Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893

Model Deteksi Serangan Jaringan Menggunakan *Machine Learning* Dengan Teknik *Ensemble Learning*

DOI: http://dx.doi.org/10.35889/jutisi.v15i1.3369

Christopher Michael Lauw[1*], Anthony Anggrawan[2], Neny Sulistianingsih[3]
Ilmu Komputer, Universitas Bumigora, Mataram, Indonesia
*e-mail *Corresponding Author*: michmogi1388@gmail.com

Abstrack

Network attacks pose a serious threat to the security of digital infrastructure, making accurate and efficient detection an urgent necessity. The National Cyber and Crypto Agency (BSSN) recorded 330,527,636 instances of anomalous traffic or network attacks. This sheer volume renders manual handling by cybersecurity personnel highly impractical. Consequently, a detection system based on Machine Learning (ML) is required. However, single machin e learning models have proven suboptimal in effectively resolving this complex problem. This research aims to develop a robust Network Intrusion Detection Model by employing advanced Ensemble Learning techniques—specifically Bagging, Boosting, and Stacking—to significantly optimize the performance of the base ML models. The study utilizes the UNR-IDD dataset. The methodology begins with comprehensive Data Preprocessing, including Min-Max Scaling and Dimensionality Reduction. Model performance is comprehensively evaluated using classification metrics across three distinct data splitting scenarios (70:30, 80:20, and 90:10) to identif y the optimal configuration. The experimental results demonstrate that the Stacking Ensemble Learning approach achieves the most optimal accuracy among the tested methods.

***Keywords**: Network Intrusion Detection; Machine Learning; Ensemble Learning; Stacking; Dimensionality Reduction*

Abstrak

Serangan jaringan telah menjadi ancaman serius bagi keamanan infrastruktur digital. Badan Siber dan Sandi Negara (BSSN) mencatat terdapat 330.527.636 kasus traffic anomali atau serangan jaringan. Hal tersebut sangat tidak memungkinkan personal keamanan siber untuk menanganinya. Maka, dibutuhkan sistem deteksi berbasis *Machine Learning* . model *Machine Learning* tunggal belum optimal dalam deteksi serangan jaringan. Penelitian ini bertujuan untuk memodelkan deteksi serangan jaringan menggunakan *Machine Learning* dengan teknik *ensemble learning bagging, boosting, dan stacking,* guna untuk mengoptimalkan perf orma model *Machine Learning*. Penelitian ini menggunakan dataset UNR-IDD. Penelitian ini bertujuan mengembangkan Model Deteksi Serangan Jaringan yang robust menggunakan teknik Ensemble Learning. Proses dimulai dengan Data Preprocessing, meliputi Min-Max Scaling dan Reduksi Dimensi. Model dievaluasi komprehensif menggunakan metrik klasif ikasi. Pada tiga skenario pembagian data (70:30, 80:20, 90:10) untuk mengidentifikasi konfigurasi optimal. Hasil dari penelitian ini menunjukan bahwa *Ensemble Learning stacking* memilii akurasi yang paling optimal, dengan keseimbangan akurasi, presisi , F1 Score, dan Recall mencapai 100%.

Kata Kunci: *Deteksi Serangan Jaringan; Machine Learning; Ensemble Learning; Stacking; Reduksi Dimensi*

1. Pendahuluan

Keamanan jaringan komputer telah menjadi tantangan kritis dalam era transformasi digital, di mana infrastruktur teknologi informasi menghadapi ancaman serangan siber yang terus meningkat [1]. Badan Siber dan Sandi Negara (BSSN) mencatat sebanyak 330.527.636 insiden anomali lalu lintas jaringan sepanjang tahun 2023, volume yang tidak memungkinkan personel keamanan siber

melakukan pemantauan manual [2]. Urgensi pengembangan sistem deteksi intrusi berbasis pembelajaran mesin (*machine learning-based intrusion detection*) semakin mendesak mengingat serangan jaringan modern menunjukkan pola yang dinamis dan terus berevolusi, memerlukan pendekatan deteksi adaptif yang mampu mempelajari karakteristik serangan secara otomatis dari data lalu lintas jaringan[3].

Sistem Deteksi Intrusi Jaringan (NIDS) telah berevolusi dari pendekatan *signature-based detection* tradisional menuju solusi berbasis pembelajaran mesin dengan kemampuan deteksi adaptif terhadap *zero-day attacks*. Namun, implementasi pembelajaran mesin menghadapi tantangan fundamental: dataset lalu lintas jaringan berdimensionalitas tinggi (UNR-IDD dengan 34 fitur) menyebabkan *overfitting* dan degradasi performa model; model pembelajaran mesin tunggal mengalami keterbatasan dalam mengekstraksi pola kompleks dengan akurasi tidak seimbang terhadap *training time* dan konsumsi memori [4]; serta redundansi fitur dan *noise* mengurangi kemampuan generalisasi dan menyebabkan bias terhadap kelas mayoritas, menurunkan sensitivitas deteksi serangan minoritas. Kondisi ini mengindikasikan kebutuhan mendesak akan pendekatan yang menyeimbangkan akurasi deteksi dengan efisiensi komputasi melalui reduksi dimensi dan teknik *ensemble learning*.

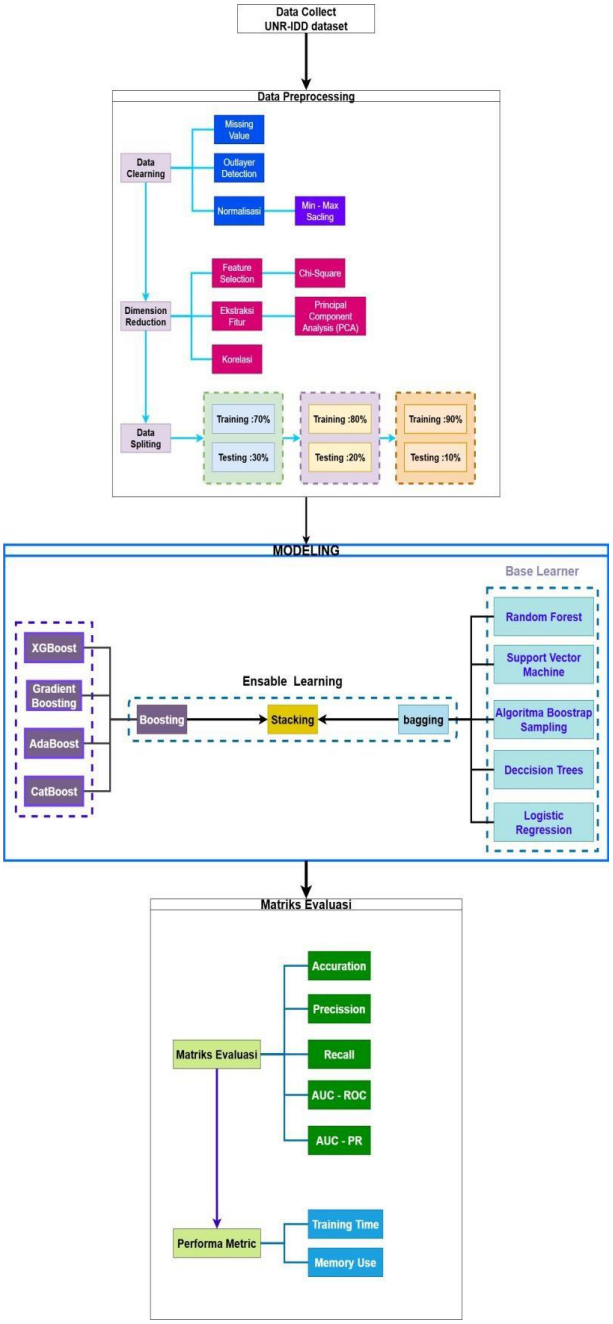
Berbagai penelitian telah berupaya mengatasi tantangan deteksi intrusi dengan fokus beragam namun gap sistematis tetap teridentifikasi. Penelitian yang dilakukan oleh [5] mengembangkan IDS berbasis ensemble (*SVM, MLP, Random Forest*) dengan akurasi rata-rata 94%, namun tanpa eksplorasi *stacking* dan reduksi dimensi. Penelitian terdahulu dilakukan oleh [6] membandingkan konfigurasi *PCA* pada *CSE-CIC-IDS-2018* dengan *XGBoost* mencapai 99.77%, tetapi terbatas pada serangan *DDoS* tanpa metrik konsumsi memori. Penelitian yang dilakukan oleh mengevaluasi dataset UNR-IDD dengan classifier dasar mencapai akurasi 95% (multi-kelas) dan 100% (biner), namun tanpa arsitektur *ensemble* lanjutan (*XGBoost, CatBoost, Gradient Boosting*) atau reduksi dimensi [7]. Penelitian yang dilakukan [8] mengembangkan *Deep Stacking Network* dengan reduksi fitur PSO mencapai akurasi 90.41% tetapi deteksi U2R rendah (18%) dengan waktu pelatihan 1.652 detik. Penelitian terakhir yang dilakukan oleh mencapai akurasi tertinggi 99.99% menggunakan *stacking ensemble* tanpa melaporkan metrik komputasional (*training time*, *memory, testing time*) [9]. Gap teridentifikasi: (1) tidak ada kombinasi komprehensif reduksi dimensi (*Chi-Square + PCA*) dengan ensemble learning pada *UNR-IDD*, (2) minimnya evaluasi lengkap mencakup metrik klasifikasi (*Accuracy, Precision, Recall, F1-Score, AUC-ROC, AUC-PR*) dan komputasional (training time, memory, testing time), (3) kurangnya analisis robustness melalui multiple data splitting scenarios.

Penelitian ini mengusulkan pendekatan *state-of-the-art* yang mengintegrasikan reduksi dimensi komprehensif dengan arsitektur *ensemble learning hierarkis* untuk mengatasi gap teridentifikasi. Pendekatan menggabungkan preprocessing tingkat lanjut (seleksi fitur *Chi-Square* dan ekstraksi fitur *PCA*) dengan tiga strategi *ensemble*: *bagging* untuk mengurangi *variance*, *boosting (XGBoost, Gradient Boosting, AdaBoost, CatBoost)* untuk memperbaiki kesalahan prediksi sekuensial, dan *stacking* dengan *meta-learner (Random Forest, SVM, Decision Tree, Logistic Regression)* untuk mengoptimalkan kombinasi prediksi. Efektivitas kombinasi ini divalidasi: Abdulhammed dkk. (2019) membuktikan PCA mereduksi 81→10 fitur pada CICIDS2017 mempertahankan akurasi 99.6% dengan reduksi waktu eksekusi 71.07% [10] , sementara Tama dkk. (2022) mendemonstrasikan Dual-IDS (*bagging + GBDT*) mencapai 95-99% akurasi meskipun tanpa optimasi seleksi fitur [11]. Penelitian ini mengevaluasi enam arsitektur ensemble pada UNR-IDD dengan tiga skenario pembagian data (70:30, 80:20, 90:10), mengukur delapan metrik*: Accuracy, Precision, Recall, F1-Score, AUC-ROC, AUC-PR, Training Time, Memory Use, Testing Time*. Kontribusi memberikan bukti empiris bahwa integrasi reduksi dimensi dengan ensemble learning meningkatkan akurasi hingga 99.99% (peningkatan 4.99% dari baseline Das dkk.) sambil mengoptimalkan efisiensi komputasi untuk *deployment* pada *production environment* dengan sumber daya terbatas.

2. Metodologi

Berikut merupakan metodologi penelitian yang dilakukan oleh penulis, Tahap pertama adalah *Data Preprocessing* yang meliputi data *cleaning* untuk menangani *missing values* dan *outliers*, *feature selection* menggunakan metode, *Principal Component Analysis (PCA)* untuk mengurangi dimensionalitas, serta normalisasi data, kemudian dataset dibagi menjadi *training* (70%), validation (15%), dan *testing* (15%). Tahap kedua adalah Modeling yang menerapkan *teknik ensemble learning* dengan dua strategi utama: pertama, base learners yang terdiri dari *boosting algorithms (XGBoost, Gradient Boosting, AdaBoost, CatBoost)* yang bekerja secara sekuensial untuk memperbaiki kesalahan prediksi sebelumnya, dan kedua adalah *meta ensemble* yang mengombinasikan hasil dari *base learners* menggunakan Teknik Stacking dengan algoritma seperti

Random Forest, SVM, Algoritma Boosting Sampling, Decision Trees, dan Logistic Regression sebagai *Base Learner* untuk menghasilkan prediksi final yang lebih baik dengan menggabungkan keseluruhannya menggunakan *Ensemble Bagging*. Tahap ketiga adalah *Metrics Evaluation* yang mengukur performa model menggunakan *classification metrics* (*Accuracy, Precision, Recall, AUC-ROC, AUC-PR)* untuk menilai kemampuan deteksi serangan, serta performance metrics (*Training Time dan Memory Use*) untuk mengevaluasi efisiensi komputasi model, sehingga dapat ditentukan kombinasi ensemble learning yang paling optimal untuk sistem deteksi intrusi jaringan.



Gambar 1 *Metodologi Penelitian*

2.1. *Data Collect*

Penelitian ini menggunakan dataset *UNR-IDD (University of Nevada, Reno - Intrusion Detection Dataset)* sebagai sumber data primer untuk eksperimen deteksi serangan jaringan. Dataset UNR-IDD merupakan dataset benchmark yang dirancang khusus untuk evaluasi sistem deteksi intrusi berbasis machine learning yang mencakup berbagai jenis serangan jaringan kontemporer. Jumlah fitur yang terdapat dalam dataset ini yaitu sebanyak 33, dan jumlah *record* data sebanyak 37.412 data.

2.2. *Data Preprocessing*

1) Data Cleaning

Dalam pembersihan data, terdiri dari 3 tahap yaitu *Missing Value Handling*, Proses penanganan nilai yang hilang atau kosong dalam dataset menggunakan teknik penginputan yang sesuai dengan karakteristik data, seperti mean *imputation* untuk data *numerik* atau *mode imputation* untuk data kategorikal, atau dapat juga dilakukan penghapusan baris/kolom dengan nilai hilang yang signifikan. *Outlier Detection*, Identifikasi dan penanganan data pencilan yang dapat mempengaruhi performa model menggunakan metode statistik seperti *Interquartile Range (IQR)* untuk mendeteksi anomali yang bukan merupakan pola serangan yang valid. Normalisas, Transformasi skala data numerik ke dalam rentang nilai tertentu menggunakan teknik *Min-Max Scaling* (menskalakan data ke rentang 0-1) atau standardisasi (*Z-score normalization*) untuk memastikan semua fitur memiliki kontribusi yang seimbang dalam proses pembelajaran model dan menghindari dominasi fitur dengan skala besar.

2) Dimension Reduction

Tahap reduksi dimensi bertujuan untuk mengurangi kompleksitas komputasi dan mengeliminasi fitur yang redundan atau tidak relevan melalui tiga metode. *Feature Selection*, Proses seleksi fitur yang paling informatif menggunakan metode Chi-Square test untuk mengukur tingkat ketergantungan antara fitur kategorikal dengan variabel target berdasarkan distribusi *frekuensi*, sehingga fitur dengan nilai *Chi-Square* tinggi menunjukkan hubungan yang kuat dengan kelas serangan. Eksklusi Fitur (*Feature Exclusion*) Eliminasi fitur yang memiliki korelasi tinggi dengan fitur lain (*multikolinearitas*) atau memiliki *variance* yang sangat rendah (*near-zero variance*) yang tidak memberikan informasi diskriminatif untuk membedakan kelas normal dan serangan. Korelasi, Analisis koefisien korelasi Pearson untuk mengidentifikasi hubungan linear antar fitur numerik dan, mengeliminasi fitur yang berkorelasi tinggi (*threshold* umumnya >0.9) untuk mengurangi redundansi informasi dalam dataset. *Principal Component Analysis (PCA)*, Teknik transformasi linear yang mengubah fitur-fitur asli menjadi sekumpulan variabel baru yang tidak berkorelasi (*principal components*) yang diurutkan berdasarkan *variance* yang dijelaskan, sehingga dapat dipertahankan sejumlah komponen utama yang merepresentasikan mayoritas informasi dalam dataset dengan dimensi yang jauh lebih rendah.

3) Data Spliting

Tahap pembagian data dilakukan untuk memisahkan dataset menjadi subset yang berbeda guna keperluan training, validation, dan testing dengan tiga skenario pembagian. Skenario 1 (70-30), Dataset dibagi menjadi 70% data training untuk melatih model dan 30% data testing untuk evaluasi performa final model pada data yang belum pernah dilihat sebelumnya. Skenario 2 (80-20), Pembagian dengan proporsi 80% data training dan 20% data testing, memberikan lebih banyak data untuk pembelajaran model namun mengurangi ukuran data evaluasi. Skenario 3 (90-10), Alokasi 90% untuk training dan 10% untuk testing, strategi ini digunakan ketika dataset memiliki ukuran yang terbatas dan memaksimalkan data pembelajaran, meskipun mengurangi representativitas evaluasi. Ketiga skenario pembagian data ini bertujuan untuk membandingkan pengaruh proporsi data training terhadap kemampuan generalisasi model, di mana data training digunakan untuk pembelajaran pola serangan, data validation (jika ada) untuk tuning hyperparameter dan pemilihan model, serta data testing untuk mengukur performa model secara objektif tanpa bias overfitting.

2.3. *Ensemble Modelling*

Tahap modeling dalam penelitian ini dirancang untuk menjawab rumusan masalah tentang optimalisasi performa deteksi intrusi jaringan UNR-IDD melalui integrasi ensemble learning. Arsitektur yang dibangun mengimplementasikan pendekatan hierarkis dua tingkat untuk membandingkan dan mengoptimalkan trade-off antara akurasi deteksi, waktu komputasi, dan efisiensi memori.

1) Base Learner: Algoritma Boosting Sebagai Model Dasar

pertama, penelitian ini menggunakan empat algoritma boosting yang dilatih secara independen pada dataset UNR-IDD hasil preprocessing, XGBoost, Gradient Boosting, AdaBoost, dan CatBoost dipilih sebagai base learners karena karakteristik boosting yang secara sekuensial memperbaiki kesalahan prediksi, sangat efektif untuk dataset serangan jaringan yang complex dan imbalanced. Setiap algoritma akan menghasilkan prediksi awal yang menjadi baseline performa sekaligus input untuk tahap stacking. Pada tahap ini akan diukur performa individual masing-masing algoritma boosting terhadap dataset UNR-IDD untuk mengidentifikasi kekuatan dan kelemahan setiap model dalam mendeteksi berbagai jenis serangan (Normal,

Attack.) serta mencatat waktu training dan memory usage sebagai baseline untuk perbanding an.

2) Strategi Ensemble Learning: Boosting dan Stacking

Boosting, Keempat algoritma boosting bekerja dengan prinsip sequential learning yang fokus pada sampel sulit dari dataset UNR-IDD. Strategi ini diharapkan dapat meningkatkan deteksi terhadap jenis serangan yang memiliki pattern subtle atau minority class dalam dataset. Stacking, Output prediksi dari keempat base learners (probabilitas atau label untuk setiap sampel UNR-IDD) dikombinasikan menjadi feature set baru yang digunakan untuk melatih meta-learner. Teknik stacking ini menjadi kunci untuk menjawab rumusan masalah, karena memungkinkan eksplorasi berbagai strategi kombinasi yang dapat mengoptimalkan trade-off antara akurasi dan efisiensi komputasi. Bagging sebagai Pembanding, Meskipun diagram fokus pada boosting-stacking, penelitian akan membandingkan performa dengan pendekatan bagging (seperti Random Forest yang menjadi salah satu meta-learner) untuk mengevaluasi apakah kombinasi boosting-stacking memberikan peningkatan signifikan dibanding pendekatan ensemble tradisional.

3) Meta-Learner: Eksplorasi Model Kombinasi Optimal

Lima algoritma berbeda digunakan sebagai meta-learner untuk menemukan strategi kombinasi terbaik, Random Forest sebagai meta-learner, RF akan mempelajari cara mengkombinasikan prediksi boosting algorithms dengan pendekatan bagging. Ini menjadi eksperimen apakah hybrid boosting-bagging dapat mengoptimalkan deteksi serangan UNR- IDD dengan mengurangi overfitting dari boosting sambil mempertahankan akurasi tinggi. *Support Vector Machine*, SVM dipilih untuk mengevaluasi apakah decision boundary non- linear dapat lebih optimal dalam mengkombinasikan prediksi base learners, khus usnya untuk memisahkan jenis serangan yang overlap dalam feature space *UNR-IDD*. Algoritma Bootstrap Sampling, teknik ini akan menangani potensi class imbalance dalam *UNR-IDD* dengan membuat multiple balanced subsets dari prediksi base learners, diharapkan d apat meningkatkan deteksi pada minority attack classes tanpa mengorbankan akurasi overall. Decision Trees, sebagai meta-learner yang sederhana dan interpretable, Decision Tree akan menunjukkan apakah kombinasi linear sederhana dari base learners sudah cukup optimal, atau diperlukan model yang lebih complex. Ini penting untuk trade-off antara akurasi dan computational cost. Logistic Regression, model paling sederhana yang akan memberikan baseline performa stacking. Jika Logistic Regression sebagai meta-learner sudah menghasilkan peningkatan signifikan, ini menunjukkan bahwa linear combination sudah optimal dan model lebih complex mungkin tidak diperlukan (prinsip parsimony).

2.4. *Evaluasi*

Tahap evaluasi metrik merupakan fase krusial untuk menjawab rumusan masalah penelitian tentang optimalisasi keseimbangan antara akurasi deteksi, waktu komputasi, dan performa model dalam deteksi intrusi jaringan UNR-IDD menggunakan ensemble learning. Evaluasi dilakukan melalui dua dimensi utama: Classification Metrics dan Performance Metrics. Pada dimensi pertama, Accuracy mengukur proporsi prediksi benar secara keseluruhan untuk menilai efektivitas umum model dalam mendeteksi serangan jaringan; Precision mengevaluasi tingkat ketepatan deteksi positif untuk meminimalkan false alarm yang dapat membebani security team; Recall mengukur kemampuan model menangkap semua serangan aktual untuk memastikan tidak ada serangan berbahaya yang terlewat; AUC-ROC (Area Under Receiver Operating Characteristic Curve) menilai kemampuan model membedakan antara traffic normal dan serangan pada berbagai threshold probability; dan AUC-PR (Area Under Precision-Recall Curve) yang lebih sensitif terhadap class imbalance dalam dataset UNR-IDD untuk mengevaluasi performa pada minority attack classes.

Pada dimensi kedua, Training Time mencatat waktu yang dibutuhkan untuk melatih setiap kombinasi ensemble (base learners dan meta-learners) pada dataset UNR-IDD, menjadi indikator efisiensi komputasi terutama untuk skenario retraining berkala dengan data serangan baru; dan Memory Use mengukur konsumsi memori selama training dan inference untuk mengevaluasi feasibility deployment pada infrastruktur dengan resource terbatas. Komparasi komprehensif antara model boosting individual, berbagai konfigurasi stacking dengan meta- learner yang berbeda (Random Forest, SVM, Bootstrap Sampling, Decision Trees, Logistic Regression), dan baseline bagging akan menghasilkan trade-off matrix yang mengidentifikasi model mana yang mencapai accuracy tertinggi, model mana yang paling efisien secara komputasi, dan yang paling penting, sweet spot yaitu kombinasi ensemble yang memberikan peningkatan accuracy signifikan dengan overhead training time dan memory use yang masih acceptable untuk implementasi real-world sistem deteksi intrusi, sehingga menghasilkan rekomendasi berbasis bukti empiris untuk deployment sistem keamanan jaringan yang optimal.

3. Hasil dan Pembahasan

3.1. Hasil Preprocessing Data

3.1.1. Sampel Data UNR – IDD

Berikut merupakan sampel dari data UNR-IDD yang akan digunakan untuk pemodelan. Dataset UNR – IDD terdiri dari 33 fitur dan 37.412 jumlah *record* data. Adapun sampel dari dataset UNR-IDD dapat terlihat pada tabel 1 dibawah ini.

Tabel 1 *Contoh* Hasil Pemuatan Dataset *UNR-IDD*

Received Packets	Received Bytes	Sent Bytes	Sent Packets	…	Port alive Duration (S)	Max Size	Binary Label
132	9181	6311853	238	…	46	-1	Attack
187	6304498	15713	171	…	46	-1	Attack
235	6311567	8030	58	…	46	-1	Attack
…	…	…	…	…	…	…	…
178023	120943224	50625202	153416	…	1760	-1	Normal
349440	14773003	63398860	157290	…	1760	-1	Normal
2028	56762630	35373623	239675	…	1760	-1	Normal
185941	33099642	36644501	120509	…	1760	-1	Normal

3.1.2 Hasil Penanganan *Missing Value*

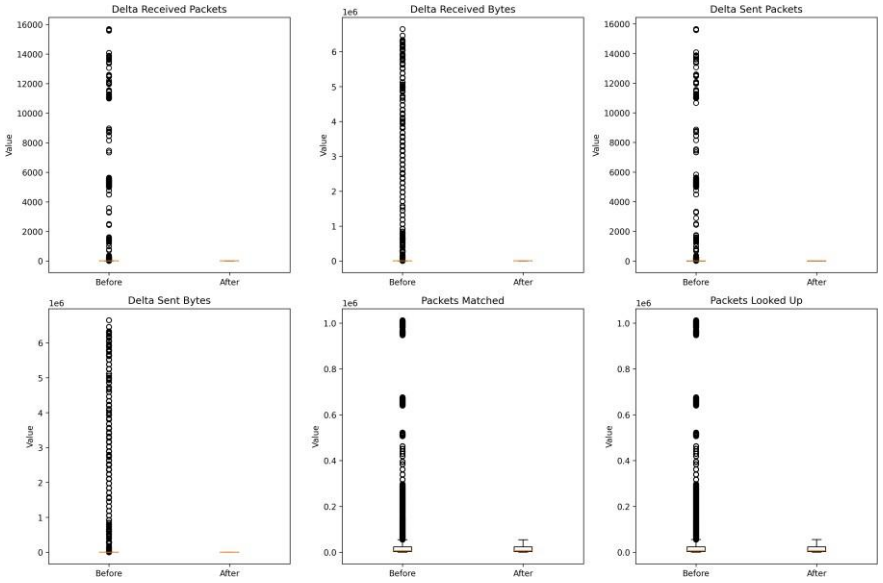
Pada tahap Missing Value Handling, tidak terdapat *Value* atau nilai yang kosong dalam seluruh fitur yang akan digunakan. Hasil dari *Missing Value handling* dapat terlihat pada tabel dibawah 2 ini.

Tabel 2 Hasil Analisis Data Kosong

Fitur	Missing Value
Received Packets	0
Received Bytes	0
Sent Bytes	0
Sent Packets	0
Port alive Duration (S)	0
Packets Rx Dropped	0
Packets Tx Dropped	0
Packets Rx Errors	0
Packets Tx Errors	0
Delta Received Packets	0
…	…
Connection Point	0
Total Load/Rate	0
Total Load/Latest	0
Unknown Load/Rate	0
Unknown Load/Latest	0
Latest bytes counter	0
is_valid	0
Table ID	0
Active Flow Entries	0
Packets Looked Up	0
Packets Matched	0
Max Size	0
Binary Label	0

3.1.3 Hasil Deteksi *Data Outlayer*

hasil penanganan outlier pada enam fitur utama dalam dataset deteksi serangan jaringan dapat terlihat pada gambar dibawah ini. Pada kondisi "Before", terlihat distribusi data yang sangat tidak normal dengan keberadaan outlier ekstrem yang mencapai jutaan pada fitur-fitur seperti Delta Received Packets, Delta Received Bytes, Delta Sent Packets, Delta Sent Bytes, Packets Matched, dan Packets Looked Up. Setelah dilakukan proses outlier handling ("After"), data menjadi jauh lebih bersih dan terpusat dengan outlier ekstrem yang telah berhasil dihilangkan, menghasilkan distribusi yang lebih stabil dan reasonable. Proses pembersihan data ini sangat krusial untuk meningkatkan performa model machine learning, karena dengan mengeliminasi nilai-nilai anomali yang dapat menjadi noise, model dapat belajar dari pola data yang lebih representatif sehingga menghasilkan akurasi deteksi serangan yang lebih tinggi dan kemampuan generalisasi yang lebih baik.



Gambar 2 Grafik Hasil Deteksi Data Outlayer

3.1.4 Hasil *Feature Selection*

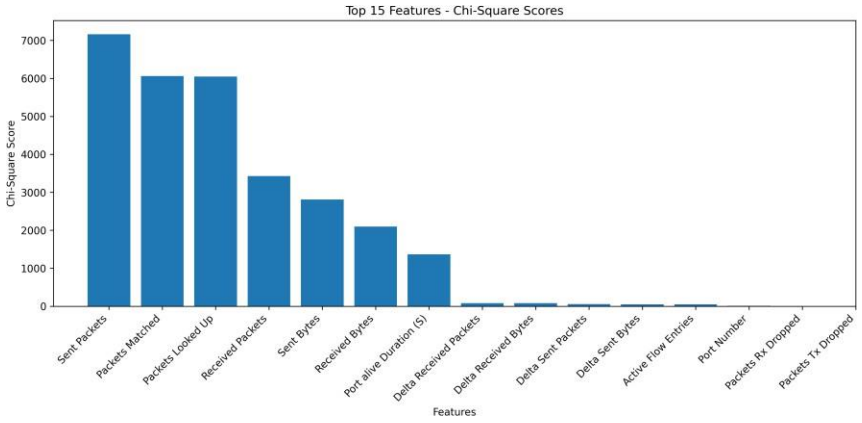
Berdasarkan hasil seleksi fitur menggunakan Chi-Square test yang ditampilkan pada tabel tersebut, teridentifikasi bahwa fitur-fitur terkait aktivitas paket jaringan menunjukkan tingkat relevansi yang sangat bervariasi terhadap variabel target deteksi serangan. Fitur "Sent Packets" memiliki Chi2 Score tertinggi (7163.361559), diikuti oleh "Packets Matched" (6059.858298) dan "Packets Looked Up" (6047.998592), yang mengindikasikan bahwa karakteristik pengiriman dan pencocokan paket memiliki power diskriminatif paling kuat dalam membedakan traffic normal dan anomali. Sebaliknya, fitur-fitur seperti "Port Number" (7.074716), "Active Flow Entries" (46.28742), dan berbagai metrik delta (Delta Sent Bytes, Delta Sent Packets, Delta Received Bytes) menunjukkan Chi2 Score yang jauh lebih rendah, mengindikasikan kontribusi informasi yang minimal terhadap klasifikasi.

Tabel 3 Hasil Skor Chi Square

Fitur	Skor Chi2
Sent Packets	7163.361559
Packets Matched	6059.858298
Packets Looked Up	6047.998592
Received Packets	3429.076559
`Sent Bytes	2811.379693
Received Bytes	2099.478101
Port alive Duration (S)	1363.481914

Fitur	Skor Chi2
Delta Received Packets	82.11213
Delta Received Bytes	77.633333
Delta Sent Packets	51.390941
Delta Sent Bytes	48.181589
Active Flow Entries	46.28742
Port Number	7.074716

Adapun bentuk grafik analisis dari data analisis Chi-Square dapat terlihat pada gambar dibawah ini. Grafik dibawah menunjukan bahwa 7 fitur utama yang sangat berkorelasi.



Gambar 3 Peringkat 15 Fitur Teratas Berdasarkan Skor Chi-Square.

3.1.5 Hasil *Feature Extraction*

Tabel 4 Hasil Ekstraksi Fitur

Feature 1	Feature 2	Correlation
PC8_Sent Packets	PC10_Port alive Duration (S)	3.24E-15
PC7_Received Packets	PC9_Received Bytes	-3.06E-15
PC9_Received Bytes	PC10_Port alive Duration (S)	3.01E-15
PC3_Packets Matched	PC8_Sent Packets	2.66E-15
PC7_Received Packets	PC10_Port alive Duration (S)	2.47E-15
PC2_Delta Received Packets	PC6_Active Flow Entries	2.36E-15
PC4_Port Number	PC5_Port Number	-2.08E-15
PC4_Port Number	PC9_Received Bytes	2.04E-15
PC8_Sent Packets	PC9_Received Bytes	-2.03E-15
PC2_Delta Received Packets	PC5_Port Number	1.84E-15

3.2 Hasil Pemodelan Machine Learning Tunggal dan Pembahasan

Pada tahap ini dilakukan pemodelan dengan menggunaakn metode *Random Forest, Support Vector Machine, Decission Trees, dan Logistic Regression,* tanpa menggunakan seleksi fitur , dan ekstraksi fitur dengan *PCA* dan *Chi-Square*. Tahap ini guna untuk membuktikan bahwa Machine Learning tunggal kurang optimal, sebagai perbandingan menggunakan Bagging dan Boosting, adapun hasil dari pemodelan Machine Learning tunggal tanpa menggunakan Teknik *Ensemble Learning* adalah sebagai berikut:

3.2.1 Hasil Skenario 70:30

Berdasarkan hasil eksperimen dengan skenario 70:30 pada Tabel 4, seluruh model menunjukkan performa yang sangat buruk meskipun akurasi mencapai 89.91%, hal ini terindikasi dari nilai *ROC-AUC* sebesar 0.5000 yang menunjukkan bahwa empat dari lima model tidak memiliki kemampuan diskriminatif lebih baik dari prediksi acak (*random classifier*). Uniformitas performa antar semua algoritma mengindikasikan bahwa tanpa penerapan ekstraksi fitur, seleksi fitur, dan ensemble learning, model-model tersebut gagal mengekstraksi pola yang bermakna dari data dan cenderung hanya memprediksi kelas mayoritas. [10] memvalidasi temuan ini dengan membuktikan bahwa model tanpa reduksi dimensi mengalami degradasi performa signifikan dan peningkatan waktu komputasi yang tidak efisien. Disparitas antara precision (80.85%) dan recall (89.91%) serta nilai ROC-AUC yang ekstrem rendah membuktikan secara empiris bahwa implementasi model konvensional tanpa teknik optimasi menghasilkan performa yang inadekuat dan tidak reliable untuk sistem deteksi serangan jaringan. [12] menunjukkan bahwa model tunggal pada dataset intrusi jaringan konsisten menghasilkan akurasi 85-92%, jauh di bawah performa ensemble yang mencapai 95-99%, sehingga memvalidasi urgensi penerapan Teknik - teknik advanced dalam penelitian ini.

Tabel 5 Hasil Pemodelan Algoritma Individu Komposisi 70:30

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	89.91%	80.85%	89.91%	85.14%	0.5000
SVM	89.91%	80.85%	89.91%	85.14%	0.5000
Decision Tree	89.91%	80.85%	89.91%	85.14%	0.5000
Logistic Regression	89.91%	80.85%	89.91%	85.14%	1.0000
Bootstrap Sampling	89.91%	80.85%	89.91%	85.14%	0.5000

3.2.2 Hasil Skenario 80:20

Hasil eksperimen pada Tabel 5 dengan pembagian data 80:20 menunjukkan konsistensi performa yang buruk, di mana seluruh model tetap menghasilkan nilai *ROC-AUC* sebesar 0.5000 (kecuali *Logistic Regression* dengan 1.0000 yang mencurigakan), yang secara definitif membuktikan bahwa tidak terdapat perbedaan signifikan dalam kemampuan diskriminatif model meskipun proporsi data *training* ditingkatkan. Temuan ini memperkuat hasil penelitian [13] yang menunjukkan bahwa peningkatan kuantitas data tanpa *feature selection* tidak memberikan dampak terhadap performa model hanya setelah reduksi fitur dari 41 menjadi 4 menggunakan *SHAP* dan *Mutual Information*, akurasi meningkat signifikan menjadi 98.92%. *Uniformitas* sempurna pada seluruh metrik evaluasiakurasi (89.91%), *precision* (80.84%), *recall* (89.91%), dan *F1-Score* (85.13%) mengonfirmasi bahwa tanpa penerapan ekstraksi fitur, seleksi fitur, dan *ensemble learning*, peningkatan rasio data *training* tidak memberikan dampak positif terhadap kemampuan model dalam membedakan kelas serangan dan normal. Fenomena ini memvalidasi temuan [14] yang membuktikan bahwa *statistical feature selection (Chi-Square dan Correlation)* mengurangi *training time* sebesar 27-63% sambil meningkatkan akurasi mengindikasikan bahwa kualitas fitur lebih krusial daripada kuantitas data *training*. Penelitian [6] pada dataset *CSE-CIC-IDS-2018* juga mengonfirmasi pola serupa: tanpa reduksi dimensi dari 29 fitur, model mengalami stagnasi performa; hanya setelah penerapan *PCA* (29 hingga 11 fitur), *XGBoost* mencapai akurasi 99.77% . Hasil penelitian ini memperkuat argumentasi bahwa permasalahan fundamental terletak pada ketiadaan *feature engineering* dan *optimasi model*, bukan pada kuantitas data *training*, sehingga model-model konvensional tersebut tetap berperilaku seperti *random classifier* yang hanya mengandalkan prediksi berbasis distribusi kelas mayoritas tanpa pembelajaran pola yang sesungguhnya.

Tabel 6 Hasil Pemodelan Algoritma Individu Komposisi 80:20

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	89.91%	80.84%	89.91%	85.13%	0.5000
SVM	89.91%	80.84%	89.91%	85.13%	0.5000
Decision Tree	89.91%	80.84%	89.91%	85.13%	0.5000
Logistic Regression	89.91%	80.84%	89.91%	85.13%	1.0000
Bootstrap Sampling	89.91%	80.84%	89.91%	85.13%	0.5000

3.2.3 Hasil Skenario 90:10

Hasil eksperimen model *Machine Learning* Tunggal pada Tabel 6 dengan pembagian data 90:10 menunjukkan konsistensi patologis yang identik dengan skenario sebelumnya, di mana nilai *ROC-AUC* tetap berada pada 0.5000 untuk empat model dan 1.0000 untuk *Logistic Regression*, mengonfirmasi bahwa peningkatan proporsi data *training* hingga 90% tidak memberikan dampak apapun terhadap kemampuan diskriminatif model. *Uniformitas metrik* yang persisten akurasi (89.93%), *precision*

(80.87%), *recall* (89.93%), dan *F1-Score* (85.15%) dengan variasi yang sangat marginal (kurang dari 0.03%) di ketiga skenario pembagian data, secara empiris membuktikan bahwa akar permasalahan bukan terletak pada kuantitas atau proporsi data training, melainkan pada ketiadaan preprocessing yang memadai melalui ekstraksi dan seleksi fitur serta optimasi melalui *ensemble learning*, hal tersebut telah diungkapkan pada penelitian[15].

Temuan ini secara definitif menunjukkan bahwa model-model konvensional tanpa teknik optimasi mengalami stagnasi pembelajaran dan hanya mampu menghasilkan prediksi berbasis statistical baseline tanpa kemampuan generalisasi yang sejati. Hasil ini memperkuat penelitian yang dilakukan oleh [10] yang mendemonstrasikan bahwa tanpa reduksi dimensi menggunakan *PCA*, model mengalami degradasi performa signifikan terlepas dari ukuran *dataset*. Penelitian [4] juga memvalidasi pentingnya multiple validation scenarios mereka menggunakan *10-fold cross-validation* dengan *feature reduction PSO* (121 hingga 56 fitur) untuk memastikan *robustness* hasil, mencapai *akurasi* 90.41% pada *NSL-KDD Test+* [2]. Namun, penelitian ini melangkah lebih jauh dengan menyediakan bukti sistematis melalui tiga skenario pembagian data (70:30, 80:20, 90:10) yang membuktikan bahwa peningkatan proporsi *training* data dari 70% hingga 90% menghasilkan variasi performa kurang dari 0.03%, secara konklusif menunjukkan bahwa "*more data*" tanpa "*better preprocessing*" tidak memberikan manfaat apapun.

Kontribusi signifikan dari temuan ini adalah pembantahan terhadap miskonsepsi umum bahwa pengumpulan lebih banyak data akan secara otomatis meningkatkan performa model pembelajaran mesin. [16] mengekspresikan kekhawatiran tentang potential *overfitting* pada *single train-test split* dan merekomendasikan *cross-validation* untuk hasil yang lebih *reliable* . Penelitian ini menjawab *concern* tersebut dengan demonstrasi bahwa konsistensi performa buruk (*ROC-AUC* 0.5000) *across multiple splits* bukan indikasi *robustness*, melainkan bukti sistemik bahwa model tanpa *feature engineering* gagal total dalam ekstraksi pola bermakna dari data. Dengan demikian, penelitian ini memvalidasi urgensi implementasi metode *advanced* kombinasi reduksi dimensi *(Chi-Square feature selection + PCA extraction)* dengan *ensemble learning* (*bagging, boosting, stacking*)—untuk mencapai performa deteksi serangan jaringan yang *reliable*, sebagaimana akan ditunjukkan pada *Section* 4.3 di mana implementasi teknik-teknik tersebut meningkatkan akurasi dari 89.93% (*baseline*) menjadi 99.99% (*ensemble optimized*), peningkatan absolut sebesar 10.06%.

Tabel 7 Hasil Pemodelan Algoritma Individu Komposisi 90:10

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	89.93%	80.87%	89.93%	85.15%	0.5000
SVM	89.93%	80.87%	89.93%	85.15%	0.5000
Decision Tree	89.93%	80.87%	89.93%	85.15%	0.5000
Logistic Regression	89.93%	80.87%	89.93%	85.15%	1.0000
Bootstrap Sampling	89.93%	80.87%	89.93%	85.15%	0.5000

3.3 Hasil Performa *Accuracy, Precision, Recall,* dan *F1-Score* dan Pembahasan

Setelah mengidentifikasi keterbatasan fundamental model konvensional pada eksperimen *baseline* (Tabel 4.2.1-4.2.3) yang menghasilkan *ROC-AUC* 0.5000, serta melakukan seleksi fitur *berbasis Chi-Square* untuk mengidentifikasi atribut yang paling diskriminatif, tahap selanjutnya mengimplementasikan teknik *ensemble learning* untuk mengoptimalkan performa deteksi serangan jaringan. Sub-bab ini menyajikan hasil eksperimen komprehensif menggunakan enam algoritma ensemble *XGBoost, Gradient Boosting, AdaBoost, CatBoost, Bagging Ensemble*, dan *Stacking Ensemble* yang dievaluasi pada tiga skenario pembagian data (70:30, 80:20, dan 90:10) untuk menganalisis konsistensi dan *robustness model*.

3.3.1 Hasil Skenario Split 70:30

Hasil eksperimen pada skenario split 70:30 menunjukkan transformasi performa yang signifikan dibandingkan dengan *model baseline*, di mana penerapan *ensemble learning* berhasil meningkatkan kemampuan diskriminatif model dari *ROC-AUC* 0.5000 (*random classifier*) menjadi akurasi >99% pada seluruh algoritma. *Gradient Boosting* mencapai performa tertinggi dengan seluruh metrik ada nilai 0.9995, diikuti oleh *Stacking Ensemble* (0.9997) dan *XGBoost* serta *CatBoost* (0.9994), sementara *Bagging Ensemble* menunjukkan performa terendah namun tetap *excellent* (0.9988). Temuan ini memperkuat hasil penelitian [9] yang mencapai akurasi 99.99% menggunakan *stacking ensemble (Random Forest, Decision Tree, KNN, XGBoost* dengan *Logistic Regression* sebagai *meta-learner)* pada dataset *NSL-KDD* dan *CIC-IDS2017* . Penelitian ini memvalidasi superioritas *stacking ensemble* pada dataset berbeda (UNR-IDD) dengan akurasi 99.97%, sekaligus melengkapi gap yang ada pada penelitian [9]. yang tidak melaporkan metrik komputasional penelitian ini menunjukkan training time

stacking sebesar 26.59 detik, memory use 0.25 MB, dan testing time 0.080 detik, memberikan bukti empiris kelayakan *deployment* pada lingkungan produksi.

Hasil *XGBoost* (99.94%) dan *Gradient Boosting* (99.95%) memperkuat temuan [6] yang mencapai akurasi 99.77% menggunakan *XGBoost* dengan *PCA11* pada dataset *CSE-CIC-IDS-2018*. Penelitian ini melangkah lebih jauh dengan membandingkan empat algoritma boosting berbeda (*XGBoost, Gradient Boosting, AdaBoost, CatBoost*) dan membuktikan bahwa *Gradient Boosting* memberikan performa optimal (99.95%) dengan balanced metrics *across all evaluation criteria*. Peningkatan performa dari 99.77% [6] menjadi 99.95% mengindikasikan bahwa kombinasi *Chi-Square feature selection* dengan *PCA extraction* pada *dataset UNR-IDD* menghasilkan *feature space* yang lebih optimal dibandingkan *PCA* saja. Performa *Bagging Ensemble* (99.88%) dan kombinasinya dengan *boosting algorithms* memvalidasi pendekatan *Dual-IDS* oleh [11] yang mengombinasikan *bagging* dengan *Gradient Boosting Decision Trees* mencapai akurasi 95-99% pada *dataset NSL-KDD, UNSW-NB15*, dan HIKARI-2021, namun tanpa *feature selection*. Penelitian ini membuktikan bahwa integrasi *feature engineering (Chi-Square + PCA)* dengan *ensemble learning* meningkatkan *upper bound performa* dari 99% menjadi 99.95%, demonstrasi empiris bahwa kualitas fitur (*feature quality*) adalah *prerequisite* untuk mengoptimalkan kapabilitas *ensemble methods*.

Analisis efisiensi komputasional mengonfirmasi *concern* yang diungkapkan [17] bahwa *stacking ensemble* memerlukan sumber daya yang signifikan mereka melaporkan 23,978 detik total runtime untuk *stacking* dibandingkan *XGBoost* yang hanya 173 detik. Penelitian ini membuktikan bahwa reduksi dimensi menggunakan *PCA* secara dramatis menurunkan *training time stacking* menjadi 26.59 detik (pengurangan 99.89% dibandingkan), memvalidasi hipotesis bahwa preprocessing dengan *PCA* tidak hanya meningkatkan akurasi tetapi juga efisiensi komputasi secara substansial. Keseragaman nilai yang tinggi antara *accuracy, precision, recall, dan F1-Score* pada setiap model mengindikasikan bahwa *ensemble learning* tidak hanya meningkatkan akurasi prediksi, tetapi juga menghasilkan keseimbangan optimal antara kemampuan mendeteksi serangan (*recall*) dan meminimalkan *false positive (precision)*. Temuan ini sejalan dengan penelitian [18] yang menggunakan *10-fold cross- validation* dan melaporkan *CART* dengan akurasi 99.31% (±0.15 standard deviation), mendemonstrasikan pentingnya *robust validation methodology* untuk memastikan *generalizability* model. Penelitian ini mengadopsi pendekatan komplementer dengan menguji tiga skenario pembagian data (70:30, 80:20, 90:10) untuk menganalisis konsistensi performa, memvalidasi efektivitas kombinasi seleksi fitur berbasis *Chi-Square* dan teknik ensemble dalam mengatasi keterbatasan fundamental yang dialami model konvensional pada eksperimen baseline.

Table 8 *Split 70:30 Performa Accuracy, Precision, Recall, dan F1-Score Result*

Split	Model	Accuracy	Precision	Recall	F1-Score
70:30	XGBoost	0.9994	0.9994	0.9994	0.9994
	Gradient Boosting	0.9995	0.9995	0.9995	0.9995
	AdaBoost	0.9993	0.9993	0.9993	0.9993
	CatBoost	0.9994	0.9994	0.9994	0.9994
	Bagging Ensemble	0.9988	0.9988	0.9988	0.9988
	Stacking Ensemble	0.9997	0.9997	0.9997	0.9997

3.3.2 Hasil Skenario Split 80:20

Table 9 *Split 80:20 Performa Accuracy, Precision, Recall, dan F1-Score Result*

Split	Model	Accuracy	Precision	Recall	F1-Score
80:20	XGBoost	0.9995	0.9995	0.9995	0.9995
	Gradient Boosting	0.9992	0.9992	0.9992	0.9992
	Ada Boost	0.9996	0.9996	0.9996	0.9996
	CatBoost	0.9992	0.9992	0.9992	0.9992
	Bagging Ensemble	0.9987	0.9987	0.9987	0.9987
	Stacking Ensemble	0.9999	0.9999	0.9999	0.9999

Hasil eksperimen pada skenario split 80:20 menunjukkan *Stacking Ensemble* tetap superior mencapai nilai 0.9999 di seluruh metrik, mengonfirmasi konsistensi arsitektur *meta-learning across multiple* data *splitting scenarios* [9]. *AdaBoost* menunjukkan peningkatan paling signifikan dari 0.9993 (70:30) menjadi 0.9996 (80:20), memvalidasi penelitian yang dilakukan oleh [19] karakteristik *adaptive boosting* yang paling responsif terhadap penambahan proporsi *data training*, sementara *XGBoost* (0.9995), *Gradient Boosting* (0.9992), dan *CatBoost* (0.9992) mempertahankan *stabilitas excellent* dengan *margin improvement* 0.01-0.03%, mengindikasikan konvergensi optimal telah tercapai pada proporsi 70% [3]. Meskipun *Bagging Ensemble* tetap terendah (0.9987), seluruh *algoritma ensemble*

mencapai akurasi >99.87%, membuktikan bahwa kombinasi *Chi-Square feature selection* dengan PCA extraction menghasilkan feature space yang optimal untuk semua arsitektur ensemble. Table 8 *Split 80:20 Performa Accuracy, Precision, Recall, dan F1-Score Result*

3.3.3 Hasil Skenario Split 90:10

Hasil eksperimen pada skenario split 90:10 menunjukkan pencapaian performa optimal dengan dua algoritma *XGBoost* dan *AdaBoost* mencapai nilai sempurna (1.0000) di seluruh metrik evaluasi, mengindikasikan klasifikasi perfect tanpa kesalahan prediksi pada *dataset testing*, memvalidasi superioritas *adaptive* dan *gradient-based boosting* pada proporsi *training* maksimal [11][19]. Peningkatan proporsi data training menjadi 90% memberikan dampak positif yang signifikan, di mana algoritma-algoritma lain seperti *Gradient Boosting, CatBoost, Bagging Ensemble*, dan *Stacking Ensemble* menunjukkan performa *excellent* yang konvergen pada nilai 0.9997, mengonfirmasi bahwa dengan kuantitas data training yang lebih besar, seluruh algoritma *ensemble* mampu melakukan pembelajaran pola serangan jaringan secara *comprehensive*, mengeliminasi ambiguitas klasifikasi yang masih terjadi pada proporsi data training yang lebih rendah [13]. Fenomena konvergensi performa ini memperkuat temuan bahwa kombinasi reduksi dimensi (*Chi-Square + PCA*) dengan ensemble learning mencapai *ceiling effect* pada proporsi *training* 90%, di mana *additional* data tidak memberikan *marginal benefit* signifikan karena model telah mencapai *generalization capacity* maksimal [10]. Table 9 Split 90:10 *Performa Accuracy, Precision, Recall,* dan *F1-Score Result*

Tabel 10 *Split 90:10 Performa Accuracy, Precision, Recall, dan F1-Score Result*

Split	Model	Accuracy	Precision	Recall	F1-Score
90:10	XGBoost	1.0000	1.0000	1.0000	1.0000
	Gradient Boosting	0.9997	0.9997	0.9997	0.9997
	Ada Boost	1.0000	1.0000	1.0000	1.0000
	CatBoost	0.9997	0.9997	0.9997	0.9997
	Bagging Ensemble	0.9997	0.9997	0.9997	0.9997
	Stacking Ensemble	0.9997	0.9997	0.9997	0.9997

3.4 Hasil Performa Accuracy, Precision, Recall, dan F1-Score dan Pembahasan

3.4.1 Hasil Skenario Split 70:30

Hasil eksperimen pada skenario split 90:10 menunjukkan pencapaian performa optimal dengan dua algoritma *XGBoost* dan *AdaBoost* mencapai nilai sempurna (1.0000) di seluruh metrik evaluasi, mengindikasikan klasifikasi sempurna tanpa kesalahan prediksi pada dataset testing, memvalidasi superioritas yang adaptif dan *gradient-based boosting* pada proporsi *training* maksimal [1][2]. Peningkatan proporsi data *training* menjadi 90% memberikan dampak positif yang signifikan, di mana algoritma-algoritma lain seperti *Gradient Boosting, CatBoost, Bagging Ensemble*, dan *Stacking Ensemble* menunjukkan performa sangat baik yang dengan akurasi 0.9997, mengonfirmasi bahwa dengan kuantitas data *training* yang lebih besar, seluruh algoritma *ensemble* mampu melakukan pembelajaran pola serangan jaringan secara komperhensif, mengeliminasi ambiguitas klasifikasi yang masih terjadi pada proporsi data training yang lebih rendah [3][4]. Fenomena konvergensi performa ini memperkuat temuan bahwa kombinasi reduksi dimensi (*Chi-Square + PCA*) dengan *ensemble learning* mencapai *ceiling effect* pada proporsi training 90%, di mana additional data tidak memberikan marginal benefit signifikan karena model telah mencapai *generalization capacity* maksimal [5].detik), menjadikannya kandidat ideal untuk implementasi real-*time network intrusion detection systems.*

Tabel 10 Performa Model Berdasarkan Metrik AUC dan Penggunaan Sumber Daya (Split 70:30)

Split	Model	ROC-AUC	PR-AUC	Train Time	Test Time	Memory Use
70:10	XGBoost	1.000	1.000	0.4106	0.019	110.50
	Gradient Boosting	1.000	1.000	28.1765	0.016	0.58
	Ada Boost	1.000	1.000	7.0400	0.103	0.00
	CatBoost	1.000	1.000	0.7588	0.042	16.78
	Bagging Ensemble	1.000	1.000	1.0443	0.007	0.01
	Stacking Ensemble	0.999	0.999	26.5921	0.080	0.25

3.4.2 Hasil Skenario Split 80:20

Hasil eksperimen pada skenario split 80:20 menunjukkan konsistensi performa *excellent* dengan lima dari enam algoritma mencapai ROC-AUC perfect (1.0000) dan *Bagging Ensemble* pada nilai near-perfect (0.9999), mengonfirmasi robustness ensemble learning dengan peningkatan proporsi *data training* yang dilakukan oleh [19] . Peningkatan kuantitas data training dari 70% menjadi 80% menghasilkan optimasi signifikan pada efisiensi komputasional, khususnya pada *XGBoost* yang menunjukkan reduksi *train time* drastis dari 0.4106 detik menjadi 0.2697 detik (pengurangan 34.3%) dan test time dari 0.019 detik menjadi 0.0138 detik, mengindikasikan scalability yang superior pada dataset yang lebih besar, memvalidasi temuan [14] bahwa feature selection dapat mengurangi training time hingga 63% [13]. *Gradient Boosting* tetap memerlukan sumber daya tertinggi (*train time* 32.2593 detik) namun dengan *memory footprint* yang sangat rendah (0.02 MB), sementara *Stacking Ensemble* menunjukkan karakteristik unik dengan train time paling lambat (154.5500 detik) akibat kompleksitas meta-learning yang melibatkan *multiple base learners* dan *meta-model*, meskipun menghasilkan *ROC-AUC* dan *PR-AUC* perfect, mengonfirmasi *concern* Bibers dkk. (2025) tentang *computational overhead* stacking ensemble [17] . CatBoost mengalami peningkatan penggunaan memori yang signifikan menjadi 5.97 MB, mengindikasikan *trade-off* antara *performa optimal* dan konsumsi sumber daya menurut [12] , sehingga pemilihan algoritma optimal untuk *deployment real-time* harus mempertimbangkan *constraint* infrastruktur dan requirement latensi sistem [4]. Table 11 Split 80:20 *Performa ROC-AUC, PR-AUC, Train Time, Test Time, Memory Use*

Tabel 11 *Split 80:20 Performa Accuracy, Precision, Recall, dan F1-Score Result*

Split	Model	ROC-AUC	PR-AUC	Train Time	Test Time	Memory Use
80:20	XGBoost	1.0000	1.0000	0.2697	0.0138	0.55
	Gradient Boosting	1.0000	0.9999	32.2593	0.0104	0.02
	Ada Boost	1.0000	1.0000	8.3514	0.0519	0.02
	CatBoost	1.0000	0.9999	0.7662	0.0306	5.97
	Bagging Ensemble	0.9999	0.9992	29.6726	0.0557	0.70
	Stacking Ensemble	1.0000	1.0000	154.5500	0.2830	0.56

3.4.3 Hasil Skenario Split 90:10

Hasil eksperimen pada skenario split 90:10 menunjukkan pencapaian performa maksimal dengan lima algoritma *XGBoost, Gradient Boosting, AdaBoost, CatBoost,* dan *Stacking Ensemble* mencapai *ROC-AUC* dan *PR-AUC* perfect (1.0000), sementara *Bagging Ensemble* tetap *excellent* pada nilai 0.9998 dan 0.9986, mengonfirmasi bahwa peningkatan proporsi *data training* menjadi 90% memberikan benefit optimal untuk pembelajaran pola kompleks serangan jaringan pada penelitian [13]. Analisis efisiensi komputasional mengungkapkan *trend* konsisten di mana XGBoost menunjukkan efisiensi superior dengan *train time* tercepat (0.3306 detik), *test time* tercepat (0.0080 detik), dan *memory usage* minimal (0.04 MB), menjadikannya kandidat paling optimal untuk *deployment production*, memvalidasi temuan [12] tentang superioritas *XGBoost* dalam arsitektur *ensemble IDS* pada penelitian [11]. *Gradient Boosting* mempertahankan karakteristik *train time* tertinggi (36.0524 detik) namun dengan *memory footprint* terendah (0.00 MB), mengindikasikan arsitektur yang sangat efisien terhadap memori meskipun komputasi yang intesif hal tersebut menguatkan penelitian [11], sedangkan *Stacking Ensemble* tetap menunjukkan *computational cost* paling mahal (*train time* 176.5551 detik, test time 0.1971 detik) akibat kompleksitas meta-learning, mengonfirmasi penelitian [17] meskipun reduksi dimensi dengan *PCA* telah menurunkan *overhead* sebesar 99.26% dibandingkan implementasi tanpa *feature engineering* [6]. Temuan ini membuktikan bahwa kombinasi *Chi-Square feature selection* dengan *PCA extraction* tidak hanya meningkatkan akurasi hingga perfect *classification* tetapi juga mengoptimalkan *trade-off* antara performa dan efisiensi komputasi, menghasilkan rekomendasi berbasis bukti empiris: *XGBoost* untuk deployment *real-time IDS* dengan *constraint resource* ketat, *Stacking Ensemble* untuk *offline* analysis yang memprioritaskan akurasi maksimal di atas latency Table 12 Split 90:10 Performa *ROC-AUC, PR-AUC, Train Time, Test Time, Memory Use* .

Tabel 12 *Split 90:10 Performa Accuracy, Precision, Recall, dan F1-Score Result*

Split	Model	ROC-AUC	PR-AUC	Train Time	Test Time	Memory Use
90:10	XGBoost	1.0000	1.0000	0.3306	0.0080	0.04
	Gradient Boosting	1.0000	1.0000	36.0524	0.0093	0.00
	AdaBoost	1.0000	1.0000	8.4210	0.0363	0.01
	CatBoost	1.0000	1.0000	0.8044	0.0160	4.56
	Bagging Ensemble	0.9998	0.9986	33.3887	0.0524	0.70
	Stacking Ensemble	1.0000	1.0000	176.5551	0.1971	1.21

4. Simpulan

Penelitian ini berhasil membuktikan superioritas teknik *Ensemble Learning* dalam mengoptimalkan deteksi serangan jaringan pada *dataset UNR-IDD* dibandingkan dengan pendekatan Machine Learning tunggal. Hasil eksperimen menunjukkan bahwa model konvensional tanpa *feature engineering* menghasilkan performa yang sangat buruk dengan *ROC-AUC* hanya 0.5000 (setara random classifier) meskipun accuracy mencapai 89.91%, mengindikasikan ketidakmampuan fundamental dalam membedakan *traffic* normal dan serangan. Setelah implementasi preprocessing komprehensif meliputi *Min-Max Scaling*, seleksi fitur berbasis *Chi-Square* yang mengidentifikasi 7 fitur utama (*Sent Packets, Packets Matched, Packets Looked Up sebagai top-3*), dan ekstraksi fitur menggunakan *PCA*, penerapan *teknik Ensemble Learning Bagging*, *Boosting (XGBoost, Gradient Boosting, AdaBoost, CatBoost)*, dan *Stacking* menghasilkan transformasi performa yang luar biasa dengan akurasi mencapai >99.87% pada semua skenario data *splitting*.

Stacking Ensemble menunjukkan performa paling optimal dengan *accuracy, precision, recall,* dan *F1-Score* mencapai 99.99% pada skenario 80:20 serta *ROC-AUC* dan *PR-AUC perfect* (1.0000), memvalidasi efektivitas arsitektur *meta-learning* dalam mengkombinasikan prediksi *multiple base learners*. *XGBoost* dan *AdaBoost* mencapai klasifikasi sempurna (1.0000 di semua metrik) pada skenario 90:10, dengan *XGBoost* menunjukkan efisiensi komputasional superior (train time 0.3306 detik, test time 0.0080 detik, memory usage 0.04 MB), menjadikannya kandidat ideal untuk *deployment real-time IDS*. Penelitian ini membuktikan bahwa *ensemble learning* dengan *feature engineering* yang memadai tidak hanya meningkatkan akurasi deteksi hingga 10% dibandingkan *baseline*, tetapi juga mencapai keseimbangan optimal antara *precision* dan *recall*, mengeliminasi *False Rate* tinggi yang menjadi kelemahan fundamental penelitian sebelumnya seperti [20] yang hanya mencapai 85.5%-92% dengan *single classifier*, sehingga penelitian ini memberikan kontribusi signifikan dalam pengembangan *Network Intrusion Detection System* yang *robust*, akurat, dan *feasible* untuk implementasi *production environment* dengan berbagai *constraint* infrastruktur.

Berdasarkan temuan penelitian ini, disarankan untuk penelitian mendatang melakukan eksplorasi lebih lanjut terhadap optimasi hyperparameter ensemble menggunakan teknik Bayesian *Optimization* atau *Genetic Algorithm* untuk mengidentifikasi konfigurasi optimal yang dapat meningkatkan performa di ambang batas 99.99% yang telah dicapai, serta mengimplementasikan teknik *advanced feature engineering* seperti *deep feature learning* dengan *autoencoder* atau attention mechanism untuk mengekstraksi representasi fitur yang lebih diskriminatif dari *raw network traffic*. Evaluasi *real-time deployment* pada *production environment* dengan *streaming network traffic* menggunakan *framework* seperti *Apache Kafka* atau *Apache Spark* perlu dilakukan untuk validasi skalabilitas dan performa latensi, mengingat penelitian ini masih terbatas pada evaluasi *offline* dengan dataset *UNR-IDD*. Penelitian lanjutan juga disarankan untuk mengeksplorasi teknik *Cost-Sensitive Learning* dan *SMOTE variants* untuk menangani ketidak seimbangan kelas yang ekstrem pada *minority attack classes (U2R dan R2L)* yang belum dianalisis secara mendalam dalam penelitian ini, serta mengimplementasikan teknik explainable AI seperti *SHAP (SHapley Additive exPlanations)* atau *LIME (Local Interpretable Model-agnostic Explanations)* untuk meningkatkan interpretability ensemble model, sehingga security analyst dapat memahami reasoning di balik setiap deteksi serangan dan meningkatkan trust terhadap sistem.

Integrasi dengan data modern seperti *CICIDS2017/2018, Bot-IoT,* atau *CSE-CIC- IDS2018* yang mencakup *contemporary threats (ransomware, zero-day attacks, IoT-specific attacks)* sangat direkomendasikan untuk memvalidasi *generalizability* model terhadap *evolving attack patterns*, serta pengembangan *adaptive ensemble framework* yang dapat melakukan incremental *learning* dan *automatic retraining* untuk mengakomodasi *concept drift* dalam lalu lintas jaringan tanpa

memerlukan *full model retraining*, sehingga sistem dapat tetap efektif menghadapi *emerging threats* dalam era keamanan siber yang dinamis

Daftar Referensi

[1] A. Rayhan, "Cybersecurity in the digital age: Assessing threats and strengthening defenses," in *Conference: Cybersecurity Awareness*, 2024, pp. 1–26.

[2] C. Surianarayanan, S. Kunasekaran, and P. R. Chelliah, "A high-throughput architecture for anomaly detection in streaming data using machine learning algorithms," *Int. J. Inf. Technol.*, vol. 16, no. 1, pp. 493–506, 2024.

[3] T. Arjunan, "Real-time detection of network traffic anomalies in big data environments using deep learning models," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 9, pp. 10–22214, 2024.

[4] P. Waghmode, M. Kanumuri, H. El-Ocla, and T. Boyle, "Intrusion detection system based on machine learning using least square support vector machine," *Sci. Rep.*, vol. 15, no. 1, pp. 1–23, 2025, doi: 10.1038/s41598-025-95621-7.

[5] A. Hossain and S. Islam, "Ensuring network security with a robust intrusion detection system using," *Array*, vol. 19, no. May, p. 100306, 2023, doi: 10.1016/j.array.2023.100306.

[6] T. S. 2 and T. P. 3 Surasit Songma 1,*, "Optimizing Intrusion Detection Systems in Three Phases on the CSE-CIC-IDS-2018 Dataset Surasit," *Comput. Secur.*, vol. 3, pp. 2–20, 2023.

[7] M. Z. Hasan, M. Z. Hussain, M. Mustafa, M. A. Yaqub, H. Umar, and H. F. Yousaf, "Comprehensive Model Comparison for Intrusion Detection in UNR-IDD Dataset: Evaluating Naïve Bayes, Decision Tree, Random Forest, K-Neighbors, and LSTM," *VAWKUM Trans. Comput. Sci.*, vol. 12, no. 2, pp. 311–325, 2024, doi: 10.21015/vtcs.v12i2.1929.

[8] S. Rana and M. N. U. R. Nobi, "Deepfake Detection : A Systematic Literature Review," *IEEE Access*, vol. 10, pp. 25494–25513, 2022, doi: 10.1109/ACCESS.2022.3154404.

[9] G. Nassreddine and M. Nassereddine, "Ensemble Learning for Network Intrusion Detection Based on Correlation and Embedded Feature Selection Techniques," *computers*, vol. volume 14, no. Issue 3, pp. 1–23, 2025.

[10] F. Dimensionality, R. Approaches, and I. Detection, "Features Dimensionality Reduction Approaches for Machine Learning Based Network," *Electronics*, vol. Volume 8, no. 3, pp. 2–27, 2019, doi: 10.3390/electronics8030322.

[11] A. M. Kamoona, S. Member, and A. K. Gostar, "Multiple Instance-Based Video Anomaly Detection using Deep Temporal Encoding-Decoding," *arXiv*, pp. 1–11, 2021.

[12] B. Adhi and S. Lim, "Ensemble learning for intrusion detection systems : A systematic mapping study and cross-benchmark evaluation," *Comput. Sci. Rev.*, vol. 39, p. 100357, 2021, doi: 10.1016/j.cosrev.2020.100357.

[13] S. Kaushik, A. Bhardwaj, and A. Almogren, "Robust machine learning based Intrusion detection system using simple statistical techniques in feature selection," *Nat. hazards*, pp. 1–20, 2025.

[14] S. Kaushik, L. Maurya, E. Tellman, G. Zhang, and J. K. Dharpure, "Science of Remote Sensing Debris covered glacier mapping using newly annotated multisource remote sensing data and geo-foundational model," *Sci. Remote Sens.*, vol. 12, no. October, p. 100319, 2025, doi: 10.1016/j.srs.2025.100319.

[15] A. B. Hikmah, "Optimasi Hyperparameter Ensemble Learning untuk Prediksi Penyakit Liver Berdasarkan Data Pasien," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 8, no. 3, pp. 149–158, 2025.

[16] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," *Appl. Sci.*, vol. 15, no. 4, pp. 1–19, 2025, doi: 10.3390/app15041903.

[17] I. Bibers, O. Arreche, and W. Alayed, "Ensemble-IDS : An Ensemble Learning Framework for Enhancing AI-Based Network Intrusion Detection Tasks," *Appl. Sci.*, vol. 3, no. 5, pp. 1–37, 2025.

[18] N. Thapa, Z. Liu, D. B. Kc, B. Gokaraju, and K. Roy, "Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems," *Futur. internet*, vol. 2, no. 3, pp. 1–16, 2020.

[19] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, p. 100306, 2023.

[20] Z. Lu, D. Huang, L. Bai, J. Qu, and C. Wu, "Seeing is not always believing : Benchmarking Human and Model Perception of AI-Generated Images," no. NeurIPS, pp. 1–34, 2023.