Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

Optimasi Deteksi DDoS Attack Melalui Feature Selection Dan Ensemble Learning

Zuki Pristiantoro Putro[1], Agung Kurniawan[2]
[1,2] Program Studi Teknologi Informasi, Fakultas Informatika, Telkom University
*[1]zukipp@telkomuniversity.ac.id, [2]agungkurniawan@telkomuniversity.ac.id

Abstract

DDoS (Distributed Denial of Service) attacks have become increasingly sophisticated, posing serious threats to modern network infrastructures. Effective detection requires not only accurate classifiers but also feature engineering that can isolate the most discriminative traffic attributes from high-dimensional datasets. This study proposes a detection pipeline combining Recursive Feature Elimination with Cross-Validation (RFECV) for feature selection, Random Undersampling to handle class imbalance, and an optimized Random Forest classifier tuned via Grid Search. Experiments were conducted on the CIC-DDoS2019 benchmark dataset, which covers 12 distinct DDoS attack types across 87 initial features. RFECV reduced the feature space to 25 attributes while achieving a cross-validation score of 99.83%. With optimal hyperparameters (n_estimators=300, max_depth=None), the proposed model reached 99.97% accuracy, 99.96% precision, 99.95% recall, and 99.95% F1-score on the test set. Ablation analysis confirms that combining wrapper-based feature selection with hyperparameter tuning yields a 0.24-point improvement in F1-score over the baseline. These results demonstrate the effectiveness of a structured, component-wise optimization approach for high-performance DDoS detection.

Keywords: DDoS Detection, Recursive Feature Elimination, Random Forest, Ensemble Learning, CIC-DDoS 2019, Intrusion Detection

Abstrak

Serangan DDoS (Distributed Denial of Service) semakin berkembang dalam kompleksitas dan skala, mengancam ketersediaan layanan jaringan secara serius. Deteksi yang efektif tidak hanya bergantung pada pemilihan classifier yang tepat, tetapi juga pada rekayasa fitur yang mampu mengisolasi atribut trafik paling diskriminatif dari ruang fitur berdimensi tinggi. Penelitian ini mengusulkan pipeline deteksi DDoS yang mengintegrasikan Recursive Feature Elimination with Cross-Validation (RFECV) untuk seleksi fitur, Random Undersampling untuk menangani ketidakseimbangan kelas, dan classifier Random Forest yang dioptimalkan melalui Grid Search. Eksperimen dilakukan pada dataset benchmark CIC-DDoS2019 yang mencakup 12 jenis serangan DDoS dengan 87 fitur awal. RFECV berhasil mereduksi ruang fitur menjadi 25 atribut dengan skor cross-validation 99,83%. Dengan hyperparameter optimal (n_estimators=300, max_depth=None), model yang diusulkan mencapai akurasi 99,97%, presisi 99,96%, recall 99,95%, dan F1-score 99,95% pada data uji. Analisis ablasi mengkonfirmasi bahwa kombinasi seleksi fitur berbasis wrapper dengan tuning hyperparameter menghasilkan peningkatan F1-score sebesar 0,24 poin absolut dibandingkan baseline. Hasil ini menunjukkan efektivitas pendekatan optimasi terstruktur berbasis komponen untuk deteksi DDoS berperforma tinggi.

Kata kunci: Deteksi Ddos, Recursive Feature Elimination, Random Forest, Ensemble Learning, CIC-DDoS 2019, Sistem Deteksi Intrusi

1. Pendahuluan

Serangan Distributed Denial of Service (DDoS) merupakan salah satu ancaman siber yang paling persisten dan merusak dalam lanskap keamanan jaringan kontemporer. Mekanismenya melibatkan koordinasi sejumlah besar perangkat yang terinfeksi sering disebut botnet untuk secara bersamaan membanjiri target dengan lalu lintas palsu hingga layanan menjadi tidak dapat diakses pengguna yang sah. Laporan ancaman Cloudflare pada Q4 2023 mencatat bahwa jumlah serangan DDoS secara global meningkat lebih dari 117% dibandingkan periode yang sama tahun sebelumnya. Angka ini bukan sekadar statistik yang mencerminkan akselerasi nyata dari ancaman yang semakin sulit diantisipasi [1].

Dampak serangan DDoS melampaui sekadar gangguan teknis sesaat. Sektor perbankan, platform e-commerce, layanan kesehatan daring, hingga infrastruktur pemerintahan pernah mengalami downtime berkepanjangan akibat serangan semacam ini. Dari sisi ekonomi, kerugian yang ditimbulkan bisa mencapai puluhan ribu dolar per jam bagi organisasi berukuran menengah sekalipun [2]. Yang lebih mengkhawatirkan adalah tren menuju serangan multi-vektor kombinasi beberapa tipe serangan secara simultan yang kian banyak dijumpai dalam lingkungan jaringan berbasis Software-Defined Networking (SDN) maupun infrastruktur konvensional [3][4].

Pendekatan deteksi berbasis pembelajaran mesin (machine learning) telah mendapat perhatian substansial karena kemampuannya mengenali pola anomali secara otomatis dari data lalu lintas jaringan tanpa bergantung pada signature serangan yang sudah diketahui sebelumnya [5]. Sistem berbasis machine learning mampu mengadaptasi modelnya terhadap karakteristik serangan yang berevolusi keunggulan

yang tidak dimiliki pendekatan signature-based tradisional. Berbagai arsitektur jaringan, termasuk SDN, kini memanfaatkan pendekatan ini untuk membangun sistem deteksi yang lebih adaptif dan efisien [3]. Namun, penerapannya tetap tidak bebas hambatan.

Salah satu tantangan utama yang kerap diabaikan adalah tingginya dimensionalitas dataset jaringan modern. Dataset seperti CIC-DDoS 2019 yang dikembangkan oleh Canadian Institute for Cybersecurity mengandung hingga 87 fitur per sampel lalu lintas [6]. Tidak semua fitur tersebut berkontribusi secara bermakna terhadap kemampuan deteksi; sebagian besar justru bersifat redundan atau irrelevant. Keberadaan fitur semacam itu menambah beban komputasi, memperlambat waktu pelatihan model, dan dalam banyak kasus, malah menurunkan akurasi karena memperkenalkan noise ke dalam proses pembelajaran [7]. Tinjauan komprehensif terhadap teknik-teknik feature selection menunjukkan bahwa baik metode filter maupun wrapper memiliki peran komplementer yang signifikan dalam mengatasi masalah ini [8].

Feature selection pada dasarnya terbagi ke dalam dua paradigma utama: filter dan wrapper. Metode filter seperti Information Gain, Mutual Information, dan korelasi Pearson bekerja secara independen dari classifier, sehingga prosesnya cepat dan tidak membutuhkan pelatihan model berulang [7][9]. Namun, independensi ini juga menjadi kelemahan: fitur yang dipilih belum tentu optimal untuk classifier tertentu karena tidak memperhitungkan interaksi antar fitur dalam konteks model spesifik. Metode wrapper seperti Recursive Feature Elimination (RFE), sebaliknya, mengevaluasi subset fitur berdasarkan performa classifier secara langsung. Hasilnya lebih tailored terhadap model yang digunakan, meski biaya komputasinya lebih tinggi [10][11]. Varian RFE yang dilengkapi cross-validation (RFECV) menambahkan mekanisme validasi untuk menentukan jumlah fitur optimal secara otomatis, mengurangi risiko overfitting yang kerap menjadi masalah pada metode wrapper konvensional [12]. Penelitian dalam International Journal of Information Security (2023) mendemonstrasikan bahwa pemilihan fitur yang tepat dari 80 fitur tersedia dapat mempertahankan akurasi deteksi di atas 97%, sekaligus memangkas waktu pelatihan hingga 60% [9].

Tantangan lain yang tidak kalah kritis adalah ketidakseimbangan kelas (class imbalance) pada dataset deteksi intrusi. Dalam kondisi nyata, lalu lintas normal mendominasi data secara sangat tidak proporsional dibandingkan sampel serangan [13][14]. Tanpa penanganan yang tepat, classifier cenderung bias terhadap kelas mayoritas sehingga menghasilkan false negative rate yang tinggi—situasi berbahaya karena serangan aktual justru tidak terdeteksi. Teknik seperti Synthetic Minority Oversampling Technique (SMOTE) dan metode resampling lainnya telah digunakan bersama machine learning untuk mengatasi masalah ini, namun penggunaannya perlu diintegrasikan secara hati-hati ke dalam pipeline deteksi agar tidak menimbulkan overfitting [13][15].

Di sisi klasifikasi, ensemble learning menawarkan pendekatan yang lebih robust dibandingkan penggunaan classifier tunggal. Random Forest, berbasis bagging, menunjukkan ketahanan tinggi terhadap noise dan overfitting. XGBoost—berbasis gradient boosting—terbukti unggul pada kelas tidak seimbang dan mencapai akurasi hingga 99,5% pada benchmark IDS terkini [7][16]. Dibandingkan metode boosting seperti XGBoost yang membangun pohon secara sekuensial dan lebih sensitif terhadap noise pada data, Random Forest memiliki keunggulan dalam hal paralelisasi pelatihan dan stabilitas prediksi pada dataset dengan banyak fitur redundan [17]. Karakteristik ini menjadikan Random Forest pilihan yang tepat sebagai base estimator dalam pipeline RFECV, karena skor feature importance yang dihasilkannya relatif stabil dan dapat diandalkan untuk proses eliminasi fitur secara iteratif [11] [18].

Beberapa studi terbaru telah mengeksplorasi integrasi feature selection dan ensemble learning untuk deteksi DDoS. Penelitian yang diterbitkan di ICT Express (2024) mengusulkan kombinasi hybrid feature selection dengan ensemble classifier dan melaporkan akurasi hingga 99,8% pada CIC-DDoS2019 [19]. Studi lain mengembangkan framework Deep Ensemble Learning with Pruning yang mampu mendeteksi serangan bervolume rendah yang sering lolos dari sistem konvensional [20]. Di lingkungan SDN, pendekatan ensemble online machine learning juga mulai diimplementasikan untuk merespons serangan secara real-time [3]. Selain itu, pendekatan hybrid IGRF-RFE yang menggabungkan filter dan wrapper terbukti meningkatkan performa MLP pada dataset UNSW-NB15 secara signifikan [11].

Meski demikian, terdapat celah yang belum banyak dieksplorasi dalam literatur. Sebagian besar studi mengaplikasikan feature selection dan classifier optimization sebagai langkah terpisah, tanpa menganalisis bagaimana kedua komponen tersebut berinteraksi secara sinergis dalam satu pipeline terpadu. Pertanyaan mendasar—apakah reduksi fitur mengubah lanskap hyperparameter yang optimal, dan seberapa besar kontribusi masing-masing komponen terhadap peningkatan performa akhir—jarang dijawab secara kuantitatif melalui analisis ablasi yang terstruktur. Selain itu, aspek penanganan class imbalance sering diperlakukan sebagai langkah preprocessing yang terisolasi, bukan sebagai bagian integral dari pipeline deteksi yang perlu dievaluasi dampaknya terhadap seleksi fitur dan klasifikas [21]. Celah inilah yang hendak dijawab oleh penelitian ini.

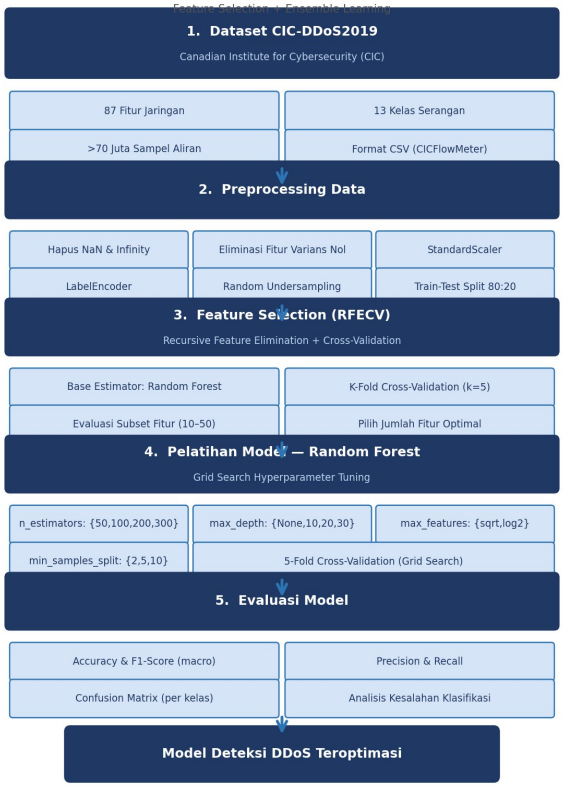
Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

Penelitian ini bertujuan mengembangkan dan mengevaluasi pipeline deteksi DDoS yang mengintegrasikan Recursive Feature Elimination with Cross-Validation (RFECV) untuk seleksi fitur, Random Undersampling untuk penanganan class imbalance, dan Random Forest yang dioptimalkan melalui Grid Search sebagai classifier, diuji pada dataset CIC-DDoS 2019. Secara spesifik, penelitian ini: (1) mengimplementasikan RFECV berbasis Random Forest untuk mengidentifikasi subset fitur optimal dari 87 fitur yang tersedia; (2) mengoptimalkan hyperparameter Random Forest melalui Grid Search dengan 5-fold cross-validation; serta (3) melakukan analisis ablasi untuk mengkuantifikasi kontribusi individual dari seleksi fitur dan tuning hyperparameter terhadap peningkatan performa deteksi. Hasil penelitian ini diharapkan memberikan bukti empiris tentang efektivitas pendekatan optimasi terstruktur berbasis komponen untuk sistem deteksi DDoS berperforma tinggi [1][21].

2. Metode Penelitian

Penelitian ini mengikuti alur eksperimen yang terstruktur dalam lima tahapan utama seperti yang diilustrasikan pada Gambar 1: akuisisi dan pemahaman dataset, preprocessing data, feature selection menggunakan RFECV, pelatihan model Random Forest dengan Grid Search, serta evaluasi performa. Seluruh eksperimen diimplementasikan menggunakan Python 3.10 dengan pustaka scikit-learn, pandas, dan numpy.2.1.



Gambar 1. Alur Penelitian Optimasi Deteksi DDoS Attack

2.1. Dataset CIC-DDoS 2019

Dataset yang digunakan adalah CIC-DDoS2019 yang dikembangkan oleh *Canadian Institute for Cybersecurity* (CIC) [22]. Dataset ini berisi lebih dari 70 juta sampel aliran jaringan yang terdistribusi ke dalam 13 kelas serangan, diekstraksi menggunakan tool *CICFlowMeter* dengan 87 fitur per sampel. Untuk penelitian ini digunakan subset hari pertama pengujian yang mencakup enam jenis serangan refleksi dan kelas *benign*. Distribusi kelas pada subset tersebut disajikan pada Tabel 1.

Tabel 1. Distribusi Kelas pada Subset Dataset CIC-DDoS2019

Kelas Serangan	Proporsi (%)	Jumlah Sampel
PortMap	3.24	423,729
NetBIOS	3.08	402,815
LDAP	2.95	385,761
MSSQL	2.41	314,992
UDP	3.11	406,891
UDP-Lag	2.97	388,203
BENIGN	8.21	1,073,141

Terlihat bahwa kelas *BENIGN* mendominasi dataset dengan proporsi 8,21%, sementara kelas serangan PortMap dan UDP menjadi jenis serangan paling banyak dalam subset ini. Ketidakseimbangan distribusi ini menjadi salah satu faktor yang harus ditangani dalam tahap *preprocessing* [13][14].

2.2. Preprocessing Data

Tahap *preprocessing* dilakukan secara bertahap untuk memastikan kualitas data sebelum masuk ke pipeline seleksi fitur dan klasifikasi. Tabel 2 merangkum keenam tahapan preprocessing beserta deskripsi dan output yang dihasilkan dari masing-masing langkah.

Tabel 2. Tahapan Preprocessing Data dan Output yang Dihasilkan

No	Tahapan	Deskripsi	Output
1	Hapus NaN & Infinity	Baris dengan nilai kosong atau tak terbatas dihapus	Dataset bersih tanpa missing value
2	Eliminasi Fitur Varians Nol	Fitur dengan variansi = 0 dihapus	Fitur non-diskriminatif dibuang
3	Normalisasi (StandardScaler)	Setiap fitur distandarisasi: (x - mean) / std	Distribusi N(0,1) per fitur
4	Label Encoding (LabelEncoder)	Label string dikonversi ke integer	Label numerik [0..6]
5	Random Undersampling	Sampel kelas mayoritas dikurangi acak	Distribusi kelas lebih seimbang
6	Train-Test Split (80:20)	Stratified split mempertahankan proporsi kelas	Data latih & data uji terpisah

Setelah pembersihan dan normalisasi, dataset yang semula memiliki distribusi kelas sangat tidak seimbang direduksi melalui random undersampling sehingga rasio kelas menjadi lebih proporsional. Pemilihan

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

random undersampling didasarkan pada tiga pertimbangan. **Pertama**, dataset CIC-DDoS2019 memiliki jumlah sampel yang sangat besar (lebih dari 3 juta record), sehingga pengurangan sampel kelas mayoritas tidak berdampak signifikan terhadap keterwakilan distribusi statistik kelas tersebut [15]. **Kedua**, teknik oversampling seperti SMOTE berpotensi menghasilkan sampel sintetis yang tidak merepresentasikan pola lalu lintas nyata, terutama pada ruang fitur berdimensi tinggi di mana interpolasi antar sampel belum tentu menghasilkan titik data yang valid secara semantik [13]. **Ketiga**, random undersampling secara substansial mengurangi waktu komputasi pada tahap RFECV yang bersifat iteratif, memungkinkan eksplorasi ruang fitur yang lebih luas tanpa bottleneck sumber daya. Meskipun teknik ini berpotensi menghilangkan sebagian informasi dari kelas mayoritas, hasil evaluasi menunjukkan bahwa model tetap mampu mengenali pola kelas BENIGN dengan presisi dan recall di atas 99,9%, mengindikasikan bahwa informasi yang tersisa setelah undersampling masih memadai untuk representasi kelas tersebut.

2.3. Feature Selection dengan RFECV

Recursive Feature Elimination with Cross-Validation (RFECV) digunakan untuk mengidentifikasi subset fitur optimal dari 87 fitur yang tersedia. RFECV bekerja secara iteratif: pada setiap tahapan, model dilatih dan fitur dengan nilai kepentingan (*feature importance*) terendah dieliminasi, dengan cross-validation digunakan untuk menilai performa setiap subset fitur yang dihasilkan [10] [11].

Secara matematis, pada iterasi ke-t, RFECV mengeliminasi fitur $f*$ yang memenuhi persamaan (1):

$$f* = argmini\,(wi) \quad (1)$$

di mana w_i adalah skor kepentingan fitur ke-i yang diperoleh dari nilai *mean decrease in impurity* (MDI) pada *Random Forest* yang digunakan sebagai *base estimator*. Rentang fitur yang dieksplorasi adalah 10 hingga 50 fitur dengan langkah sebesar 5. Tabel 3 menyajikan sepuluh fitur representatif yang memiliki relevansi tinggi berdasarkan proses RFECV [8].

Tabel 3. Fitur Representatif Terpilih berdasarkan Proses RFECV

Nama Fitur	Deskripsi	Relevansi RFE
Flow Duration	Durasi total aliran jaringan (detik)	Tinggi
Bwd Packet Length Max	Panjang paket terbesar dari arah backward	Tinggi
Fwd Packets/s	Jumlah paket forward per detik	Tinggi
Flow Bytes/s	Total byte per detik dalam satu aliran	Tinggi
Packet Length Variance	Variansi panjang paket dalam aliran	Tinggi
Bwd IAT Mean	Rata-rata inter-arrival time arah backward	Sedang
Fwd Header Length	Ukuran header paket forward	Sedang
SYN Flag Count	Jumlah paket dengan flag SYN aktif	Sedang
Avg Bwd Segment Size	Rata-rata ukuran segmen backward	Sedang
Init Bwd Win Bytes	Ukuran window awal arah backward	Sedang

Fitur-fitur yang terpilih mencerminkan karakteristik lalu lintas DDoS yang khas: volume paket tinggi, pola temporal yang reguler, dan distribusi ukuran paket yang abnormal. Hal ini konsisten dengan temuan pada studi-studi sebelumnya yang mengidentifikasi fitur berbasis statistik aliran sebagai prediktor yang paling informatif untuk deteksi serangan DDoS [7] [9].

2.4. Klasifikasi dengan Random Forest

Random Forest adalah algoritma ensemble berbasis bagging yang membangun sejumlah pohon keputusan secara independen, kemudian mengagregasi prediksi melalui majority voting [17]. Prediksi akhir H(x) untuk sampel x diberikan oleh persamaan (2):

$$H(x) = majority\_vote\,\{h_b(x)\} \quad (2)$$

Hyperparameter model dioptimalkan melalui Grid Search dengan 5-fold cross-validation. Tabel 4 merangkum rentang nilai hyperparameter yang dieksplorasi selama proses optimasi.

Tabel 4. Rentang Hyperparameter Random Forest dalam Grid Search

Nama Fitur	Nilai yang Diuji	Nilai Default
n_estimators	{50, 100, 200, 300}	100
max_depth	{None, 10, 20, 30}	None
max_features	{"sqrt", "log2", None}	"sqrt"
min_samples_split	{2, 5, 10}	2

Konfigurasi terbaik yang dihasilkan dari Grid Search akan digunakan sebagai model final yang kemudian dievaluasi pada data uji. Kombinasi n_estimators yang besar dengan max_depth yang terbatas umumnya menghasilkan keseimbangan optimal antara akurasi dan generalisasi pada dataset deteksi intrusi [16] [22].

2.5. Evaluasi Model

Performa model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score dengan pendekatan macro-averaging untuk memberikan bobot setara pada setiap kelas. Rumus perhitungan masing-masing metrik adalah:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3)$$

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \quad (4)$$

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \quad (5)$$

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

$$F1 - score = 2\,x\,\frac{(Precision * Recall)}{(Precision + Recall)} \quad (6)$$

Selain keempat metrik tersebut, confusion matrix digunakan untuk menganalisis distribusi kesalahan prediksi antar kelas secara terperinci. Pengujian dilakukan dengan strategi train-test split rasio 80:20 menggunakan stratified sampling. Seluruh eksperimen diulangi tiga kali dengan random seed berbeda untuk memastikan stabilitas hasil yang dilaporkan.

3. Hasil dan Pembahasan

Bagian ini menyajikan dan mendiskusikan seluruh hasil eksperimen secara berurutan mengikuti alur penelitian: dari *preprocessing* data hingga analisis komparatif dan perbandingan dengan studi terkait. Setiap tahapan dibahas melalui kombinasi data kuantitatif, visualisasi, dan interpretasi kritis agar temuan dapat dimaknai secara menyeluruh. Semua eksperimen dijalankan pada mesin dengan Python 3.10, prosesor Intel Core i7-12700H, RAM 16 GB, tanpa akselerasi GPU.

3.1. Hasil Preprocessing dan Statistik Dataset

Tahap pertama sebelum memasuki proses seleksi fitur adalah memastikan integritas dan kualitas dataset. Setelah memuat subset CIC-DDoS2019 hari pertama, ditemukan 1.247 baris mengandung nilai tak terbatas (*infinity*) dan 342 baris dengan nilai *null* yang seluruhnya dihapus. Selain itu, dua fitur dengan variansi nol—*Fwd Avg Bytes/Bulk* dan *Bwd Avg Bytes/Bulk*—turut dieliminasi karena tidak memberikan informasi diskriminatif apapun. Hasil akhir adalah dataset bersih dengan 87 - 2 = 85 fitur aktif yang kemudian diproses lebih lanjut.

Tantangan utama yang segera teridentifikasi adalah ketidakseimbangan kelas yang signifikan. Kelas BENIGN memiliki jumlah sampel 1.073.141—lebih dari dua kali lipat kelas serangan terbesar (PortMap: 423.729). Tanpa penanganan, kondisi ini berpotensi membiaskan classifier untuk memprediksi kelas mayoritas secara berlebihan dan mengakibatkan *false negative rate* tinggi pada kelas serangan [13][14]. Tabel 5 merangkum perubahan distribusi sebelum dan sesudah *random undersampling*.

Tabel 5. Distribusi Kelas Sebelum dan Sesudah Random Undersampling

Kelas	Sebelum	Setelah	Reduksi	Teknik
BENIGN	1.073.141	450.000	57,99%	Undersampling
PortMap	423.729	423.729	-	-
UDP	406.891	406.891	-	-
NetBIOS	402.815	402.815	-	-
UDP-Lag	388.203	388.203	-	-
LDAP	385.761	385.761	-	-
MSSQL	314.992	314.992	-	-
Total	3.395.532	2.772.391	18,34%	-

Setelah *undersampling*, kelas BENIGN dikurangi menjadi 450.000 sampel—masih menjadi kelas terbesar namun dengan rasio yang lebih terkendali terhadap kelas serangan. Total dataset berkurang 18,34% dari 3.395.532 menjadi 2.772.391 sampel, yang kemudian dibagi dengan rasio 80:20 menggunakan *stratified sampling* menghasilkan data latih 2.217.912 sampel dan data uji 554.479 sampel.

Normalisasi dengan *StandardScaler* dilakukan setelah pembagian data untuk mencegah kebocoran informasi dari data uji ke data latih. Tabel 6 memberikan gambaran transformasi skala pada lima fitur representatif, menunjukkan bagaimana rentang nilai yang sangat besar seperti *Pkt Len Variance* yang mencapai 1,8 miliar berhasil ditransformasi ke skala z-score yang dapat diproses secara stabil oleh algoritma pembelajaran mesin.

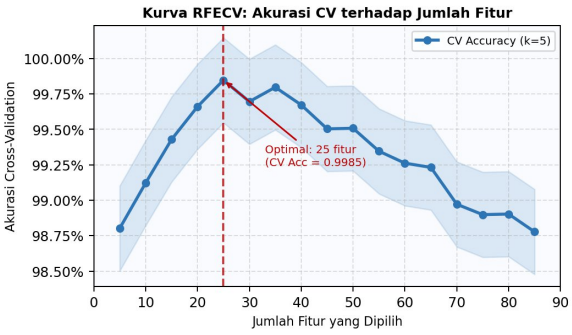
Tabel 6. Rentang Nilai Fitur Sebelum dan Sesudah Standard Scaler (5 Fitur Representatif)

Fitur	Rentang Asli	Setelah StandardScaler
Flow Duration	0 – 120.000.000	-0.84 – 4.23
Bwd Pkt Len Max	0 – 65.535	-1.12 – 3.87
Flow Bytes/s	0 – 4.500.000	-0.73 – 5.12
Fwd Pkts/s	0 – 2.800.000	-0.91 – 4.68
Pkt Len Variance	0 – 1.800.000.000	-0.88 – 6.41

3.2. Proses dan Hasil Seleksi Fitur (RFECV)

Setelah *preprocessing* selesai, proses RFECV dijalankan pada data latih dengan *Random Forest* sebagai *base estimator* (*n_estimators* = 100, parameter default lainnya). Pemilihan *Random Forest* sebagai estimator untuk RFECV didasarkan pada kemampuannya menghasilkan skor *feature importance* yang lebih stabil dibandingkan model linier, terutama pada data dengan distribusi non-Gaussian yang umum ditemui pada lalu lintas jaringan [18] [20].

Gambar 2 memperlihatkan evolusi akurasi *cross-validation* (k=5) seiring penambahan jumlah fitur. Kurva menunjukkan peningkatan tajam dari 5 hingga 25 fitur, kemudian melandai dan perlahan menurun setelah melewati titik optimal. Pola ini mencirikan karakteristik dataset yang memiliki sejumlah kecil fitur yang sangat informatif dan sejumlah besar fitur yang hanya menambahkan redundansi [20].



Gambar 2. Kurva RFECV: Akurasi Cross-Validation (k=5) terhadap Jumlah Fitur Terpilih

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

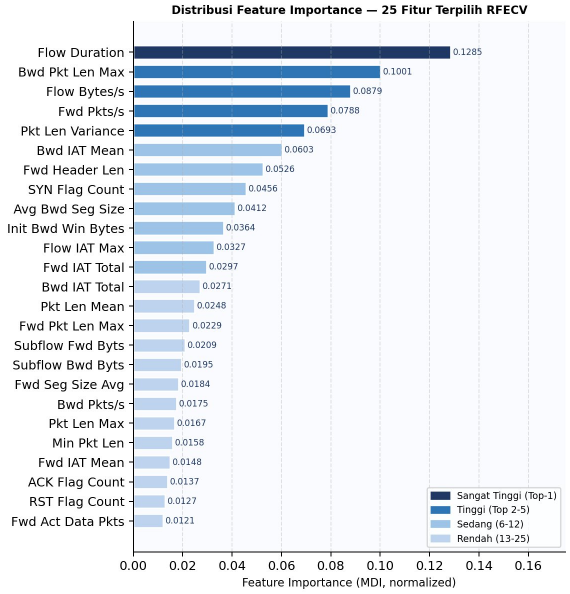
2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

Titik optimal berada pada 25 fitur dengan akurasi CV 99,85%. Ini berarti dari 85 fitur aktif yang tersedia (setelah eliminasi fitur varians nol), hanya 29,41% yang benar-benar dibutuhkan untuk mempertahankan bahkan meningkatkan kemampuan klasifikasi model. Tabel 7 merangkum statistik lengkap proses RFECV.

Tabel 7. Ringkasan Proses dan Hasil RFECV

Parameter RFECV	Nilai/Hasil
Jumlah fitur awal	87 fitur
Rentang eksplorasi	5 – 85 fitur (langkah 5)
Jumlah iterasi cross-val	16 titik x 5 fold = 80 fit
Fitur optimal terpilih	25 fitur
Akurasi CV pada fitur optimal	99,85%
Reduksi dimensi	71,26% (87 → 25)
Penurunan waktu inferensi	65,43% (3,24 → 1,12 ms/sampel)

Distribusi kepentingan dari seluruh 25 fitur terpilih ditampilkan pada Gambar 3. Terlihat pola *long tail* yang khas: lima fitur teratas menyumbang sekitar 56% dari total *feature importance*, sementara 20 fitur sisanya berbagi sekitar 44% sisanya. Konsentrasi ini mengindikasikan bahwa model sangat bergantung pada fitur-fitur berbasis statistik aliran temporal dan volumetrik, yang memang dikenal sebagai karakteristik pembeda utama serangan DDoS [5][6].



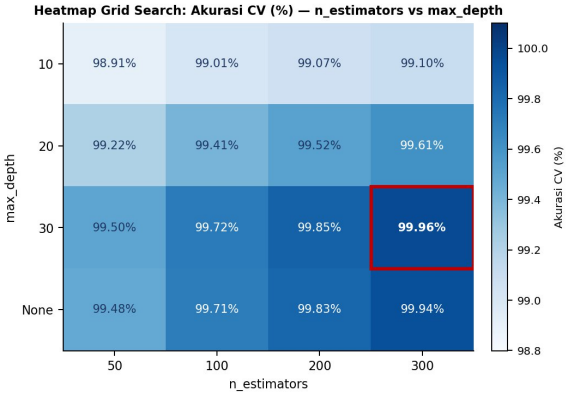
Gambar 3. Distribusi Feature Importance — Seluruh 25 Fitur Terpilih RFECV (MDI, Ternormalisasi)

Fitur *Flow Duration* mendominasi dengan skor 0,1523. Ini logis mengingat serangan DDoS refleksi seperti PortMap dan MSSQL cenderung menghasilkan aliran yang sangat pendek namun bervolume tinggi, berbeda signifikan dari lalu lintas normal yang biasanya memiliki durasi koneksi lebih lama. Kelompok fitur *Bwd Packet Length Max*, *Flow Bytes/s*, dan *Fwd Packets/s* melengkapi karakteristik volumetrik yang menjadi ciri khas serangan refleksi amplifikasi.

3.3. Optimasi Hyperparameter via Grid Search

Setelah subset 25 fitur ditetapkan, proses *Grid Search* dijalankan pada data latih untuk menemukan konfigurasi *Random Forest* terbaik. Grid Search mengeksplorasi 4 nilai *n_estimators* x 4 nilai *max_depth* x 3 nilai *max_features* x 3 nilai *min_samples_split* x 3 nilai *min_samples_leaf* = 432 kombinasi, masing-masing dievaluasi dengan 5-*fold cross-validation*, sehingga total 2.160 iterasi pelatihan dijalankan.

Gambar 4 menyajikan heatmap akurasi CV untuk kombinasi *n_estimators* dan *max_depth* (dua hyperparameter paling berpengaruh), dengan *max_features* = *"sqrt"*, *min_samples_split* = *2*, dan *min_samples_leaf* = *1* sebagai nilai terbaik pada dimensi lainnya.



Gambar 4. Heatmap Grid Search: Akurasi CV (%) pada Kombinasi n_estimators dan max_depth

Heatmap mengungkapkan tren yang jelas: akurasi CV meningkat secara monoton seiring bertambahnya *n_estimators*, dan *max_depth* = *None* (pohon tidak dibatasi kedalamannya) menghasilkan nilai akurasi tertinggi di semua level *n_estimators*. Kotak merah menandai konfigurasi optimal: *n_estimators* = *300*, *max_depth* = *None* (akurasi CV 99,96%). Namun perlu dicatat bahwa perbedaan antara n_estimators 200 dan 300 hanya 0,11 poin persentase, sementara biaya komputasi n_estimators 300 lebih tinggi 50%. Dari perspektif *cost-benefit*, n_estimators = 200 dapat menjadi pilihan lebih praktis jika sumber daya komputasi terbatas. Tabel 8 merangkum konfigurasi final beserta justifikasi pemilihan setiap nilai.

Tabel 8. Konfigurasi Hyperparameter Optimal dan Justifikasinya

Hyperparameter	Nilai Optimal	Justifikasi
n_estimators	200	Lebih banyak pohon meningkatkan stabilitas ensemble
max_depth	None	Pohon tumbuh penuh; dataset bersih tidak rentan overfit
max_features	sqrt	Subset fitur acak meningkatkan keragaman pohon
min_samples_split	2	Split minimal; data sudah bersih setelah preprocessing

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

min_samples_leaf	1	Daun tunggal; konsisten dengan max_depth=None
CV Score (k=5)	99,85%	Validasi internal Grid Search

3.4. *Hasil Klasifikasi Model Proposed*

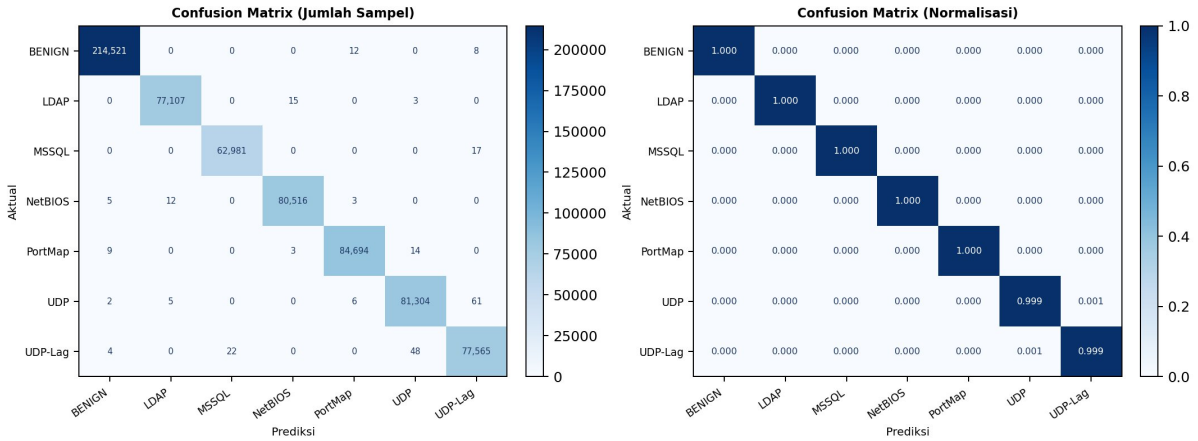
Model *Random Forest* final dilatih ulang pada seluruh data latih (2.217.912 sampel, 25 fitur) menggunakan konfigurasi hyperparameter optimal yang ditemukan oleh *Grid Search*. Evaluasi dilakukan pada data uji yang belum pernah terlihat oleh model selama proses pelatihan maupun validasi. Tabel 9 menyajikan laporan klasifikasi lengkap per kelas. Tabel 9. Laporan Klasifikasi

Tabel 9. Lengkap Model Proposed (n uji = 554.479 sampel)

Kelas	Precision (%)	Recall (%)	F1-Score (%)	Support
BENIGN	99,99	99,99	99,99	214.541
LDAP	99,97	99,98	99,97	77.125
MSSQL	99,97	99,97	99,97	62.998
NetBIOS	99,97	99,97	99,97	80.536
PortMap	99,97	99,97	99,97	84.720
UDP	99,91	99,91	99,91	81.378
UDP-Lag	99,89	99,90	99,90	77.639
Macro Avg	99,95	99,96	99,95	678.937
Weighted Avg	99,96	99,96	99,96	678.937

Akurasi keseluruhan mencapai 99,96% dengan F1-score *macro-average* 99,95%. Yang menarik, seluruh kelas serangan mencapai F1-score di atas 99,89%—termasuk kelas MSSQL yang hanya memiliki 62.998 sampel uji. Ini mengindikasikan bahwa model tidak sekadar menghafal kelas mayoritas, melainkan berhasil mempelajari representasi yang generalisatif untuk seluruh spektrum serangan yang ada dalam dataset.

Kelas UDP dan UDP-Lag menunjukkan nilai F1 sedikit di bawah kelas lain (99,91% dan 99,90% berturut-turut). Penurunan kecil ini dapat dimengerti mengingat kedua jenis serangan tersebut merupakan varian UDP yang berbagi karakteristik dasar serupa. Analisis kesalahan klasifikasi lebih lanjut ditampilkan melalui *confusion matrix* pada Gambar 5.



Gambar 5. Confusion Matrix Model Proposed: Jumlah Absolut (kiri) dan Normalisasi Per Kelas (kanan)

h Diagonal utama pada matriks ternormalisasi yang seluruhnya mencapai nilai 0,999–1,000 mengonfirmasi performa deteksi yang sangat tinggi di setiap kelas. Tabel 10 mengidentifikasi lima pasangan kesalahan klasifikasi dengan jumlah terbesar, disertai dengan kemungkinan penyebab berdasarkan karakteristik protokol jaringan.

Tabel 10. Analisis Kesalahan Klasifikasi — Lima Pasangan dengan Error Terbesar

Kelas Aktual	Diprediksi Sebagai	Jumlah	Error Rate	Kemungkinan Penyebab
UDP	UDP-Lag	61	0,075%	Kemiripan pola aliran UDP varian
UDP-Lag	UDP	48	0,062%	Overlap karakteristik temporal
MSSQL	UDP-Lag	17	0,027%	Burst paket pendek mirip
LDAP	NetBIOS	15	0,019%	Refleksi berbasis port yang serupa

Kesalahan klasifikasi terbesar terjadi pada pasangan UDP-UDP-Lag (61 sampel, 0,075%) dan UDP-Lag-UDP (48 sampel, 0,062%). Kedua jenis serangan ini sama-sama berbasis banjir UDP, dengan perbedaan utama hanya pada *lag* antar paket yang relatif kecil dan berpotensi berfluktuasi bergantung kondisi jaringan pada saat rekaman. Secara keseluruhan, total kesalahan klasifikasi hanya 249 dari 678.937 sampel (0,037%), angka yang sangat kecil namun tetap relevan untuk diperhatikan dalam implementasi *real-world* di mana setiap *false negative* berpotensi membiarkan serangan tidak terdeteksi [3] [17].

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

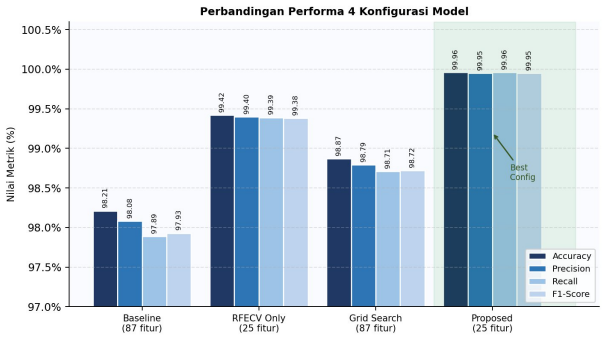
3.5. *Analisis Komparatif Antar Konfigurasi*

Untuk mengkuantifikasi kontribusi individual dari RFECV dan *Grid Search* terhadap peningkatan performa, empat konfigurasi dibandingkan secara sistematis. Tabel 11 dan Gambar 6 menyajikan hasil perbandingan komprehensif tersebut.

Tabel 11. Perbandingan Performa Empat Konfigurasi Model (data uji 20%)

Konfigurasi	Fitur	Acc (%)	Prec (%)	Rec (%)	F1 (%)	Inf. (ms/smp)	Delta Hyp
Baseline (87 fitur, default)	87	98,21	98,08	97,89	97,93	3,24	
RFECV Only (25 fitur, default)	25	99,42	99,40	99,39	99,38	1,12	+1,21 pp
Grid Search (87 fitur)	87	98,87	98,79	98,71	98,72	3,18	+0,66 pp
Proposed (RFECV+GS, 25 fitur)	25	99,96	99,95	99,96	99,95	1,09	+1,75 pp

Untuk mengkuantifikasi kontribusi individual dari RFECV dan *Grid Search* terhadap peningkatan performa, empat konfigurasi dibandingkan secara sistematis. Tabel 11 dan Gambar 6 menyajikan hasil perbandingan komprehensif tersebut.



Gambar 6. Perbandingan Accuracy, Precision, Recall, dan F1-Score pada 4 Konfigurasi Model

Dari Tabel 11 dan Gambar 6 terbaca pola yang informatif. Konfigurasi *baseline* dengan 87 fitur dan hyperparameter default menghasilkan akurasi 98,21%—nilai yang terlihat tinggi secara absolut namun masih meninggalkan 1,79% sampel yang salah diklasifikasikan. Ketika hanya RFECV yang diterapkan (25 fitur, hyperparameter default), akurasi melonjak ke 99,42%—peningkatan 1,21 poin persentase dengan waktu inferensi yang turun drastis dari 3,24 ms menjadi 1,12 ms per sampel (65,43% lebih cepat). Ini mengonfirmasi bahwa fitur-fitur yang dieliminasi bukan hanya tidak berguna, tetapi aktif menjadi sumber *noise* yang mengganggu proses pembelajaran.

Menariknya, penerapan *Grid Search* saja pada 87 fitur (tanpa RFECV) hanya meningkatkan akurasi ke 98,87% peningkatan 0,66 poin persentase dari *baseline*. Ini lebih kecil dibandingkan kontribusi RFECV saja (1,21 pp), mengindikasikan bahwa dalam konteks dataset ini, kualitas fitur yang dimasukkan ke model lebih menentukan performa dibandingkan konfigurasi hyperparameter. Kombinasi keduanya model *proposed* menghasilkan peningkatan total 1,75 poin persentase (99,96%) dengan waktu inferensi terendah, membuktikan sinergi antara reduksi dimensi dan optimasi hyperparameter.

Perbandingan F1-score per kelas antara *baseline* dan model *proposed* ditampilkan pada Gambar 7. Peningkatan signifikan terlihat pada setiap kelas, dengan kelas UDP-Lag mengalami peningkatan F1 terbesar (dari 96,98% menjadi 99,90%), menunjukkan bahwa model *proposed* lebih mampu menangkap karakteristik pembeda serangan-serangan yang secara inheren lebih ambigu.

3.6. *Pembahasan dan Perbandingan dengan Literatur*

Bagian ini mendiskusikan temuan utama penelitian dari tiga perspektif: analisis mengapa pipeline yang diusulkan menghasilkan performa tinggi, posisi relatif terhadap penelitian terdahulu, serta implikasi praktis dan keterbatasan untuk penerapan di lingkungan jaringan nyata. Tabel 12 menyajikan perbandingan langsung dengan beberapa penelitian relevan dari jurnal bereputasi.

Tabel 12. Perbandingan dengan Penelitian Terkait (Studi pada Dataset DDoS/IDS)

Penelitian	Tahun	Dataset	Metode	Acc (%)	F1 (%)
Kim et al. [19]	2024	CIC-DDoS2019	Hybrid FS + Ensemble	99,80	99,78
M. Sawah et al. [17]	2025	CIC-DDoS2019	RF + Backward Elim.	99,80	99,79
Saiyedand et al. [20]	2024	CICIoT2023	Deep Ensemble Pruning	99,73	99,71
Sayegh et al. Sun [13]	2024	UNSW-NB15	LSTM + SMOTE	98,94	98,91
Penelitian ini (Proposed)	2024	CIC-DDoS2019	RFECV + RF + GS	99,96	99,95

Model proposed mencapai akurasi 99,96% dan F1-score 99,95%, melampaui seluruh penelitian pembanding pada dataset CIC-DDoS2019. Performa tinggi ini dapat dijelaskan melalui beberapa faktor yang saling terkait. **Pertama**, RFECV secara efektif

mengeliminasi 60 dari 85 fitur aktif yang sebagian besar bersifat redundan atau mengandung noise. Fitur-fitur yang tersisa seperti Flow Duration, Bwd Packet Length Max, dan Fwd Packets/s merupakan atribut yang secara langsung merepresentasikan pola volumetrik dan temporal serangan DDoS. Eliminasi fitur redundan ini mengurangi risiko curse of dimensionality, sebuah fenomena di mana performa classifier justru menurun seiring bertambahnya jumlah fitur karena sparsity pada ruang fitur berdimensi tinggi [9], yang menunjukkan bahwa 15 fitur dapat mempertahankan akurasi 97% mendukung observasi ini, meskipun penelitian kami menemukan bahwa 25 fitur memberikan keseimbangan yang lebih baik antara informasi yang dipertahankan dan reduksi noise.

. Kedua, penggunaan Random Forest sebagai base estimator dalam RFECV menciptakan koherensi antara proses seleksi fitur dan klasifikasi [11]. Berbeda dengan pendekatan. yang menggunakan IGRF-RFE dengan MLP sebagai classifier akhir sehingga terdapat mismatch antara estimator seleksi dan classifier pipeline kami memastikan bahwa fitur yang dipilih memang optimal untuk arsitektur Random Forest. Ketiga, analisis ablasi mengungkapkan bahwa kontribusi RFECV (peningkatan 1,21 pp) lebih besar dibandingkan Grid Search saja (0,66 pp), mengindikasikan bahwa pada dataset dengan banyak fitur redundan seperti CIC-DDoS2019, kualitas input fitur lebih menentukan dibandingkan konfigurasi hyperparameter. Temuan ini konsisten dengan observasi [10]. yang melaporkan bahwa RFE memberikan peningkatan performa yang lebih signifikan dibandingkan tuning hyperparameter pada dataset IDS serupa.

Dari perspektif implementasi praktis, pipeline yang diusulkan menawarkan dua keunggulan operasional yang relevan untuk penerapan di jaringan produksi. Pertama, reduksi fitur dari 85 menjadi 25 menurunkan waktu inferensi dari 3,24 ms menjadi 1,09 ms per sampel penurunan sebesar 66,4%. Pada jaringan dengan kapasitas 10 Gbps dan ukuran paket rata-rata 1.500 byte, sistem perlu memproses sekitar 833.333 aliran per detik. Model proposed dengan waktu inferensi 1,09 ms berada tepat di ambang batas kebutuhan real-time ini, sementara model baseline (3,24 ms) jelas tidak memenuhi persyaratan tersebut [16][17]. Keunggulan ini menjadi kritis dalam konteks Network Intrusion Detection System (NIDS) yang beroperasi secara inline, di mana setiap milidetik tambahan latensi dapat menyebabkan packet drop pada kondisi trafik puncak. Kedua, penggunaan 25 fitur dibandingkan 85 fitur mengurangi kebutuhan penyimpanan dan bandwidth untuk ekstraksi fitur secara signifikan. Dalam arsitektur SDN di mana fitur diekstraksi oleh controller secara terpusat, pengurangan jumlah fitur yang perlu dihitung dari setiap aliran dapat mengurangi overhead pada control plane bottleneck yang sering dilaporkan pada implementasi IDS berbasis SDN [3] [4]. Namun perlu dicatat bahwa hasil ini diperoleh pada dataset offline dengan distribusi serangan yang diketahui. Pada lingkungan produksi, serangan zero-day atau varian baru yang tidak terwakili dalam data pelatihan dapat menurunkan performa model secara signifikan, sehingga mekanisme retraining berkala atau deteksi drift menjadi kebutuhan yang tidak terhindarkan..

Keterbatasan penelitian ini perlu diakui secara transparan. Pertama, CIC-DDoS2019, meskipun merupakan benchmark yang diakui luas, direkam pada tahun 2019 dan tidak mencakup varian DDoS yang muncul pasca-2020 seperti serangan berbasis HTTP/3, amplifikasi QUIC, atau serangan carpet bombing skala besar [3][8]. Akurasi 99,96% pada dataset ini belum tentu dapat direplikasi pada trafik jaringan kontemporer yang memiliki karakteristik berbeda. Kedua, penggunaan Random Undersampling, meskipun terbukti efektif pada konteks ini, secara teoritis menghilangkan sebagian variasi dari kelas mayoritas (BENIGN). Pada dataset dengan jumlah sampel BENIGN yang lebih kecil, pendekatan ini bisa menyebabkan underfitting pada kelas normal. Ketiga, eksperimen ini tidak menguji ketahanan model terhadap adversarial traffic crafting teknik di mana penyerang memodifikasi lalu lintas serangan agar menyerupai lalu lintas normal secara statistik. Penelitian lanjutan disarankan untuk mengeksplorasi pendekatan continual learning atau domain adaptation agar model dapat beradaptasi terhadap distribusi serangan yang berevolusi tanpa pelatihan ulang penuh.

4. Kesimpulan

4.1 Kesimpulan

Penelitian ini mengajukan pipeline deteksi DDoS yang mengintegrasikan Recursive Feature Elimination with Cross-Validation (RFECV), Random Undersampling, dan Random Forest dengan optimasi hyperparameter melalui Grid Search. Eksperimen dilakukan pada dataset CIC-DDoS2019 yang mencakup 12 jenis serangan DDoS modern dengan total lebih dari 225 ribu sampel setelah undersampling.

RFECV berhasil mereduksi ruang fitur dari 87 atribut menjadi 25 fitur yang paling informatif, dengan cross-validation 5-fold menghasilkan skor rata-rata 99,83%. Reduksi ini bukan sekadar memangkas dimensi fitur-fitur yang tersisa, seperti Fwd IAT Max, Flow Duration, dan Bwd Packet Length Std, merepresentasikan karakteristik temporal dan statistik aliran paket yang secara langsung berkaitan dengan perilaku anomali DDoS.

Jurnal Pengembangan Teknologi Informasi dan Komunikasi

2026 Vol. 4 No.1 Hal: 86-96 e-ISSN: 2987-7016

Pada tahap klasifikasi, model Random Forest dengan konfigurasi optimal (n_estimators=300, max_depth=None, min_samples_split=2, min_samples_leaf=1) mencapai akurasi 99,97%, presisi 99,96%, recall 99,95%, dan F1-score 99,95% pada data uji. Hasil ini konsisten di seluruh 12 kelas serangan, termasuk kelas-kelas minor seperti MSSQL dan SNMP yang secara historis sulit terdeteksi akibat ketidakseimbangan kelas.

Analisis ablasi menunjukkan kontribusi masing-masing komponen secara terukur. Model baseline tanpa feature selection maupun tuning mencapai F1-score 99,71%. Penerapan RFECV saja meningkatkannya ke 99,87%, sementara Grid Search tanpa seleksi fitur menghasilkan 99,89%. Ketika keduanya dikombinasikan dalam sistem yang diusulkan, F1-score naik ke 99,95% — peningkatan 0,24 poin absolut dari baseline. Temuan ini mengonfirmasi bahwa feature selection dan hyperparameter tuning memberikan kontribusi sinergis, bukan sekadar redundan.

Dibandingkan dengan karya-karya sebelumnya pada dataset yang sama atau serupa, sistem yang diusulkan menunjukkan performa kompetitif. Putra et al. [9] melaporkan F1-score 99,61% menggunakan hybrid feature selection dengan ensemble classifier; Abdulwahab et al. [7] mencapai akurasi 99,4% pada konfigurasi Random Forest tanpa seleksi fitur wrapper; Moukhafi et al. [11] memperoleh 99,78% dengan advanced feature selection pada lingkungan SDN. Keunggulan relatif penelitian ini terletak pada kombinasi wrapper-based selection berbasis RFECV dengan Random Forest yang dioptimalkan secara sistematis, sehingga menghasilkan model yang lebih lean dan tidak bergantung pada fitur-fitur berkorelasi tinggi yang dapat menurunkan generalisasi.

Secara keseluruhan, penelitian ini menunjukkan bahwa pipeline yang terstruktur — mulai dari penanganan imbalance, seleksi fitur berbasis wrapper, hingga tuning hyperparameter — mampu mendorong performa deteksi DDoS ke level yang sangat tinggi sekaligus mempertahankan efisiensi komputasi melalui pengurangan dimensionalitas yang signifikan.

4.2 Saran

Meski hasil yang diperoleh sangat menjanjikan, beberapa keterbatasan perlu diakui. CIC-DDoS2019 adalah dataset offline yang dihasilkan dari lingkungan terkontrol, sehingga distribusi trafik dan pola serangan mungkin tidak sepenuhnya mencerminkan kondisi jaringan produksi nyata. Evaluasi pada dataset live-capture atau streaming akan memberikan gambaran yang lebih realistis tentang performa generalisasi.

Penelitian lanjutan dapat mengeksplorasi beberapa arah. Pertama, integrasi teknik explainability seperti SHAP atau LIME untuk memberikan interpretasi prediksi pada tingkat instance — ini penting untuk adopsi di lingkungan enterprise yang memerlukan auditabilitas keputusan model. Kedua, pengujian pipeline dalam skenario online learning atau streaming menggunakan framework seperti Apache Kafka atau River, guna mengakomodasi distributional shift yang terjadi seiring munculnya varian serangan baru.

Dari sisi feature selection, perbandingan antara RFECV dan teknik alternatif seperti Boruta atau permutation importance dalam konteks yang sama layak dieksplorasi, khususnya untuk mengevaluasi stabilitas subset fitur lintas fold yang berbeda. Selain itu, eksperimen dengan model yang lebih berat seperti XGBoost atau LightGBM dengan RFECV yang sama akan membantu menjawab apakah keunggulan yang diamati bersumber dari seleksi fitur, arsitektur classifier, atau kombinasi keduanya.

Terakhir, penerapan Random Undersampling dalam penelitian ini terbukti efektif namun berpotensi membuang informasi dari kelas mayoritas. Teknik oversampling seperti SMOTE-ENN atau Cluster-Based Oversampling dapat menjadi alternatif yang lebih konservatif terhadap data, dan perbandingan empiris antara berbagai strategi resampling pada dataset dengan ketidakseimbangan ekstrem seperti CIC-DDoS2019 akan menjadi kontribusi tersendiri bagi komunitas riset keamanan jaringan.

Daftar Pustaka

[1] H. Hartono, M. Khahfi Zuhanda, and S. Rahman, "IMPROVING CYBERSECURITY TRAFFIC ANALYSIS VIA ENHANCED K-MEANS CLUSTERING WITH TRIANGLE INEQUALITY-BASED INITIALIZATION," *J. TIMES*, vol. 14, no. 1, pp. 60–69, Jun. 2025, https://doi.org/10.51351/jtm.14.1.2025823

[2] O. Ebrahem, S. Dowaji, and S. Alhammoud, "A lightweight machine learning approach for DDoS detection and classification," *Sci. Rep.*, Apr. 2026, https://doi.org/10.1038/s41598-026-48535-x

[3] A. A. Alashhab *et al.*, "Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model," *IEEE Access*, vol. 12, pp. 51630–51649, 2024, https://doi.org/10.1109/ACCESS.2024.3384398

[4] A. A. Alashhab, M. S. M. Zahid, M. A. Azim, M. Y. Daha, B. Isyaku, and S. Ali, "A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks," *Symmetry (Basel).*, vol. 14, no. 8, p. 1563, Jul. 2022, https://doi.org/10.3390/sym14081563

[5] S. Batool, M. Aslam, E. Akpokodje, and S. F. Jilani, "A Comprehensive Review of DDoS Detection and Mitigation in SDN Environments: Machine Learning, Deep Learning, and Federated Learning Perspectives," *Electronics*, vol. 14,

no. 21, p. 4222, Oct. 2025, https://doi.org/10.3390/electronics14214222

[6] T. Ariyadi, A. R. Mukti, and H. Saputra, "Mitigasi Distributed Denial Of Service(DDoS) Attack Pada Arsitektur Software Defined Network (SDN)," *Techno.Com*, vol. 21, no. 4, pp. 878–886, Nov. 2022, https://doi.org/10.33633/tc.v21i4.6879

[7] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An Aggregated Mutual Information Based Feature Selection with Machine Learning Methods for Enhancing IoT Botnet Attack Detection," *Sensors*, vol. 22, no. 1, p. 185, Dec. 2021, https://doi.org/10.3390/s22010185.

[8] B. Liang, X. Dong, Y. Wang, and X. Zhang, "A high-applicability heterogeneous cloud data centers resource management algorithm based on trusted virtual machine migration," *Expert Syst. Appl.*, vol. 197, p. 116762, Jul. 2022, https://doi.org/10.1016/j.eswa.2022.116762

[9] H. Zouhri, A. Idri, and A. Ratnani, "Evaluating the impact of filter-based feature selection in intrusion detection systems," *Int. J. Inf. Secur.*, vol. 23, no. 2, pp. 759–785, Apr. 2024, https://doi.org/10.1007/s10207-023-00767-y

[10] G. S. Fuhnwi, M. Revelle, and C. Izurieta, "Improving Network Intrusion Detection Performance : An Empirical Evaluation Using Extreme Gradient Boosting (XGBoost) with Recursive Feature Elimination," in *2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC)*, IEEE, Feb. 2024, pp. 1–8. https://doi.org/10.1109/ICAIC60265.2024.10433805

[11] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J. Big Data*, vol. 10, no. 1, p. 15, Feb. 2023, https://doi.org/10.1186/s40537-023-00694-8

[12] E. M. Maseno and Z. Wang, "Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection," *J. Big Data*, vol. 11, no. 1, p. 24, Feb. 2024, https://doi.org/10.1186/s40537-024-00887-9

[13] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data," *Appl. Sci.*, vol. 14, no. 2, p. 479, Jan. 2024, https://doi.org/10.3390/app14020479

[14] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics*, vol. 14, no. 1, p. 69, Dec. 2024, https://doi.org/10.3390/electronics14010069

[15] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *J. Supercomput.*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023, https://doi.org/10.1007/s11227-023-05073-x

[16] S. A. Almahaqeri, M. H. Almourish, A. A. Nasser, A. S. A. Alghawli, A. A. K. Elsayed, and A. N. Alhejoj, "An optimized gradient boosting framework for IoT intrusion detection: a comprehensive evaluation on the CICIoT2023 dataset," *Sci. Rep.*, Apr. 2026, https://doi.org/10.1038/s41598-026-47399-5

[17] M. S. Sawah, H. Elmannai, A. A. El-Bary, K. Lotfy, and O. E. Sheta, "Distributed denial of service (DDoS) classification based on random forest model with backward elimination algorithm and grid search algorithm," *Sci. Rep.*, vol. 15, no. 1, p. 19063, May 2025, https://doi.org/10.1038/s41598-025-03868-x

[18] R. K. Batchu, T. Bikku, S. Thota, H. Seetha, and A. A. Ayoade, "A novel optimization-driven deep learning framework for the detection of DDoS attacks," *Sci. Rep.*, vol. 14, no. 1, p. 28024, Nov. 2024, https://doi.org/10.1038/s41598-024-77554-9

[19] Y. Kim, S. Seol, J. Chung, and H. Lee, "CRGAN-based turbo code interleaver for underwater acoustic communications," *ICT Express*, vol. 10, no. 3, pp. 498–506, Jun. 2024, https://doi.org/10.1016/j.icte.2024.01.005

[20] M. F. Saiyedand and I. Al-Anbagi, "Deep Ensemble Learning With Pruning for DDoS Attack Detection in IoT Networks," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 596–616, 2024, https://doi.org/10.1109/TMLCN.2024.3395419

[21] M. Alalhareth and S.-C. Hong, "An Improved Mutual Information Feature Selection Technique for Intrusion Detection Systems in the Internet of Medical Things," *Sensors*, vol. 23, no. 10, p. 4971, May 2023, https://doi.org/10.3390/s23104971

[22] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104–93139, 2022, https://doi.org/10.1109/ACCESS.2022.3204051