dalam Nilai Numerik

Label Asli	Nilai enkode
BENIGN	0
DrDoS_DNS	1
DrDoS_ LDAP	2
DrDoS_MSSQL	3
DrDoS_NTP	4
DrDoS_NetBIOS	5
DrDoS_SNMP	6
DrDoS_SSDP	7
DrDoS_UDP	8
SYN	9
TFTP	10
UDP-lag	11

Proses standardisasi data dilakukan menggunakan metode *StandardScaler* setelah dilakukan pembagian data menjadi data latih dan data uji. Distribusi data sebelum dan sesudah standardisasi dapat dilihat pada Gambar 4 dan Gambar 5. Pada Gambar 4 terlihat bahwa data memiliki skala yang sangat bervariasi antar fitur. Beberapa fitur memiliki rentang nilai yang jauh lebih besar dibandingkan fitur lainnya sehingga persebaran data tampak terkonsentrasi di sekitar titik tertentu. Gambar 5 memperlihatkan hasil distribusi data setelah standardisasi, di mana seluruh fitur telah berada pada skala yang relatif seragam dengan rata-rata mendekati nol.



Gambar 4. Distribusi Data Sebelum Proses Standardisasi

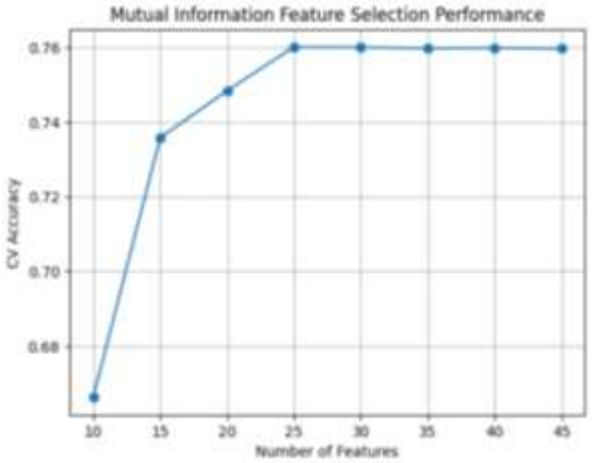


Gambar 5. Distribusi Data Setelah Proses Standardisasi

Seleksi Fitur dengan Mutual Information

Proses seleksi fitur menggunakan metode Mutual Information menghasilkan sejumlah fitur yang memiliki tingkat relevansi tinggi terhadap variabel target. Untuk menentukan jumlah fitur yang optimal, dilakukan evaluasi menggunakan *3-fold cross-validation* dengan model Random Forest terhadap berbagai jumlah fitur. Berdasarkan hasil yang ditampilkan pada Gambar 6, akurasi model memang cenderung meningkat seiring bertambahnya jumlah fitur, tetapi peningkatan tersebut menjadi sangat kecil

setelah melewati angka 25 fitur. Oleh karena itu, ditentukan sebanyak 25 fitur sebagai jumlah optimal untuk menjaga efisiensi tanpa mengorbankan performa model secara signifikan.



Gambar 6. Mutual Information dengan *3-fold Cross-Validation*

Sebanyak 25 fitur yang dipilih melalui proses seleksi tersebut merupakan fitur-fitur dengan nilai Mutual Information tertinggi terhadap variabel target. Fitur-fitur ini dianggap paling informatif dalam membedakan lalu lintas normal dan berbagai jenis serangan DDoS. Berikut daftar 25 fitur terpilih yang digunakan dalam proses pelatihan model disajikan pada Tabel 2 berikut.

Tabel 2. Daftar Fitur Hasil Mutual Information

No.	Fitur	No.	Fitur	No.	Fitur
1.	Packet Length Mean	11.	Min Packet Length	21.	Fwd IAT Mean
2.	Max Packet Length	12.	Flow IAT Mean	22.	Fwd IAT Max
3.	Average Packet Size	13.	Flow Packets/s	23.	Fwd Header Length
4.	Fwd Packet Length Mean	14.	Fwd Packets/s	24.	Fwd IAT Std
5.	Avg Fwd Segment Size	15.	Flow Duration	25.	Init_Win_bytes_forward
6.	Fwd Packet Length Max	16.	act_data_pkt_fwd		
7.	Subflow Fwd Bytes	17.	Flow IAT Max		
8.	Total Length of Fwd Packets	18.	Unnamed: 0		
9.	Fwd Packet Length Min	19.	Fwd IAT Total		
10.	Flow Bytes/s	20.	Flow IAT Std		

Penyesuaian Hyperparameter dengan GridSearchCV

Untuk menentukan nilai *hyperparameter* optimal, dilakukan proses iteratif menggunakan pendekatan *grid search*. Kombinasi nilai *hyperparameter* didefinisikan secara eksplisit dalam ruang pencarian, seperti yang terlihat pada Tabel 3. Selanjutnya,

proses pencarian dilakukan menggunakan *GridSearchCV* dengan *3-fold cross validation* dan metrik evaluasi berupa akurasi. Kombinasi nilai *hyperparameter* terbaik kemudian dipilih berdasarkan nilai akurasi tertinggi yang diperoleh selama proses validasi.

Tabel 3. *Hyperparameter* yang Diuji

Model	Parameter
KNN	n_neighbors = range(1, 40)
NB	var_smoothing = np.logspace(0, -9, num = 100)
LR	solver = [‘saga’, ‘lbfgs’, ‘newton-cg’] penalty = [‘l1’, ‘l2’, ‘elasticnet’] C = [0.01, 0.1, 1, 10]
DT	criterion = ['gini', 'entropy', 'log_loss'] max_depth = [None, 5, 10, 20, 30] min_samples_split = [2, 5, 10] min_samples_leaf = [1, 2, 4] max_features = [None, 'sqrt', 'log2']
SVM	kernel = ['linear', 'rbf', 'poly'] C = [0.1, 0.5, 1, 10] gamma = ['scale', 'auto']

Hasil kombinasi *hyperparameter* terbaik yang diperoleh selama proses validasi dapat dilihat pada Tabel 4. Nilai-nilai tersebut dipilih karena menghasilkan angka akurasi tertinggi dan digunakan sebagai konfigurasi akhir dalam pelatihan model untuk evaluasi selanjutnya.

Tabel 4. Nilai *Hyperparameter* Optimal

Model	Parameter	Akurasi
KNN	n_neighbors = 17	76%
NB	var_smoothing = np.float64(1e-06)	50%
LR	solver = 'newton-cg' penalty = 'l2' C = 10,	65%
DT	criterion = 'entropy' max_depth = 10 max_features = None min_samples_leaf = 2 min_samples_split = 2	78%
SVM	kernel = 'rbf' C = 10 gamma = 'scale'	73%

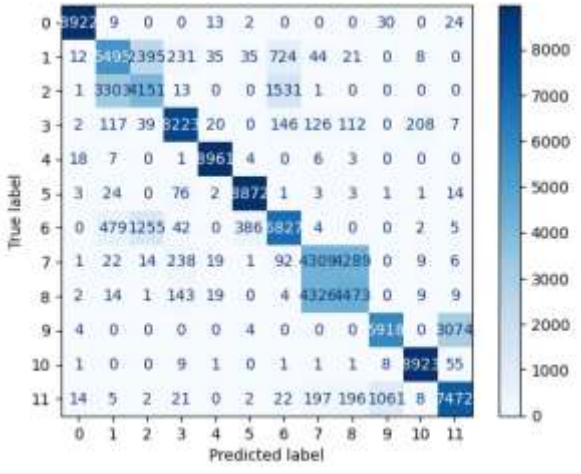
Evaluasi Model

Dataset yang telah melalui proses pra-pemrosesan data kemudian dibagi menjadi dua bagian, yaitu 70% data latih dan 30% data uji. Setiap algoritma individual yang digunakan dalam penelitian ini akan dievaluasi kinerja prediksinya sebelum dibandingkan

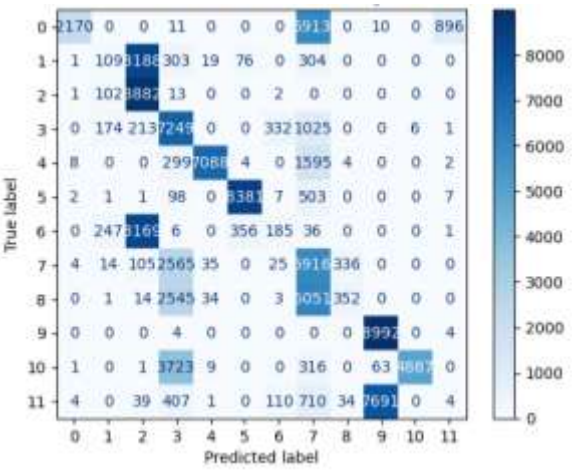
dengan performa model *ensemble learning*. Dalam penelitian ini, digunakan skenario klasifikasi *multiclass* sehingga model tidak hanya membedakan antara dua kelas, tetapi juga mengklasifikasikan berbagai jenis serangan secara lebih spesifik.

1. Confusion Matrix

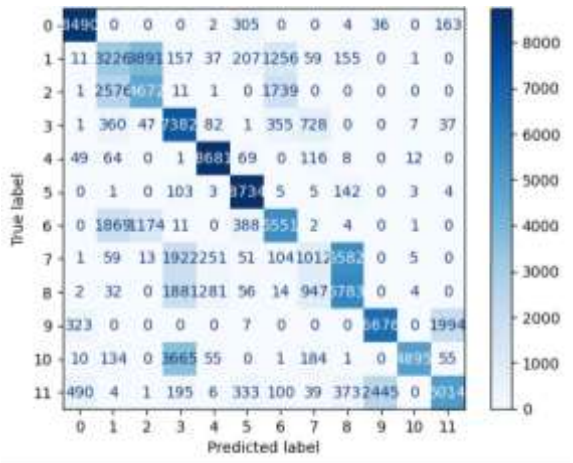
   Dilakukan visualisasi hasil prediksi dalam bentuk *confusion matrix*. Visualisasi ini membantu dalam mengidentifikasi pola keberhasilan maupun kesalahan klasifikasi pada setiap kelas secara lebih rinci. Gambar 7 menampilkan *confusion matrix* dari model K-Nearest Neighbors (KNN). Gambar 8 menyajikan *confusion matrix* untuk model Naive Bayes. Gambar 9 menunjukkan *confusion matrix* untuk model Logistic Regression. Gambar 10 menunjukkan *confusion matrix* untuk model Decision Tree. Gambar 11 menunjukkan *confusion matrix* untuk model Decision Tree. Gambar 11 menunjukkan *confusion matrix* untuk model *ensemble stacking*.



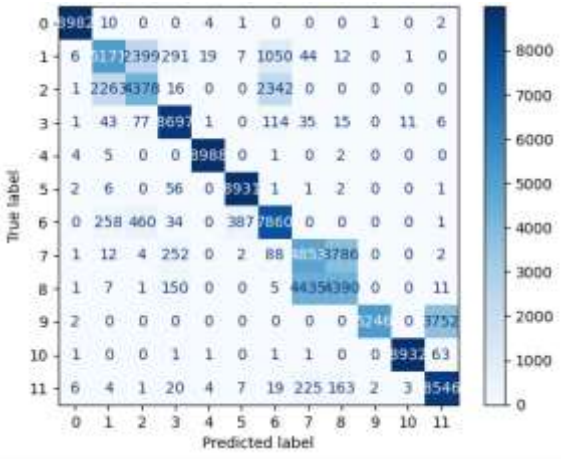
Gambar 7. *Confusion Matrix* Model KNN



Gambar 8. *Confusion Matrix* Model Naive Bayes

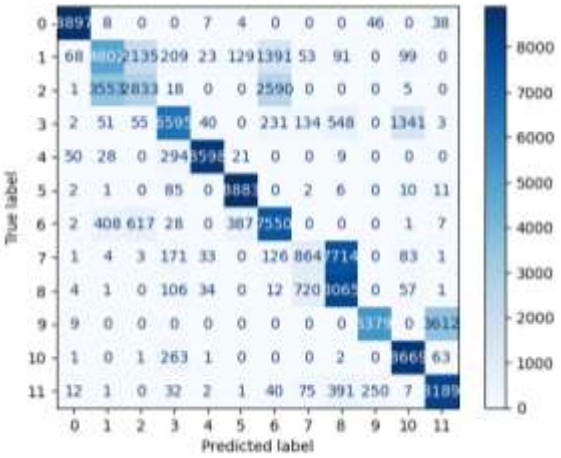


Gambar 9. *Confusion Matrix* Model Logistic



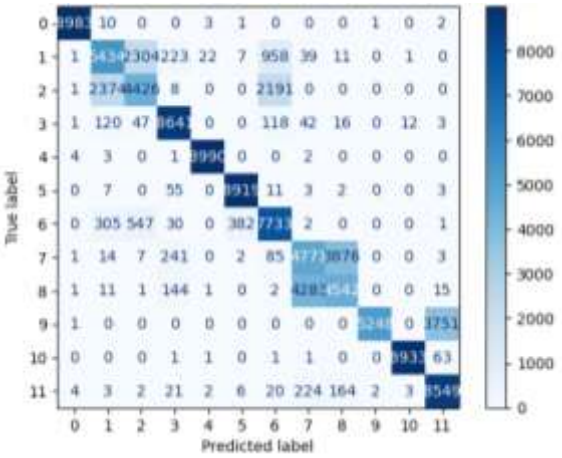
Gambar 10. *Confusion Matrix* Model

Regression



Gambar 9. *Confusion Matrix* Support Vector Machine

Decision Tree



Gambar 10. *Confusion Matrix Ensemble Stacking*

2. Accuracy, Precision, Recall, dan F1-Score

Tabel 5 menyajikan skor evaluasi dari setiap model yang diuji dalam klasifikasi *multiclass*. Model Decision Tree memberikan hasil kinerja terbaik di antara model individual lainnya, dengan *accuracy* mencapai 78,6% dan *F1-score* sebesar 78,2%. Sebaliknya, model Naive Bayes menunjukkan performa paling rendah, dengan *Accuracy* sebesar 50,1% *F1-score* hanya 43,1%, yang mengindikasikan ketidaksesuaian model tersebut terhadap karakteristik data.

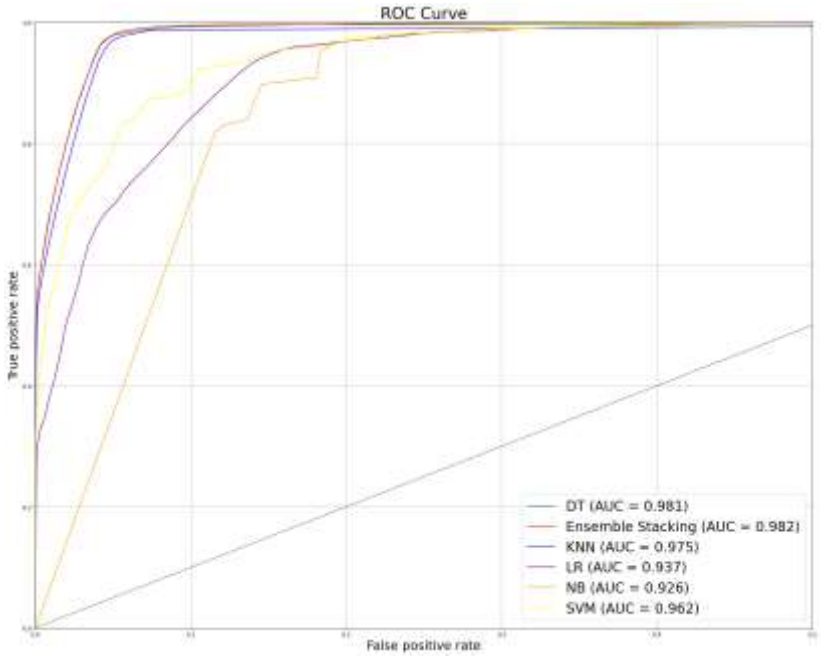
Metode *ensemble stacking* berhasil memberikan hasil prediksi yang sedikit lebih unggul dibandingkan model individual, dengan *accuracy* 78,8%, *precision* 79,6%, *recall* 78,8%, dan *F1-score* 78,4%. Walaupun peningkatannya tergolong tipis dibandingkan Decision Tree, model *ensemble stacking* tetap menunjukkan keunggulan konsisten dalam seluruh metrik evaluasi. Hasil ini mengindikasikan bahwa pendekatan *ensemble* mampu meningkatkan stabilitas dan akurasi prediksi, meskipun tidak secara signifikan.

Tabel 5. Kinerja Model Klasifikasi *Multiclass*

Model	Accuracy	Precision	Recall	F1-Score
KNN	76,4%	76,4%	76,4%	76,2%
NB	50,1%	53,5%	50,1%	43,1%
LR	64,9%	65,5%	64,9%	63,7%
DT	78,6%	79,4%	78,6%	78,2%
SVM	73,4%	73,8%	73,4%	71,2%
Ensemble Stacking	**78,8%**	**79,6%**	**78,8%**	**78,4%**

3. ROC-AUC

Gambar 13 menampilkan hasil evaluasi ROC-AUC dengan pendekatan *macro-average One-vs-Rest* (OvR) dari lima model individual dan satu model *ensemble stacking* yang digunakan dalam penelitian. Model *ensemble stacking* berhasil memperoleh nilai AUC tertinggi sebesar 0,982, mengungguli seluruh model individual. Model Decision Tree berada di posisi kedua dengan AUC sebesar 0,981, diikuti oleh K-Nearest Neighbors (KNN) dengan AUC 0,975, Support Vector Machine (SVM) sebesar 0,962, Logistic Regression sebesar 0,937, dan Naive Bayes sebesar 0,926.



Gambar 13. Kurva ROC Setiap Model

Tabel 6 merangkum nilai AUC dari semua model yang diuji, baik model individual maupun *ensemble stacking*. Data dalam tabel tersebut mempertegas keunggulan model Decision Tree di antara algoritma individual, sekaligus menegaskan bahwa pendekatan *ensemble stacking* memberikan hasil terbaik secara keseluruhan.

Tabel 6. Nilai AUC Model Individual

Model	AUC
KNN	0,975
NB	0,926
LR	0,937
DT	0,981
SVM	0,962
Ensemble Stacking	**0,982**

Pembahasan

Hasil evaluasi menunjukkan bahwa di antara model individual, Decision Tree menunjukkan performa terbaik dengan *accuracy* sebesar 78,6% dan *F1-score* sebesar 78,2%. Sementara itu, Naive Bayes menjadi model dengan performa terendah, mengindikasikan keterbatasannya dalam menangani karakteristik kompleks dari dataset yang digunakan.

Model *ensemble stacking* yang dikembangkan dengan memanfaatkan empat algoritma dasar—K-Nearest Neighbors, Decision Tree, Naive Bayes, dan Support Vector Machine sebagai model dasar (*base learner*)—dan menggunakan Logistic Regression sebagai *meta-learner*, menunjukkan hasil yang paling stabil dan kompetitif. Meskipun peningkatan kinerjanya terhadap model individual terbaik tergolong tipis, *ensemble stacking* tetap unggul dalam semua metrik evaluasi dengan *accuracy* 78,8%, *precision* 79,6%, *recall* 78,8%, dan *F1-score* 78,4%.

Dari evaluasi ROC-AUC dengan menggunakan pendekatan *macro-average One-vs-Rest* (OvR), hasil kurva ROC mengindikasikan bahwa seluruh model memiliki kemampuan diskriminatif yang tinggi, dengan Decision Tree mencatatkan AUC tertinggi di antara model individual yaitu sebesar 0,981. Namun demikian, model *ensemble stacking* tetap menjadi yang terbaik dengan AUC sebesar 0,982.

Secara keseluruhan, pendekatan *ensemble learning* terbukti efektif dalam meningkatkan performa sistem deteksi intrusi berbasis *machine learning*, terutama dalam konteks klasifikasi *multiclass* serangan DDoS. Pendekatan ini memberikan ketahanan klasifikasi yang lebih stabil, akurasi lebih tinggi, serta kemampuan generalisasi yang lebih baik dibandingkan dengan penggunaan model individual.

SIMPULAN

Penelitian ini telah mengembangkan sebuah model *Intrusion Detection System* (IDS) berbasis *machine learning* untuk mendeteksi serangan DDoS dalam skenario klasifikasi *multiclass*. Penelitian ini mengadopsi pendekatan *ensemble learning* guna meningkatkan akurasi dan ketahanan klasifikasi terhadap berbagai jenis serangan.

Model *ensemble stacking* yang mengombinasikan prediksi dari beberapa algoritma (*base learners*) dan diproses melalui Logistic Regression sebagai *meta learner* terbukti efektif dalam meningkatkan performa secara keseluruhan. Model *ensemble stacking* ini memperoleh *accuracy* 78,8% dan AUC 0,982, dengan performa yang stabil dan konsisten

pada semua metrik evaluasi (*precision*, *recall*, *F1-score*).

Secara keseluruhan, pendekatan *ensemble learning* terbukti efektif dalam meningkatkan performa sistem deteksi intrusi berbasis *machine learning*, terutama dalam konteks klasifikasi *multiclass* serangan DDoS. Pendekatan ini memberikan ketahanan klasifikasi yang lebih stabil, akurasi lebih tinggi, serta kemampuan generalisasi yang lebih baik dibandingkan dengan penggunaan model individual.

DAFTAR PUSTAKA

Aamir, M., & Ali Zaidi, S. M. (2021). Clustering based semi-supervised machine learning for DDoS attack classification. *Journal of King Saud University - Computer and Information Sciences*, *33*(4), 436–446. https://doi.org/10.1016/J.JKSUCI.2019.02.003

Adedeji, K. B., Abu-Mahfouz, A. M., & Kurien, A. M. (2023). DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges. *Journal of Sensor and Actuator Networks 2023, Vol. 12, Page 51*, *12*(4), 51. https://doi.org/10.3390/JSAN12040051

Alashhab, A. A., Zahid, M. S., Isyaku, B., Elnour, A. A., Nagmeldin, W., Abdelmaboud, A., … Maiwada, U. D. (2024). Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model. *IEEE Access*, *12*, 51630–51649. https://doi.org/10.1109/ACCESS.2024.3384398

Arya, A., Kumar, A., & Ahmad, S. S. (2023). DDoS Attack Detection Using Ensemble Machine Learning Approach. *2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*. https://doi.org/10.1109/ICCCNT56998.2023.10306750

Bhayo, J., Shah, S. A., Hameed, S., Ahmed, A., Nasir, J., & Draheim, D. (2023). Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. *Engineering Applications of Artificial Intelligence*, *123*, 106432. https://doi.org/10.1016/J.ENGAPPAI.2023.106432

Butt, H. A., Harthy, K. S. Al, Shah, M. A., Hussain, M., Amin, R., & Rehman, M. U. (2024). Enhanced DDoS Detection Using Advanced Machine Learning and Ensemble Techniques in Software Defined Networking. *Computers, Materials and Continua*, *81*(2), 3003–3031. https://doi.org/10.32604/CMC.2024.057185

Cheng, J., Sun, J., Yao, K., Xu, M., & Cao, Y. (2022). A variable selection method based on mutual information and variance inflation factor. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, *268*, 120652. https://doi.org/10.1016/J.SAA.2021.120652

Comito, C., & Pizzuti, C. (2022). Artificial intelligence for forecasting and diagnosing COVID-19 pandemic: A focused review. *Artificial Intelligence in Medicine*, *128*, 102286. https://doi.org/10.1016/J.ARTMED.2022.102286

DDoS evaluation dataset (CIC-DDoS2019) | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Retrieved 06/19/2025 from https://www.unb.ca/cic/datasets/ddos-2019.html

Deepa, V., Sudar, K. M., & Deepalakshmi, P. (2019). Design of Ensemble Learning Methods for DDoS Detection in SDN Environment. *Proceedings - International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN 2019*. https://doi.org/10.1109/VITECON.2019.8899682

Fathima, A., Devi, G. S., & Faizaanuddin, M. (2023). Improving distributed denial of service attack detection using supervised machine learning. *Measurement: Sensors*, *30*, 100911. https://doi.org/10.1016/J.MEASEN.2023.100911

Kamble, V. H., & Dale, M. P. (2022). Machine learning approach for longitudinal face recognition of children. *Machine Learning for Biometrics: Concepts, Algorithms and Applications*, 1–27. https://doi.org/10.1016/B978-0-323-85209-8.00011-0

Lazzarini, R., Tianfield, H., & Charissis, V. (2023). A stacking ensemble of deep learning models for IoT intrusion detection. *Knowledge-Based Systems*, *279*, 110941. https://doi.org/10.1016/J.KNOSYS.2023.110941

Liu, G., Li, X., Guo, Y., Zhang, L., Liu, H., & Ai, H. (2024). Ensemble multiclassification model for predicting developmental toxicity in zebrafish. *Aquatic Toxicology*, *271*, 106936. https://doi.org/10.1016/J.AQUATOX.2024.106936

Macedo, F., Valadas, R., Carrasquinha, E., Oliveira, M. R., & Pacheco, A. (2022). Feature selection using Decomposed Mutual Information Maximization. *Neurocomputing*, *513*, 215–232. https://doi.org/10.1016/J.NEUCOM.2022.09.101

Mienye, I. D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access*, *10*, 99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287

Mihoub, A., Fredj, O. Ben, Cheikhrouhou, O., Derhab, A., & Krichen, M. (2022). Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Computers & Electrical Engineering*, *98*, 107716. https://doi.org/10.1016/J.COMPELECENG.2022.107716

Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, *72*, 102994. https://doi.org/10.1016/J.SCS.2021.102994

Munir, M., Ardiansyah, I., Santoso, J. D., Mustopa, A., & Mulyatun, S. (2022). DETECTION AND MITIGATION OF DISTRIBUTED DENIAL OF SERVICE ATTACKS ON NETWORK ARCHITECTURE SOFTWARE DEFINED NETWORKING USING THE NAIVE BAYES ALGORITHM. *Journal of Information System Management (JOISM)*, *3*(2), 51–55. https://doi.org/10.24076/JOISM.2022V3I2.656

NETSCOUT DDoS Threat Intelligence Report - Latest Cyber Threat Intelligence Report. Retrieved 06/16/2025 from https://www.netscout.com/threatreport

Ogunsanya, M., Isichei, J., & Desai, S. (2023). Grid search hyperparameter tuning in additive manufacturing processes. *Manufacturing Letters*, *35*, 1031–1042. https://doi.org/10.1016/J.MFGLET.2023.08.056

Paranjape, A., Katta, P., & Ohlenforst, M. (2022). Automated Data Pre-processing for Machine Learning based Analyses. In *COLLA 2022: The Twelfth International*

Conference on Advanced Collaborative Networks, Systems and Applications (pp. 1–8). International Academy, Research, and Industry Association (IARIA). Retrieved from https://personales.upv.es/thinkmind/dl/conferences/colla/colla_2022/colla_2022_1_10_50012.pdf

Rafie, A., Moradi, P., & Ghaderzadeh, A. (2023). A Multi-Objective online streaming Multi-Label feature selection using mutual information. *Expert Systems with Applications*, *216*, 119428. https://doi.org/10.1016/J.ESWA.2022.119428

Taha, A. A., & Malebary, S. J. (2022). A Hybrid Meta-Classifier of Fuzzy Clustering and Logistic Regression for Diabetes Prediction. *Computers, Materials & Continua*, *71*(3), 6089–6105. https://doi.org/10.32604/CMC.2022.023848

Targeted by 20.5 million DDoS attacks, up 358% year-over-year: Cloudflare's 2025 Q1 DDoS Threat Report. Retrieved 06/16/2025 from https://blog.cloudflare.com/ddos-threat-report-for-2025-q1/

Zhou, H., Wang, X., & Zhu, R. (2022). Feature selection based on mutual information with correlation coefficient. *Applied Intelligence*, *52*(5), 5457–5474. https://doi.org/10.1007/S10489-021-02524-X/METRICS