JIKTEKS : Jurnal Ilmu Komputer dan Teknologi Informasi
Vol. 04, No. 03, Agustus 2026, hlm. 151 - 159
ISSN : 2986-5417
DOI : doi.org/10.70404/jikteks.v4i03.726
Laman Jurnal: jurnal.faatuatua.com/index.php/JIKTEKS

Analisis Performa Algoritma Decision Tree dengan Correlation-based Feature Selection (CFS) untuk Deteksi Serangan Jaringan

Pandu Satria[1*], Wicaksono Yuli Sulistyo[2]
[1*, 2]Sistem Informasi, Universitas Siber Muhammadiyah, Indonesia
e-mail: [1*]*pandu20210200004@sibermu.ac.id*
e-mail: [2]*wicaksono@sibermu.ac.id*

Keywords:

Decision Tree;
CFS;
Network Intrusion Detection;
CICIDS2017;
Efficiency–Accuracy Trade-off.

ABSTRACT

Network security requires Intrusion Detection Systems (IDS) capable of detecting attacks accurately and efficiently. The high dimensionality of network data can increase model complexity and computational overhead. This study evaluates the effect of Correlation-based Feature Selection (CFS) on the performance of a Decision Tree algorithm using the NSL-KDD and CICIDS2017 datasets. Models without feature selection were used as baselines and compared with CFS-based models using Accuracy, Precision, Recall, F1-Score, ROC-AUC, and training time. The results show that CFS substantially reduces the number of features and training time across both datasets, but at the cost of reduced classification performance, particularly Recall on NSL-KDD. The Wilcoxon Signed-Rank test with 10-Fold Cross-Validation indicates a statistically significant difference (p = 0.001953). These findings demonstrate a trade-off between computational efficiency and detection capability, suggesting that the application of CFS should be aligned with the operational priorities of IDS deployment.

Kata Kunci:

Decision Tree;
CFS;
Deteksi Intrusi Jaringan;
CICIDS2017;
Tradeoff Efisiensi-Akurasi.

ABSTRAK

Keamanan jaringan membutuhkan Intrusion Detection System (IDS) yang mampu mendeteksi serangan secara akurat dan efisien. Tingginya jumlah fitur pada data jaringan dapat meningkatkan kompleksitas model dan beban komputasi. Penelitian ini mengevaluasi pengaruh Correlation-based Feature Selection (CFS) terhadap kinerja algoritma Decision Tree pada dataset NSL-KDD dan CICIDS2017. Model tanpa seleksi fitur digunakan sebagai baseline dan dibandingkan dengan model berbasis CFS menggunakan metrik Accuracy, Precision, Recall, F1-Score, ROC-AUC, serta waktu pelatihan. Hasil menunjukkan bahwa CFS secara signifikan mengurangi jumlah fitur dan waktu pelatihan pada kedua dataset, tetapi berdampak pada penurunan performa klasifikasi, terutama Recall pada NSL-KDD. Uji Wilcoxon Signed-Rank pada 10-Fold Cross-Validation menunjukkan adanya perbedaan yang signifikan (p = 0,001953). Temuan ini menunjukkan adanya trade-off antara efisiensi komputasi dan kemampuan deteksi, sehingga penerapan CFS perlu disesuaikan dengan prioritas sistem IDS.

Korespondensi Penulis *):

Pandu Satria
Universitas Siber Muhammadiyah
Jl. HOS Cokroaminoto No. 17, Pakuncen, Wirobrajan, Kota Yogyakarta, D.I. Yogyakarta

Diajukan: 10-05-2026 | Direvisi: 11-06-2026 | Diterima: 18-08-2027 | Diterbitkan: Agustus 2026

1. PENDAHULUAN

Perkembangan teknologi jaringan meningkatkan kompleksitas ancaman siber, sehingga diperlukan mekanisme deteksi yang efektif untuk mengidentifikasi aktivitas berbahaya [1,2]. Intrusion Detection System (IDS), khususnya Network Intrusion Detection System (NIDS), berperan dalam memantau lalu lintas jaringan dan mendeteksi aktivitas abnormal secara real-time. Pendekatan machine learning semakin banyak digunakan karena mampu mengenali pola serangan yang lebih beragam dibandingkan metode berbasis aturan [3], sementara integrasi Large Language Model (LLM) mulai mendukung otomatisasi analisis keamanan jaringan [4]. Salah satu algoritma yang banyak digunakan adalah Decision Tree karena memiliki interpretabilitas dan efisiensi komputasi yang baik [7].

Namun, dataset IDS seperti NSL-KDD dan CICIDS2017 memiliki dimensi fitur yang tinggi, yang dapat meningkatkan kompleksitas komputasi dan risiko overfitting. Feature selection dapat mengurangi fitur yang tidak

relevan dengan tetap mempertahankan informasi penting untuk klasifikasi [8,9]. Salah satu metode yang digunakan adalah Correlation-based Feature Selection (CFS), yang memilih fitur berdasarkan relevansi terhadap kelas dan rendahnya redundansi antarfitur [7]. Studi sebelumnya menunjukkan bahwa CFS dan variannya dapat meningkatkan efisiensi serta beberapa metrik performa klasifikasi [10,11,12,13]. Namun, sebagian besar penelitian masih berfokus pada satu dataset sehingga tradeoff antara reduksi fitur, performa, efisiensi komputasi, dan stabilitas hasil belum dikaji secara komprehensif [8,11]. Sebagai upaya mengatasi kesenjangan tersebut, penelitian ini mengevaluasi CFS pada Decision Tree menggunakan dataset NSL-KDD dan CICIDS2017 berdasarkan performa klasifikasi, efisiensi komputasi, dan stabilitas seleksi fitur. Penelitian ini memberikan bukti empiris mengenai trade-off antara reduksi fitur, performa deteksi, dan efisiensi komputasi dalam pengembangan NIDS yang lebih efisien.

2. METODE PENELITIAN

2.1 Dataset

Penelitian ini menggunakan dua dataset benchmark yang memiliki karakteristik berbeda sebagaimana disajikan pada Tabel 1.

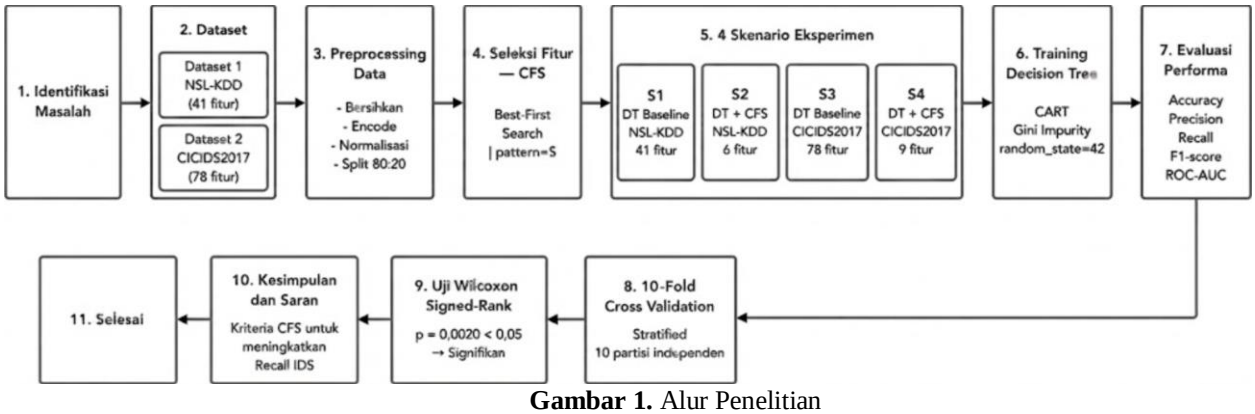
Tabel 1. Karakteristik Dataset

Dataset	Tahun	Fitur	Sampel Total	Normal	Serangan
NSL-KDD	1999/2009	41	148.517	77.054 (51,9%)	71.463 (48,1%)
CICIDS2017	2017	78	692.703	440.031 (63,5%)	252.672 (36,5%)

NSL-KDD dipilih karena ketersediaan ratusan baseline perbandingan dalam literatur IDS. CICIDS2017 dipilih karena merepresentasikan serangan modern (DoS Hulk, DDoS, PortScan, Brute Force, Bot) dengan fitur statistik aliran jaringan yang lebih kaya [12] . Penggunaan dua dataset berbeda era memungkinkan analisis generalisasi temuan lintas generasi dataset.

2.2 Alur Penelitian

Gambar 1 menunjukkan alur penelitian yang digunakan untuk mengevaluasi pengaruh penerapan *Correlation-based Feature Selection* (CFS) terhadap performa algoritma *Decision Tree* pada sistem deteksi intrusi jaringan. Alur penelitian disusun secara sistematis mulai dari identifikasi permasalahan hingga penarikan kesimpulan.



Gambar 1. Alur Penelitian

Proses penelitian diawali dengan identifikasi masalah, dilanjutkan dengan pengumpulan dan praproses dataset. Tahap berikutnya meliputi seleksi fitur, penyusunan skenario eksperimen, pelatihan model, dan evaluasi performa. Hasil eksperimen kemudian divalidasi menggunakan *cross validation* dan uji statistik sebelum dilakukan analisis untuk menghasilkan kesimpulan dan rekomendasi penelitian.

2.3 Preprocessing

Preprocessing dilakukan dalam lima tahap. Pertama, pembersihan data: mengganti nilai tak terbatas (inf/-inf) dengan NaN kemudian menghapus baris yang mengandung NaN, serta menghapus duplikat. Kedua, encoding fitur kategorik menggunakan Label Encoding pada tiga fitur string NSL-KDD (protocol_type, service, flag). Ketiga, binarisasi label: kelas normal dikodekan 0 dan semua jenis serangan dikodekan 1. Keempat, pembagian data menggunakan stratified split 80:20 untuk mempertahankan proporsi kelas. Kelima, normalisasi Min-Max Scaling ke rentang [0,1] yang di-fit hanya pada data training untuk mencegah data leakage[13].

Tabel 2. Ringkasan Preprocessing dan Pembagian Data

Dataset	Sampel Train (80%)	Sampel Test (20%)	Fitur Awal
NSL-KDD	118.813	29.704	41
CICIDS2017	488.393	122.099	78

Tabel 2 menunjukkan ringkasan tahap preprocessing berupa pembagian dataset menjadi data latih (80%) dan data uji (20%). Pada dataset NSL-KDD, total data setelah preprocessing menghasilkan 118.813 sampel untuk training dan 29.704 sampel untuk testing dengan 41 fitur awal. Sementara itu, dataset CICIDS2017 memiliki jumlah data yang lebih besar, yaitu 488.393 sampel training dan 122.099 sampel testing dengan 78 fitur awal. Pembagian ini bertujuan untuk memastikan model dapat dilatih secara optimal serta dievaluasi secara objektif pada data yang belum pernah dilihat sebelumnya[5].

2.4 Implementasi Correlation-based Feature Selection (CFS)

CFS diimplementasikan secara kustom menggunakan Python dengan korelasi Pearson untuk menghitung merit dan algoritma *Best First Search* dengan parameter patience=5. Proses seleksi hanya dilakukan pada data training untuk mencegah kebocoran informasi, kemudian fitur terpilih diterapkan secara konsisten pada data test [7]. Stabilitas seleksi dievaluasi melalui analisis sensitivitas patience dengan rentang nilai 1–20. Seluruh konfigurasi menghasilkan subset identik yang terdiri atas enam fitur dengan Recall konsisten sebesar 90,21%. Hasil tersebut menunjukkan bahwa proses konvergensi CFS pada NSL-KDD bersifat deterministik dan tidak sensitif terhadap perubahan parameter patience, tetapi lebih dipengaruhi oleh struktur korelasi intrinsik dataset. Dengan demikian, penurunan Recall yang diamati tidak disebabkan oleh konfigurasi parameter, melainkan oleh karakteristik informasi diskriminatif pada fitur yang dipertahankan.

2.5 Desain Eksperimen

Empat skenario dibandingkan: (S1) DT Baseline NSL KDD semua 41 fitur; (S2) DT CFS NSL KDD subset fitur terpilih CFS; (S3) DT Baseline CICIDS2017 semua 78 fitur; (S4) DT+CFS CICIDS2017 subset fitur terpilih CFS. Decision Tree menggunakan implementasi CART dari Scikit-learn (criterion='gini', random_state=42). Validasi menggunakan 10-Fold Stratified Cross Validation. Signifikansi statistik diuji menggunakan Wilcoxon Signed-Rank Test (alpha = 0,05).

2.6 Matrik Evaluasi

Performa dievaluasi menggunakan Accuracy (proporsi prediksi benar), Precision (proporsi true positive dari prediksi positif), Recall (proporsi serangan yang berhasil terdeteksi — metrik terkritis untuk IDS), F1-Score (rata-rata harmonik precision dan recall), dan ROC-AUC. Efisiensi komputasi diukur melalui training time dan tree depth.

3. HASIL DAN ANALISIS

3.1 Hasil Seleksi CFS

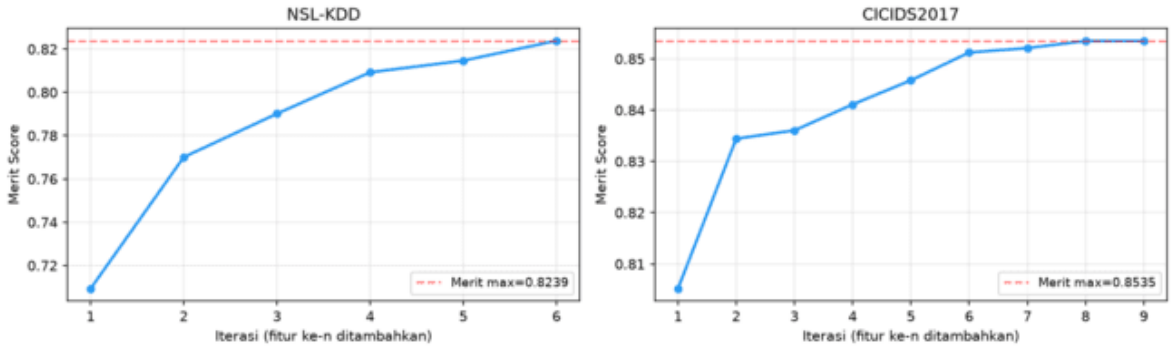
Metode Correlation-based Feature Selection (CFS) diterapkan untuk mengidentifikasi subset fitur yang memiliki relevansi tinggi terhadap kelas target dengan redundansi yang rendah antarfitur. Tujuan utama tahap ini adalah mengurangi dimensi data sehingga proses klasifikasi menjadi lebih efisien tanpa menghilangkan informasi penting yang dibutuhkan untuk mendeteksi serangan jaringan.

Tabel 3. Ringkasan Hasil Seleksi Fitur CFS

Dataset	Fitur Awal	Fitur Terpilih	Reduksi (%)	Merit Akhir	Waktu CFS
NSL KDD	41	6	85,4%	0,8239	±3 detik
CICIDS2017	78	9	88,5%	0,8535	±120 detik

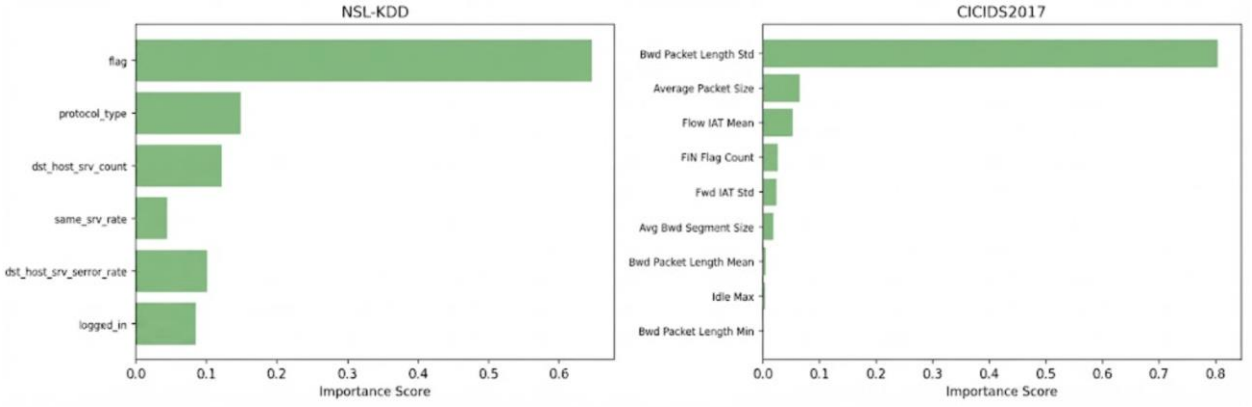
Tabel 3 menunjukkan hasil seleksi fitur menggunakan CFS pada kedua dataset. CFS menghasilkan reduksi fitur sebesar 85,4% pada NSL KDD dan 88,5% pada CICIDS2017, lebih tinggi dibandingkan penelitian sebelumnya yang umumnya melaporkan reduksi sekitar 40 - 60% pada dataset IDS [7,11]. Hasil tersebut menunjukkan efektivitas CFS dalam mengurangi fitur redundan dengan tetap mempertahankan representasi data yang relevan, ditunjukkan oleh nilai merit yang tinggi dan waktu komputasi yang relatif efisien. Temuan ini konsisten dengan penelitian sebelumnya bahwa seleksi fitur berbasis korelasi dapat meningkatkan efisiensi komputasi sekaligus mempertahankan kemampuan diskriminatif fitur pada IDS berbasis machine learning [7].

Fitur yang dipilih oleh CFS cenderung memiliki korelasi yang lebih tinggi terhadap kelas dibandingkan fitur yang dieliminasi. Hal ini menunjukkan bahwa proses seleksi berhasil mempertahankan fitur-fitur yang paling relevan untuk mendukung proses klasifikasi. Setelah korelasi fitur dievaluasi, proses pencarian subset fitur terbaik dilakukan menggunakan mekanisme Best-First Search yang mengevaluasi nilai merit pada setiap iterasi. Perkembangan nilai merit selama proses seleksi ditunjukkan pada Gambar 2.



Gambar 2. ROC Curve Decision Tree Sebelum dan Sesudah CFS

Gambar 2 menunjukkan nilai merit meningkat secara bertahap pada kedua dataset hingga mencapai kondisi konvergen. Pada NSL-KDD, peningkatan terbesar terjadi di iterasi awal sebelum mencapai nilai maksimum 0,8239 dengan enam fitur terpilih, sementara CICIDS2017 meningkat konsisten hingga mencapai 0,8535 dengan sembilan fitur terpilih. Setelah titik tersebut, penambahan fitur baru tidak lagi memberikan kontribusi signifikan terhadap kualitas subset. Pola konvergensi ini menunjukkan bahwa sebagian besar informasi diskriminatif dapat direpresentasikan oleh sedikit fitur berkorelasi tinggi. Stabilitas nilai merit di akhir iterasi mengindikasikan pencarian telah menemukan subset optimal, sehingga eksplorasi fitur tambahan tidak lagi diperlukan [7],11]. Kontribusi masing-masing fitur hasil seleksi terhadap proses klasifikasi dianalisis menggunakan metode feature importance pada algoritma Decision Tree. Hasil analisis disajikan pada Gambar 3



Gambar 3. Feature Importance Fitur Terpilih Hasil CFS

Gambar 3 menunjukan bahwa tidak semua fitur terpilih berkontribusi sama terhadap proses klasifikasi, di mana beberapa fitur menunjukkan nilai importance dominan dalam pembentukan struktur pohon keputusan. Kondisi ini membuktikan bahwa sebagian besar kemampuan diskriminatif model berasal dari sejumlah kecil atribut yang efektif membedakan lalu lintas normal dan serangan. Hasil tersebut memperkuat efektivitas CFS dalam mempertahankan fitur relevan serta mengeliminasi atribut redundan berkontribusi rendah. Dengan demikian, reduksi fitur yang diperoleh berhasil meningkatkan efisiensi komputasi sekaligus menghasilkan model yang lebih sederhana dan mudah diinterpretasikan [7,11].

3.2 Perbandingan Performa Klasifikasi

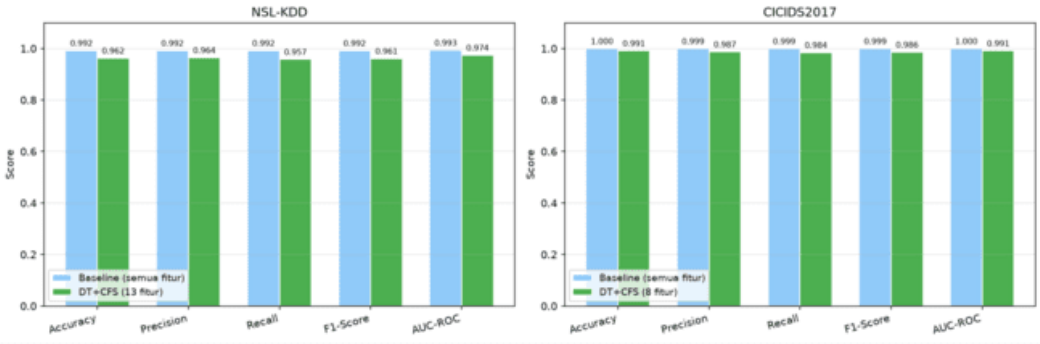
Setelah proses seleksi fitur menggunakan CFS selesai dilakukan, tahap berikutnya adalah mengevaluasi pengaruh reduksi fitur terhadap performa klasifikasi menggunakan algoritma Decision Tree. Evaluasi dilakukan dengan membandingkan model baseline (tanpa seleksi fitur) dan model yang menggunakan fitur hasil CFS pada dataset NSL-KDD dan CICIDS2017. Metrik yang digunakan meliputi accuracy, precision, recall, F1-score, ROC-AUC, kedalaman pohon keputusan *(tree depth)*, serta waktu pelatihan model. Tabel 5 menyajikan perbandingan lengkap keempat skenario eksperimen.

Tabel 4. Perbandingan Performa Decision Tree (Baseline vs CFS)

Skenario	Fitur	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Depth	Train (s)
DT Baseline NSL-KDD	41	99,22%	99,18%	99,20%	99,19%	99,34%	38	1,0512
DT+CFS NSL-KDD	6	94,40%	98,00%	90,21%	93,95%	97,60%	28	0,1500
DT Baseline CICIDS2017	78	99,96%	99,94%	99,93%	99,93%	99,95%	37	25,8805
DT+CFS CICIDS2017	9	98,88%	99,80%	96,67%	98,21%	99,79%	31	3,3181

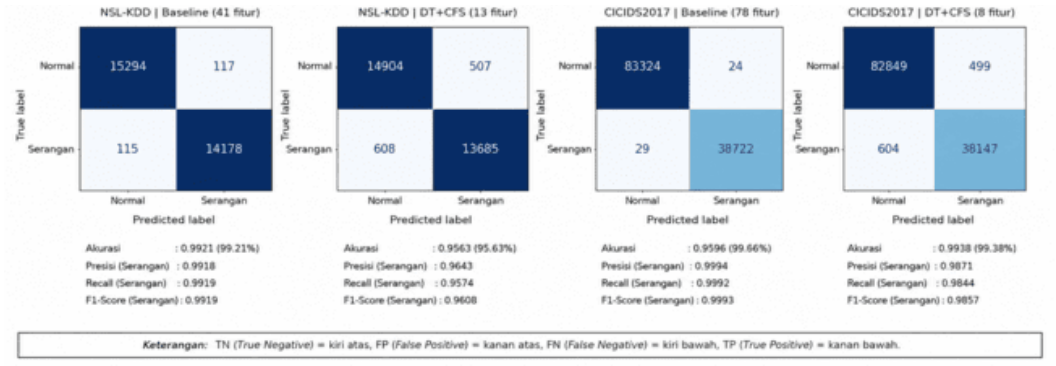
Berdasarkan Tabel 4, model baseline menghasilkan performa klasifikasi sangat tinggi, dengan accuracy 99,22% dan recall 99,20% pada NSL-KDD, serta accuracy 99,96% dan recall 99,93% pada CICIDS2017. Setelah penerapan CFS, jumlah fitur berkurang signifikan dari 41 menjadi 6 pada NSL-KDD dan dari 78 menjadi 9 pada CICIDS2017. Reduksi ini diikuti penurunan performa, terutama pada recall NSL-KDD yang turun menjadi 90,21%. Meski fiturnya berkurang lebih dari 85%, model hasil seleksi tetap kompetitif dengan accuracy 98,88% dan ROC-AUC 99,79% pada CICIDS2017, serta accuracy 94,40% dan precision 98,00% pada NSL-KDD.

Selain itu, CFS meningkatkan efisiensi komputasi dengan mengurangi kedalaman pohon keputusan (dari 38 menjadi 28 pada NSL-KDD; 37 menjadi 31 pada CICIDS2017) dan memangkas waktu pelatihan secara drastis (dari 1,0512 menjadi 0,1500 detik pada NSL-KDD; 25,8805 menjadi 3,3181 detik pada CICIDS2017). Temuan ini membuktikan bahwa reduksi fitur menghasilkan model yang lebih sederhana dan cepat tanpa menghilangkan sebagian besar kemampuan deteksinya, sejalan dengan penelitian terdahulu [7,11]. Perubahan performa komparatif dari hasil evaluasi ini divisualisasikan secara lebih jelas pada Gambar 4.



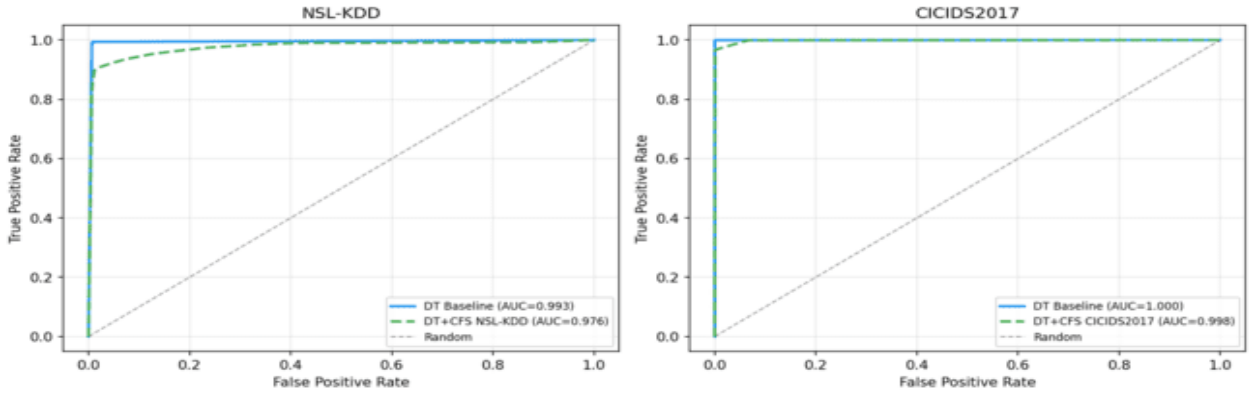
Gambar 4. Perbandingan Accuracy, Precision, Recall, F1-Score, dan ROC-AUC

Gambar 4 menunjukkan bahwa penerapan CFS hanya menyebabkan penurunan kecil pada sebagian besar metrik evaluasi. Penurunan terbesar terjadi pada *recall* dataset NSL-KDD, sedangkan metrik lainnya tetap berada pada tingkat yang tinggi. Dataset CICIDS2017 mempertahankan performa yang sangat baik meskipun jumlah fitur berkurang secara signifikan. Temuan ini mengindikasikan bahwa efektivitas CFS dipengaruhi oleh karakteristik dataset. Analisis *confusion matrix* dilakukan untuk memberikan gambaran yang lebih rinci mengenai pola kesalahan klasifikasi setelah reduksi fitur. Visualisasi hasil klasifikasi pada setiap skenario disajikan pada Gambar 5.



Gambar 5. Confusion Matrix Decision Tree Sebelum dan Sesudah CFS

Berdasarkan Gambar 5, sebagian besar sampel berhasil diklasifikasikan dengan benar pada kedua model. Namun, penerapan CFS meningkatkan jumlah False Negative pada NSL-KDD, sehingga Recall menurun dari 99,20% menjadi 90,21%. Hal ini menunjukkan bahwa sebagian fitur yang dieliminasi masih mengandung informasi penting untuk mendeteksi variasi serangan, terutama karena CFS berbasis pada korelasi linier antarfitur [7]. Sebaliknya, hasil pada CICIDS2017 relatif stabil dengan jumlah False Negative yang tetap rendah, menunjukkan bahwa fitur terpilih masih mampu merepresentasikan karakteristik serangan secara efektif [11]. Selanjutnya, kemampuan model dalam membedakan trafik normal dan serangan dievaluasi menggunakan kurva Receiver Operating Characteristic (ROC), yang menunjukkan hubungan antara True Positive Rate (TPR) dan False Positive Rate (FPR). Kurva yang semakin mendekati sudut kiri atas menunjukkan kemampuan klasifikasi yang semakin baik.



Gambar 6. ROC Curve Decision Tree Sebelum dan Sesudah CFS.

Berdasarkan Gambar 6, seluruh model menghasilkan kurva ROC yang mendekati sudut kiri atas, menunjukkan kemampuan klasifikasi yang sangat baik. Nilai AUC pada model baseline mencapai 99,34% untuk NSL-KDD dan 99,95% untuk CICIDS2017. Setelah penerapan CFS, nilai AUC hanya menurun menjadi 97,60% dan 99,79%, sehingga kemampuan model dalam membedakan kelas tetap tinggi meskipun jumlah fitur berkurang secara signifikan.

Penurunan recall pada NSL-KDD tidak diikuti penurunan AUC yang sebanding, mengindikasikan bahwa model masih memiliki kemampuan separasi kelas yang baik, tetapi sensitivitas terhadap sebagian sampel serangan menurun setelah seleksi fitur. Temuan ini menunjukkan bahwa evaluasi performa tidak dapat bergantung pada satu metrik saja. Kombinasi hasil ROC dan confusion matrix memberikan gambaran yang lebih komprehensif mengenai trade-off antara reduksi fitur, performa deteksi, dan efisiensi komputasi CFS [7,11].

3.3 Analisis Tradeoff dan Validasi Statistik

Selain mengevaluasi performa klasifikasi, penelitian ini juga menganalisis tradeoff antara efisiensi komputasi dan kemampuan deteksi yang dihasilkan oleh penerapan CFS. Analisis ini penting karena keberhasilan seleksi fitur tidak hanya ditentukan oleh tingkat akurasi yang dicapai, tetapi juga oleh kemampuan metode dalam mengurangi kompleksitas model dan kebutuhan komputasi. Ringkasan tradeoff yang diperoleh pada kedua dataset disajikan pada Tabel 8.

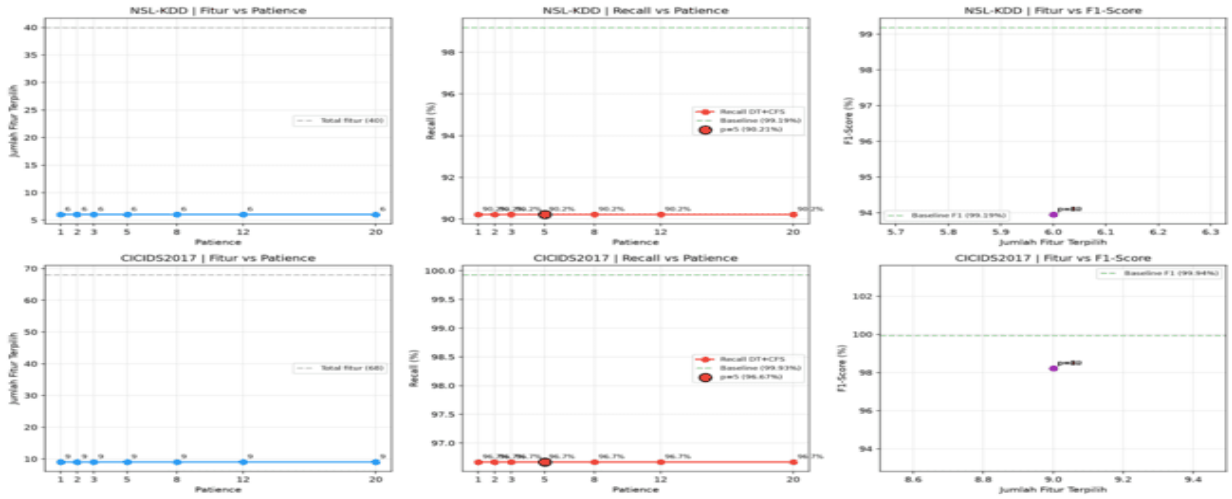
Tabel 5. Analisis Tradeoff Lengkap: Efisiensi vs Akurasi

Aspek	DT Baseline NSL-KDD	DT+CFS NSL-KDD	DT Baseline CICIDS2017	DT+CFS CICIDS2017
Jumlah Fitur	41	6 (↓85,4%)	78	9 (↓88,5%)
Accuracy	99,22%	94,40%	99,96%	98,88%
Precision	99,18%	98,00%	99,94%	99,80%
Recall	99,20%	90,21%	99,93%	96,67%
F1-Score	99,19%	93,95%	99,93%	98,21%
ROC-AUC	99,34%	97,60%	99,95%	99,79%
Train Time	1,0512 s	0,1500 s	25,8805 s	3,3181 s
Tree Depth	38	28	37	31

Berdasarkan Tabel 5, CFS meningkatkan efisiensi komputasi dengan menurunkan waktu pelatihan sebesar 85,7% pada NSL-KDD dan 87,2% pada CICIDS2017. Kedalaman pohon juga berkurang masing-masing sebesar 26,3% dan 16,2%, menunjukkan bahwa reduksi fitur menyederhanakan struktur model dan mempercepat pelatihan tanpa optimasi kompleks [7,11]. Namun, peningkatan efisiensi diikuti penurunan performa, terutama Recall yang turun 8,99 poin persentase pada NSL-KDD dan 3,26 poin persentase pada CICIDS2017. Perbedaan tersebut menunjukkan bahwa dampak seleksi fitur dipengaruhi oleh karakteristik dataset dan distribusi informasi atribut [15]. Peningkatan False Negative pada NSL-KDD mengindikasikan bahwa beberapa fitur yang dieliminasi kemungkinan mengandung informasi komplementer yang bersifat non-linear atau kontekstual [7,15]. Sebaliknya, fitur terpilih pada CICIDS2017 tetap mampu merepresentasikan karakteristik serangan secara efektif. Hal ini menunjukkan bahwa fitur statistik aliran jaringan pada CICIDS2017 memiliki kemampuan diskriminatif yang lebih kuat terhadap pola serangan dibandingkan atribut pada NSL-KDD [13].

3.3.1 Analisis Sensitivitas Parameter Patience

Stabilitas proses seleksi fitur dievaluasi melalui uji sensitivitas terhadap parameter patience dengan memvariasikan nilainya pada rentang 1 hingga 20.



Gambar 7. Analisis sensitivitas parameter patience CFS

Berdasarkan Gambar 7, nilai Recall stabil di kisaran 90,2% untuk NSL-KDD dan 96,7% untuk CICIDS2017, begitu pula dengan F1-Score yang tidak mengalami perubahan berarti di seluruh konfigurasi patience. Temuan ini mengindikasikan bahwa kualitas subset fitur lebih dipengaruhi oleh karakteristik korelasi antarfitur dibandingkan durasi pencarian algoritma. Oleh karena itu, konfigurasi patience=5 dalam penelitian ini sudah memadai sebagai mekanisme early stopping untuk menjaga efisiensi komputasi tanpa memengaruhi performa model. Hal ini sejalan dengan penelitian terdahulu yang menunjukkan bahwa seleksi fitur berbasis korelasi cenderung menghasilkan subset yang stabil ketika struktur korelasi data telah terdefinisi dengan baik [7,11].

3.3.2 Validasi Statistik Menggunakan Wilcoxon Signed-Rank Test

Pengujian statistik menggunakan Wilcoxon Signed-Rank Test dilakukan untuk memastikan bahwa perbedaan performa antara model Decision Tree baseline dan Decision Tree dengan Correlation-based Feature Selection (DT+CFS) tidak disebabkan oleh variasi pembagian data selama 10-Fold Stratified Cross Validation. Uji Wilcoxon dipilih karena tidak mengasumsikan distribusi normal dan direkomendasikan untuk membandingkan performa dua model klasifikasi pada data berpasangan hasil cross validation [10,16].

Tabel 6. Hasil Wilcoxon Signed-Rank Test pada F1-Score 10-Fold Cross Validation

Dataset	Mean F1 Baseline	Mean F1 DT+CFS	W	p-value	Keputusan
NSL-KDD	0,9919 ± 0,0005	0,9389 ± 0,0017	0,000	0,001953	Signifikan
CICIDS2017	0,9993 ± 0,0001	0,9809 ± 0,0008	0,000	0,001953	Signifikan

Tabel 6 menunjukkan bahwa seluruh pengujian menghasilkan p-value = 0,001953 (α = 0,05), sehingga H_0 ditolak. Hal ini membuktikan bahwa penerapan CFS menghasilkan perubahan performa yang signifikan secara statistik pada kedua dataset. Nilai W = 0 menunjukkan arah perubahan yang konsisten pada seluruh fold. Pada NSL-KDD, rata-rata F1-Score menurun dari 0,9919 menjadi 0,9389 setelah reduksi fitur dari 41 menjadi 6, sedangkan pada CICIDS2017 menurun dari 0,9993 menjadi 0,9809 setelah reduksi dari 78 menjadi 9 fitur. Temuan ini menunjukkan

bahwa penurunan performa berkaitan dengan reduksi dimensi, bukan semata-mata variasi pembagian data. Di sisi lain, reduksi fitur meningkatkan efisiensi komputasi lebih dari 85%, sehingga menunjukkan trade-off yang jelas antara efisiensi dan performa klasifikasi. Hasil ini konsisten dengan penelitian sebelumnya bahwa reduksi dimensi yang agresif dapat menurunkan kemampuan diskriminatif model meskipun memberikan keuntungan komputasi [7,15].

4. KESIMPULAN

Penelitian ini menunjukkan bahwa CFS mampu mengurangi jumlah fitur secara signifikan dan menurunkan waktu pelatihan pada NSL-KDD dan CICIDS2017. Namun, efisiensi tersebut disertai penurunan Recall, terutama pada NSL-KDD. Penurunan ini menunjukkan bahwa beberapa fitur yang dihapus kemungkinan memiliki informasi diskriminatif penting untuk membedakan lalu lintas normal dan serangan. Dampaknya, sebagian serangan dapat lebih sulit dikenali oleh Decision Tree, terutama pada dataset dengan karakteristik fitur yang lebih kompleks. Hal ini menegaskan bahwa efektivitas CFS bergantung pada karakteristik dataset dan kebutuhan IDS. Meskipun CFS meningkatkan efisiensi komputasi, reduksi fitur yang terlalu agresif dapat meningkatkan risiko false negative dan menurunkan kemampuan deteksi serangan. Penelitian selanjutnya dapat menguji generalisasi menggunakan dataset UNSW-NB15 dan CICIDS2018, serta membandingkan CFS dengan metode non-linear seperti Mutual Information dan Random Forest Importance. Kombinasi CFS dengan algoritma ensemble seperti Random Forest dan XGBoost juga dapat dievaluasi untuk mengkaji pengaruh reduksi fitur terhadap deteksi berbagai jenis serangan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan dan kontribusi dalam pelaksanaan penelitian, khususnya dalam penyediaan data, proses eksperimen, analisis, dan penyusunan artikel ini.

REFERENSI

[1] L. Iriani, M. N. Hafizh, and K. E. Setyaputri, "Analisis Forensik Jaringan Serangan ARP Spoofing Menggunakan Metode National Institute of Justice (NIJ)," IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi, vol. 4, no. 2, pp. 150-160, 2025, doi: 10.24246/itexplore.v4i2.2025.pp150-160.

[2] I. Anshori, K. E. S. Putri, and U. Ghoni, "Analisis Barang Bukti Digital Aplikasi Facebook Messenger pada Smartphone Android Menggunakan Metode NIJ," IT Journal Research and Development, vol. 5, no. 2, pp. 118-134, 2021, doi: 10.25299/itjrd.2021.vol5(2).4664.

[3] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," IEEE Access, vol. 11, pp. 80015-80034, 2023, doi: 10.1109/ACCESS.2023.3296444.

[4] W. Y. Sulistyo and J. Supriyanto, "Integrasi Moodle API dan LLM dalam Otomasi Monitoring Capaian Pembelajaran Lulusan Berbasis OBE," JITU: Journal Informatic Technology and Communication, vol. 10, no. 1, pp. 172-182, 2026, doi: 10.36596/jitu.v10i1.2307.

[5] N. F. Rozam, T. N. Sari, M. R. A. Yudianto, and D. F. Rahman, "Deteksi Anomali Lalu Lintas Jaringan Berbasis Machine Learning Menggunakan Dataset CIC-IDS2017," UPGRADE: Jurnal Pendidikan Teknologi Informasi, vol. 3, no. 2, 2026, doi: 10.30812/upgrade.v3i2.6174.

[6] S. Kumar and A. Sharma, "Enhancing IoT Data Analysis with Machine Learning: A Comprehensive Overview," International Journal for Multidisciplinary Research (IJFMR), 2024.

[7] Y. G. Damtew, H. Chen, and Z. Yuan, "Heterogeneous Ensemble Feature Selection for Network Intrusion Detection System," International Journal of Computational Intelligence Systems, vol. 16, no. 1, Art. no. 9, 2023, doi: 10.1007/s44196-022-00174-6.

[8] J. Manurung, A. Mardamsyah, and B. Sianipar, "Enhanced Cyber Attack Detection Using Optimized Random Forest with SMOTE-Based Class Balancing and Feature Selection," Journal of Defense Technology and Engineering, 2026.

[9] F. Ahmad et al., "A Comprehensive Review of Feature Selection Methods in Financial Machine Learning Applications," IEEE Access, 2023.

[10] R. Panigrahi, S. Borah, A. K. Bhoi, M. F. Ijaz, M. Pramanik, Y. Kumar, and R. R. Jha, "A Performance Evaluation of Supervised Machine Learning Algorithms for Network Intrusion Detection," IEEE Access, vol. 9, pp. 83701-83713, 2021, doi: 10.1109/ACCESS.2021.3087593.

[11] M. Di Mauro, G. Galatro, and F. Postiglione, "Feature Selection for Network Intrusion Detection Systems: A Systematic Review and Comparative Study," IEEE Access, vol. 9, pp. 127265-127284, 2021, doi: 10.1109/ACCESS.2021.3112444.

[12] S. Bhaskara, B. Sriman, and K. V. S. N. Murthy, "Detection of DDoS Attacks Using Correlation-Based Feature Selection and Ensemble Learning in CSE-CIC-IDS2018 Dataset," International Journal of Information Security and Privacy, vol. 16, no. 1, pp. 1-18, 2022, doi: 10.4018/IJISP.292150.

[13] S. Gu and X. Lu, "An Adaptive Correlation-Based Feature Selection Method for Network Intrusion Detection System," Journal of Cloud Computing, vol. 10, no. 1, pp. 1-14, 2021, doi: 10.1186/s13677-021-00251-x.

[14] K. Kurniabudi, D. Stiawan, A. Darmawahyuni, A. Hanis, and R. Budiarto, "CICIDS2017 Dataset Feature Selection Using Random Forest," JOIV: International Journal on Informatics Visualization, vol. 5, no. 2, pp. 160-165, 2021, doi: 10.30630/joiv.5.2.551.

[15] A. Maulana, S. Anam, and H. A. Bukhori, "Improving Lateral-Movement Intrusion Detection in Virtualized Networks Using SHAP Feature Selection, SMOTE, and a Voting Ensemble Classifier," Jurnal Teknik Informatika (JUTIF), vol. 6, no. 4, 2025, doi: 10.52436/1.jutif.2025.6.4.5233.

[16] D. J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 6th ed. Boca Raton, FL, USA: CRC Press, 2020.