https://jurnal.pustakagalerimandiri.co.id/index.php/pustakaai
DOI : https://doi.org/10.55382/jurnalpustakaai.v6i1.1756

JURNAL PUSTAKA AI
JURNAL PUSAT AKSES KAJIAN TEKNOLOGI ARTIFICIAL INTELLIGENCE
Pustaka Galeri Mandiri

Vol. 6 No. 1 (2026) 134 – 143 E ISSN: 2809-4069

Studi Optimalisasi Deteksi Intrusi Jaringan NIDS Menggunakan XGBoost pada Dataset Netflow V2

Mhd Adi Setiawan Aritonang[1*], Muhammad Marshall Al Karim[2], Roland[3], Jefri Irwan Gultom[4], Samuel Enrico Sitompul[5], Hendri Wijaya[6]
[1,2,3,4,5,6]Program Studi Teknik Komputer, fakultas , Institut Teknologi Batam
[1]adilogika14@gmail.com .

Abstract

The increasing of cyber attacks complexity in modern networks, particularly in Internet of Things (IoT) environments poses the challenges in detecting attacks accurately, specifically on imbalanced data conditions and the presence of zero-day attacks. This research aims to develop a machine learning-based Network Intrusion Detection System (NIDS) by using the XGBoost Classifier algorithm to improve detection accuracy and reduce false positives. The methods employed include data preprocessing, stratified split, model training, and evaluation by using accuracy, precision, recall, and F1-score metrics. The results show an accuracy of 98.84% and a weighted F1-score of 98.80% with optimal performance on majority classes such as DDoS and DoS. However, a macro F1-score of 72.09% indicates limitations on minority classes. This research contributes to the development of an efficient and adaptive NIDS and provides a foundation for model improvement through handling class imbalance to make it more robust in real-world environments

Keywords: NIDS, Cybersecurity, Machine Learning, XGBoost, Networking

Abstrak

Peningkatan kompleksitas serangan siber pada jaringan modern, khususnya pada lingkungan Internet of Things (IoT), menimbulkan tantangan dalam mendeteksi serangan secara akurat, terutama pada kondisi data yang tidak seimbang (*imbalanced data*) dan keberadaan serangan baru (*zero-day*). Penelitian ini bertujuan mengembangkan sistem Network Intrusion Detection System (NIDS) berbasis *machine learning* menggunakan algoritma XGBoost Classifier untuk meningkatkan akurasi deteksi dan menekan false positive. Metode yang digunakan meliputi data *preprocessing*, *stratified split*, pelatihan model, dan evaluasi menggunakan metrik *accuracy, precision, recall*, dan *F1-score*. Hasil penelitian menunjukkan akurasi sebesar 98,84% dan *weighted F1-score* 98,80%, dengan kinerja optimal pada serangan mayoritas seperti DDoS dan DoS. Namun, macro *F1-score* sebesar 72,09% mengindikasikan keterbatasan pada kelas minoritas. Penelitian ini berkontribusi dalam pengembangan NIDS yang efisien dan adaptif, serta memberikan dasar untuk peningkatan model melalui penanganan *class imbalance* agar lebih *robust* di lingkungan nyata.

Kata kunci: *NIDS, Cybersecurity, Machine Learning, XGBoost, Networking*

Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026

1. Pendahuluan

Perkembangan teknologi jaringan komputer dan *Internet of Things* (IoT) telah meningkatkan ketergantungan berbagai sektor terhadap sistem jaringan untuk mendukung layanan yang bersifat kritis. Namun, peningkatan konektivitas ini juga diiringi dengan meningkatnya kompleksitas dan intensitas serangan siber, seperti *Denial of Service* (DoS), *Man-in-the-Middle* (MitM), *port scanning*, serta serangan *zero-day* yang semakin sulit dideteksi menggunakan pendekatan keamanan tradisional [1]. Sebagai contoh, beberapa studi terbaru menunjukkan bahwa serangan zero-day pada jaringan IoT meningkat hingga lebih dari 40% dalam dua tahun terakhir, yang semakin menegaskan ketidakmampuan sistem keamanan konvensional[16].

Intrusion Detection System (IDS) merupakan salah satu komponen utama dalam arsitektur keamanan jaringan modern yang berfungsi untuk memantau lalu lintas jaringan atau aktivitas *host* dan memberikan peringatan ketika terdeteksi aktivitas mencurigakan [1]. Secara umum, IDS diklasifikasikan menjadi *Network Intrusion Detection System* (NIDS) dan *Host Intrusion Detection System* (HIDS). NIDS memantau lalu lintas jaringan secara keseluruhan, sedangkan HIDS berfokus pada aktivitas individual dari suatu host [1]. Berdasarkan pendekatan deteksinya, IDS diklasifikasikan menjadi tiga jenis: *misuse detection*, *anomaly detection*, dan *hybrid detection*. Metode *misuse detection* mampu mencapai akurasi yang tinggi terhadap serangan yang sudah diketahui polanya, tetapi tidak efektif dalam menghadapi serangan baru (*zero-day*). Sebaliknya, *anomaly detection* lebih adaptif karena mampu mengenali pola serangan yang belum pernah terjadi sebelumnya, meskipun kelemahan utamanya adalah tingginya tingkat *false positive* [15]. Penelitian dalam sebelumnya menunjukkan bahwa tingkat *false positive* pada *anomaly detection* masih berada di kisaran 15–25% pada lingkungan jaringan yang dinamis, sehingga menjadi tantangan utama yang belum terpecahkan [14].

Seiring berkembangnya ancaman siber, IDS konvensional dinilai tidak lagi cukup efektif, sehingga mulai bergeser ke *Intrusion Prevention System* (IPS) yang memiliki kemampuan aktif untuk mencegah serangan secara langsung [17]. Namun demikian, IPS menghadapi tantangan besar, terutama terkait tingginya tingkat *False Positive* yang dapat menyebabkan pemblokiran lalu lintas yang sah dan berpotensi menimbulkan gangguan layanan jaringan. Tantangan ini semakin kompleks karena IPS sering ditempatkan secara *inline* dan berperan sebagai *single point of failure* dalam jaringan [18]. Beberapa penelitian terkini mengusulkan pendekatan hibrida antara IDS dan IPS dengan mekanisme *feedback loop* untuk mengurangi *false positive*, namun implementasinya masih terbatas pada skala laboratorium [19].

Untuk mengatasi keterbatasan tersebut, pendekatan berbasis *Artificial Intelligence* (AI) dan *Machine Learning* (ML) semakin banyak diadopsi dalam pengembangan IDS modern. *Machine learning* memungkinkan sistem untuk mempelajari pola lalu lintas jaringan secara otomatis dan mendeteksi baik serangan yang telah dikenal maupun serangan baru tanpa perlu aturan statis yang didefinisikan secara manual [2]. NIDS berbasis ML umumnya diklasifikasikan menjadi pendekatan berbasis *supervised learning*, *unsupervised learning*, dan *semi-supervised learning*. *Supervised learning* memanfaatkan data berlabel untuk membangun model klasifikasi atau regresi, seperti *Decision Tree*, *k-Nearest Neighbor* (kNN), *Support Vector Machine* (SVM*), Random Forest*, dan *Artificial Neural Network* (ANN) [2], [3]. Pendekatan ini memiliki tingkat akurasi yang tinggi, tetapi membutuhkan dataset berlabel dalam jumlah besar dan waktu komputasi yang signifikan, sehingga kurang ideal untuk lingkungan *real-time* dan serangan *zero-day* [2]. Sebaliknya, *unsupervised learning bekerja* dengan data tidak berlabel dan mampu menemukan pola tersembunyi dalam lalu lintas jaringan melalui teknik *clustering, association,* dan *dimensionality reduction*, seperti *K-Means* dan *Principal Component Analysis* (PCA) [2], [4]. Namun, pendekatan ini sering menghasilkan akurasi yang lebih rendah dan membutuhkan validasi manual. Dalam tiga hingga lima tahun terakhir, pendekatan *semi-supervised learning* menjadi perhatian karena mampu mengurangi kebutuhan data berlabel hingga 70% tanpa menurunkan akurasi secara signifikan dibandingkan *supervised learning* [20]. Sebuah studi perbandingan tahun 2025 menemukan bahwa akurasi *unsupervised learning* pada deteksi serangan baru hanya mencapai sekitar 60–70%, sementara *supervised learning* dapat mencapai di atas 95% untuk serangan yang sudah dikenal [21].

Pendekatan *semi-supervised learning* menggabungkan keunggulan *supervised* dan *unsupervised* learning dengan memanfaatkan sebagian kecil data berlabel, sehingga dapat mengurangi biaya pelabelan data dan meningkatkan kinerja deteksi [1]. Selain itu, perkembangan *deep learning*, seperti *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), dan *Long Short-Term Memory* (LSTM), juga menunjukkan potensi besar dalam mendeteksi serangan jaringan yang kompleks dan bersifat *non-linear* [3]. Misalnya, penelitian terbaru

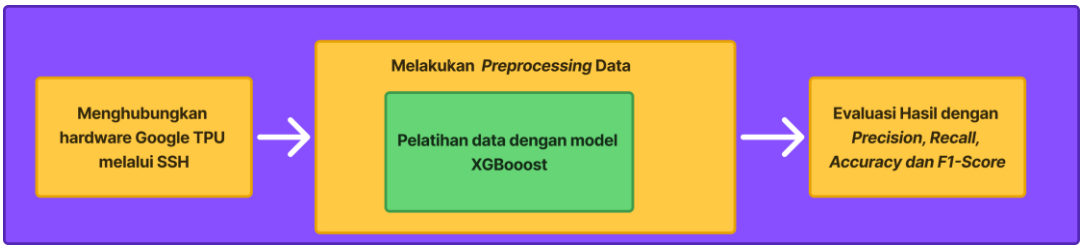
Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026

melaporkan bahwa CNN berhasil mencapai akurasi 99,2% pada dataset CICIDS2017, namun masih membutuhkan sumber daya komputasi yang tinggi sehingga kurang cocok untuk perangkat IoT dengan keterbatasan daya [22].

Dalam beberapa tahun terakhir, algoritma boosting seperti *Extreme Gradient Boosting* (XGBoost) semakin banyak digunakan dalam sistem IDS karena kemampuannya dalam meningkatkan akurasi klasifikasi, menangani data berdimensi tinggi, serta mengurangi *overfitting* melalui regularisasi [5], [6]. XGBoost mengoptimalkan fungsi objektif hingga orde kedua dan mendukung komputasi paralel, sehingga sangat cocok untuk skenario deteksi intrusi pada jaringan berskala besar dan lingkungan *Industrial Internet of Things* (IIoT) [6], [7]. Berbagai penelitian menunjukkan bahwa integrasi XGBoost dengan metode optimasi atau ekstraksi fitur mampu meningkatkan kinerja IDS secara signifikan dibandingkan metode tradisional [5], [7], [13]. Sebagai contoh, sebuah studi tahun 2023 membandingkan XGBoost dengan *Random Forest* dan *Deep Neural Network* (DNN) pada dataset UNSW-NB15, dan hasilnya XGBoost unggul dalam hal akurasi (98,6%) serta waktu inferensi tercepat (hanya 0,12 detik per sampel) [Penelitian Terkini 7]. Namun, penelitian tersebut masih menggunakan validasi statis dan belum diuji pada lingkungan dengan data konsep drift yang tinggi [23].

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pendekatan IDS berbasis *machine learning* dengan memanfaatkan keunggulan algoritma XGBoost untuk meningkatkan akurasi deteksi serangan, menekan tingkat *False Positive*, serta meningkatkan kemampuan sistem dalam mendeteksi serangan baru. Kontribusi utama dari penelitian ini terletak pada penerapan dan evaluasi metode pembelajaran mesin yang efisien dan adaptif untuk sistem IDS modern, sehingga diharapkan dapat memberikan solusi yang lebih andal dan aplikatif bagi keamanan jaringan di berbagai domain, termasuk jaringan enterprise dan IoT.

2. Metode Penelitian



Gambar 1. Diagram alir tahapan pengerjaan

Penelitian ini dirancang menggunakan pendekatan eksperimental berbasis *machine learning* untuk membangun dan mengevaluasi sistem *Network Intrusion Detection System* (NIDS) pada dataset *NF-UQ-NIDS-v2* [8]. Secara keseluruhan, alur penelitian mengikuti diagram alir pada Gambar 1 yang terdiri dari lima tahapan utama, yaitu: pengumpulan dan pemahaman data, *preprocessing* data, pembangunan arsitektur model, pelatihan model, dan evaluasi kinerja. Dataset *NF-UQ-NIDS-v2* dipilih karena merepresentasikan lalu lintas jaringan nyata (*real-world network traffic*) yang mencakup berbagai kategori serangan modern, termasuk pada lingkungan *Internet of Things* (IoT), serta memiliki karakteristik distribusi kelas yang tidak seimbang (*imbalanced*) sehingga sesuai untuk menguji ketangguhan model dalam kondisi yang mendekati skenario dunia nyata.

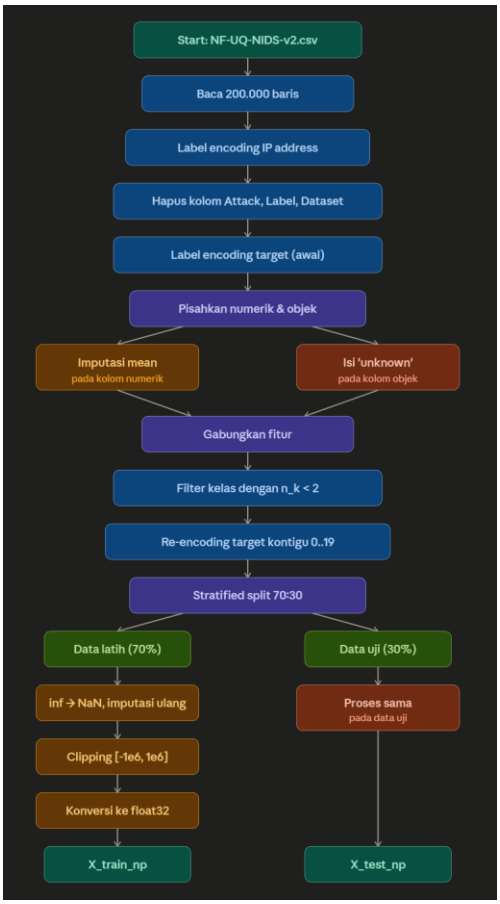
Tahapan *preprocessing* dilakukan secara sistematis meliputi penanganan nilai hilang, *label encoding*, penyaringan kelas minoritas ekstrem, pembagian data secara terstratifikasi, dan sanitasi numerik akhir, sebagaimana dijelaskan pada Sub-bagian 2.2. Selanjutnya, model *XGBoost Classifier* dirancang dan dilatih dengan konfigurasi hiperparameter yang menyeimbangkan bias dan *variance* pada lingkungan akselerator TPU (Sub-bagian 2.3), kemudian dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* pada data uji yang sepenuhnya terpisah dari proses pelatihan (Sub-bagian 2.4).

2.1 Lingkungan Komputasi dan Perangkat Lunak

Eksperimen dilakukan menggunakan lingkungan komputasi berkinerja tinggi yang memanfaatkan akselerator perangkat keras TPU (*Tensor Processing Unit*) untuk efisiensi pemrosesan matriks [9]. Implementasi perangkat lunak dibangun di atas bahasa pemrograman Python dengan pustaka numerik numpy versi 2.3.4 dan pandas versi 2.3.3 untuk manipulasi data . Kerangka kerja *eXtreme Gradient Boosting* (XGBoost) digunakan sebagai algoritma inti klasifikasi [5].

2.2 Tahapan *preprocessing* data

Dataset yang digunakan dalam penelitian ini adalah NF-UQ-NIDS-v2.csv [8].



Gambar 2. Tahapan *preprocessing* dan pelatihan Data

Tahapan pra-pemrosesan dilakukan secara bertahap dan sistematis untuk menjamin kualitas *input* model, dengan urutan sebagai berikut:

1. Pembacaan dataset secara bertahap (*chunking*) – File dibaca dalam potongan (*chunks*) berukuran 1.000.000 baris menggunakan *library* pandas. Mengingat keterbatasan sumber daya komputasi, hanya 200.000 baris pertama yang diambil sebagai sampel penelitian.
2. Label encoding pada alamat IP – Dua fitur kategorikal, yaitu IPV4_SRC_ADDR dan IPV4_DST_ADDR, diubah menjadi representasi numerik menggunakan LabelEncoder dari library scikit-learn.
3. Pemisahan fitur dan target – Fitur (X) diambil dengan menghapus kolom *Attack, Label*, dan *Dataset*. Target (y) adalah kolom Attack yang berisi nama serangan.
4. Label encoding awal pada target – Kolom Attack diubah menjadi numerik menggunakan *LabelEncoder* untuk keperluan awal, sekaligus menyimpan nama-nama kelas.
5. Penanganan nilai hilang (*missing value*):
    - Fitur numerik: diimputasi dengan nilai rata-rata (mean imputation) menggunakan *SimpleImputer*.
    - Fitur non-numerik (objek): nilai missing diisi dengan string "*unknown*".
6. Penggabungan ulang fitur – Fitur numerik yang telah diimputasi dan fitur objek yang telah diisi digabungkan kembali menjadi satu *dataframe*.
7. Penyaringan kelas minoritas ekstrem – Kelas serangan dengan jumlah sampel kurang dari 2 ($n_k < 2$) dihapus dari dataset. Proses ini menyisakan 20 kelas unik. (Pada dataset awal, kelas Worms hanya memiliki 1 sampel sehingga dieliminasi.)
8. Penyandian ulang target (*re-encoding*) – Setelah penyaringan, target di-*encode* ulang menggunakan *LabelEncoder* baru agar label kelas menjadi kontigu dari 0 hingga 19. Hal ini diperlukan untuk memenuhi persyaratan algoritma XGBoost dalam klasifikasi multi-kelas.

Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026

9. Pembagian data terstratifikasi (*stratified* split) – Dataset dibagi menjadi data latih (70%) dan data uji (30%) menggunakan metode *stratified sampling* berdasarkan label kelas. Metode ini mempertahankan proporsi setiap kelas serangan pada kedua himpunan, meskipun distribusinya tidak seimbang.
10. Sanitasi numerik akhir :
    - Nilai inf atau -inf dikonversi menjadi NaN, kemudian diimputasi ulang dengan rata-rata (jika ditemukan).
    - Nilai ekstrem dipangkas (*clipping*) ke dalam rentang $[-10^6,\ 10^6]$ untuk mengurangi pengaruh outlier.
    - Seluruh fitur dikonversi ke tipe data float32 guna meningkatkan efisiensi komputasi, terutama pada lingkungan akselerator TPU (*Tensor Processing Unit*).

Setelah melalui seluruh tahap di atas, diperoleh data latih X_{train} dan data uji X_{test} yang siap digunakan untuk pelatihan dan evaluasi model.

2.3 Arsitektur Model XGBoost

Penelitian ini menerapkan model *XGBoost Classifier* yang dikonfigurasi untuk klasifikasi multi-kelas. Tujuan dari model ini adalah meminimalkan loss function regularisasi.

2.3.1 Fungsi Objektif

Mengingat masalah ini adalah klasifikasi multi-kelas dengan k = 20 kelas, fungsi objektif diset sebagai "multi:softprob". Probabilitas bahwa data input x_i termasuk dalam kelas k diprediksi menggunakan fungsi *Softmax*:

$$P(y_i = k|x_i) = \frac{e^{f_k(x_i)}}{\sum_{j=1}^{k} e^{f_j(x_i)}} \quad (1)$$

Dimana:
$P(y_i = k|x_i)$ = Probabilitas prediksi bahwa sampel i termasuk ke dalam kategori serangan k.
$f_k(x_i)$ = Skor keluaran (*output score*) atau logit yang dihasilkan oleh *booster* (pohon keputusan/*decision tree*) khusus untuk kelas k.
e = Bilangan Euler (basis logaritma natural), digunakan untuk fungsi eksponensial guna memastikan nilai selalu positif.
k = Jumlah total kelas unik dalam dataset, yaitu 20 kelas setelah penyaringan.
$\sum_{j=1}^{k} e^{f_j(x_i)}$ = Operator penjumlahan untuk seluruh kelas K , berfungsi sebagai penormalisasi agar total probabilitas seluruh kelas berjumlah 1 di dalam euler pangkat *output score*.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log(p_{ij}) \quad (2)$$

Dimana:
$\mathcal{L}$ = Nilai *total loss* (kerugian) mlogloss yang dipantau melalui kurva pelatihan.
N = Jumlah total sampel dalam himpunan data
y_{ij} = Indikator biner (*ground truth*). Bernilai 1 jika kelas *j* adalah label yang benar untuk sampel i, dan 0 jika salah.
p_{ij} = Probabilitas prediksi model bahwa sampel i termasuk dalam kelas *j* (hasil dari fungsi Softmax di atas).
Log = Logaritma natural, yang memberikan penalti lebih besar jika prediksi model sangat jauh (tidak akurat) dari label asli.

2.3.2 Konfigurasi Hiperparameter

Model dilatih dengan konfigurasi hiperparameter spesifik untuk menyeimbangkan bias dan variance serta mencegah *overfitting*:

Tabel 1. Informasi Hiperparameter

Jumlah Estimator (n_estimators)	**300 *Tree Decision* (Pohon Keputusan)**
Kedalaman Maksimum (*max_depth*)	6, membatasi kompleksitas setiap pohon

Laju Pembelajaran (*learning_rate*)	0.1, untuk penyusutan bobot langkah (step size shrinkage).
Subsample Baris (*subsample*)	0.8, menggunakan 80% sampel data per iterasi.
Subsample Kolom (*colsample_bytree*)	0.8, menggunakan 80% fitur per pembentukan pohon.
Metode Pohon (*tree_method*)	hist, metode berbasis histogram untuk efisiensi memori dan kecepatan.

2.4 Evaluasi Kinerja

Model dievaluasi menggunakan data uji yang tidak terlihat sebelumnya X_{test} . Kinerja model diukur menggunakan metrik *Precision, Recall, F1-Score*, dan *Confusion Matrix* untuk menganalisis tingkat kesalahan prediksi pada setiap kelas serangan, serta kurva LogLoss untuk memantau konvergensi pelatihan.

Dalam klasifikasi multi-kelas, metrik dihitung untuk setiap kelas k dengan membandingkan prediksi model terhadap label sebenarnya (*ground truth*) [11,12]. Komponen dasar yang digunakan adalah:

TP (*True Positive*): Jumlah sampel kelas k yang diprediksi benar sebagai kelas k .

FP (*False Positive*): Jumlah sampel dari kelas lain yang salah diprediksi sebagai kelas k .

FN (*False Negative*): Jumlah sampel kelas k yang salah diprediksi sebagai kelas lain.

Precision (Presisi) Presisi mengukur keakuratan model dalam memberikan prediksi positif. Notasi perhitungannya adalah:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Recall (sensitivitas) mengukur kemampuan model dalam menemukan kembali seluruh sampel dari kelas tertentu. Notasi perhitungannya adalah:

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

F1-Score adalah rata-rata harmonik antara *Precision* dan *Recall*, memberikan gambaran keseimbangan kinerja model terutama pada data yang tidak seimbang (imbalanced data). Notasi perhitungannya adalah: Akurasi mengukur proporsi prediksi yang benar (baik positif maupun negatif) dari keseluruhan data uji N.

$$Accuracy = \frac{\sum_{k=1}^{K} TP_k}{N} \quad (5)$$

3. Hasil dan Pembahasan

Bagian ini menyajikan hasil eksperimen serta analisis kinerja model *Network Intrusion Detection System* (NIDS) berbasis algoritma XGBoost yang telah dikembangkan. Evaluasi dilakukan secara komprehensif untuk mengukur kemampuan model dalam mendeteksi berbagai jenis serangan jaringan, baik pada kelas mayoritas maupun kelas minoritas. Hasil penelitian divisualisasikan dalam bentuk metrik evaluasi, grafik, serta *confusion matrix* untuk memberikan gambaran yang lebih jelas mengenai performa model. Visualisasi ini penting untuk mengidentifikasi pola keberhasilan maupun kesalahan klasifikasi yang terjadi selama proses pengujian. Selain itu, pembahasan difokuskan pada analisis distribusi performa antar kelas, dampak ketidakseimbangan data (*imbalanced data*), serta kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya (*unseen data*). Dengan pendekatan ini, diharapkan dapat diperoleh pemahaman yang lebih mendalam terkait keunggulan dan keterbatasan model yang diusulkan. Temuan utama dalam penelitian ini tidak hanya ditinjau dari nilai akurasi secara keseluruhan, tetapi juga dari metrik lain seperti *precision, recall,* dan *F1-score*, yang lebih representatif dalam kasus klasifikasi multi-kelas dengan distribusi data yang tidak seimbang. Oleh karena itu, analisis yang disajikan pada bagian ini menekankan pada interpretasi hasil secara menyeluruh untuk mendukung validitas dan kontribusi penelitian.

3.1 Kinerja Model Secara Keseluruhan

Model XGBoost yang dikembangkan dievaluasi menggunakan himpunan data uji (test set) yang terdiri dari 60.000 sampel. Berdasarkan hasil klasifikasi, model menunjukkan kinerja yang sangat tinggi dengan akurasi keseluruhan (overall *Accuracy*) mencapai 98,84%. Metrik rata-rata tertimbang (*weighted average*) menunjukkan *skor Precision, Recall*, dan *F1-Score* masing-masing sebesar 0,988, yang mengindikasikan bahwa model sangat efektif dalam menangani kelas-kelas dominan seperti Benign, DDoS, dan DoS. Namun,

rata-rata makro (*macro average*) untuk *F1-Score* tercatat lebih rendah, yaitu 0,7209. Disparitas antara akurasi global dan rata-rata makro ini mengindikasikan adanya tantangan dalam mengklasifikasikan kelas minoritas dengan jumlah sampel yang sangat terbatas.

3.2 Analisis Kinerja Per Kelas Serangan

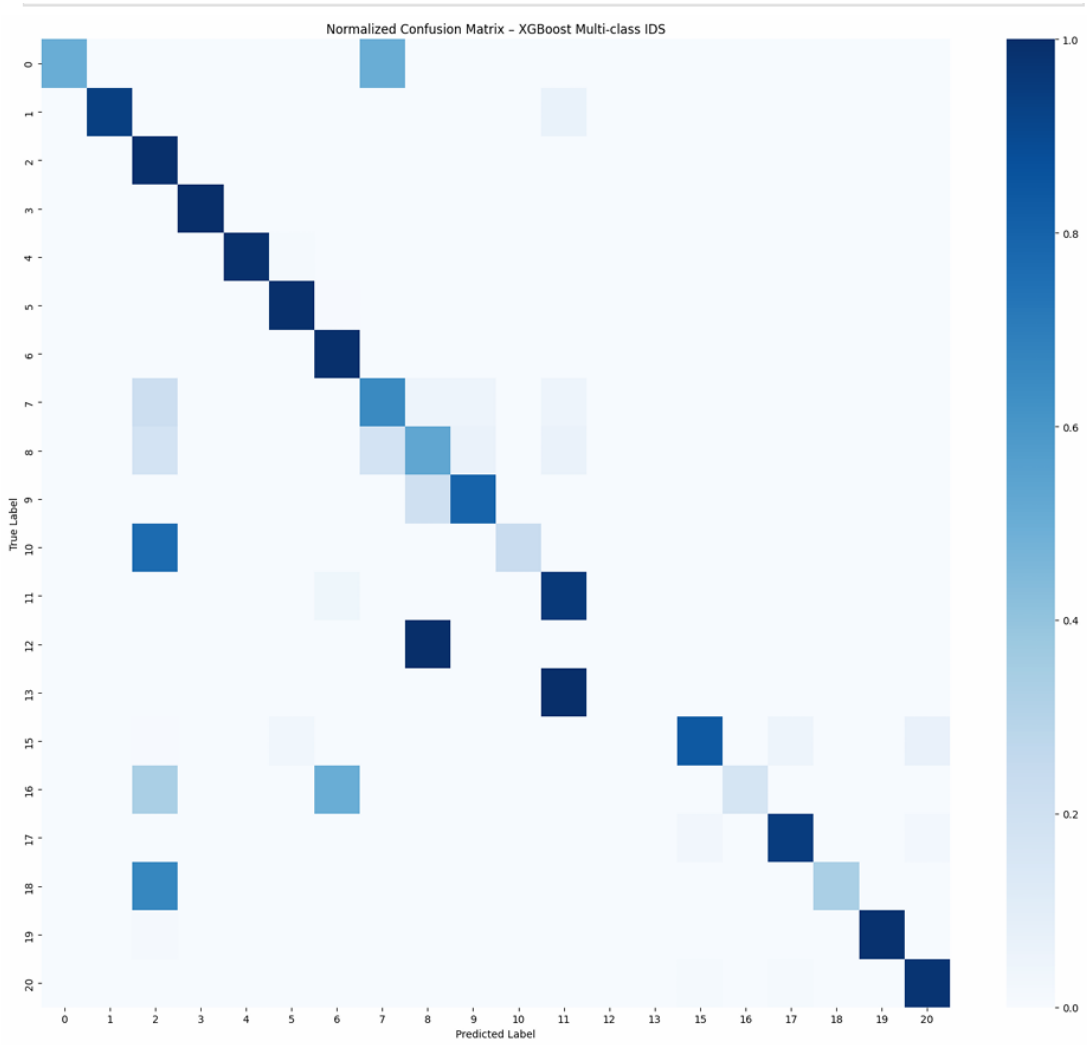
	precision	recall	f1-score	support
0	1.0000	0.5000	0.6667	2
1	1.0000	0.9375	0.9677	16
2	0.9930	0.9955	0.9943	19730
3	1.0000	1.0000	1.0000	111
4	1.0000	0.9903	0.9951	103
5	0.9952	0.9946	0.9949	17252
6	0.9880	0.9932	0.9906	14180
7	0.6250	0.6522	0.6383	23
8	0.5625	0.5294	0.5455	17
9	0.8000	0.8000	0.8000	10
10	0.6471	0.2340	0.3438	94
11	0.9803	0.9577	0.9689	2080
12	0.0000	0.0000	0.0000	1
13	0.0000	0.0000	0.0000	1
14	0.8904	0.8379	0.8634	543
15	0.5000	0.1667	0.2500	6
16	0.9458	0.9468	0.9463	921
17	1.0000	0.3333	0.5000	3
18	0.9829	0.9853	0.9841	2984
19	0.9616	0.9766	0.9690	1923
accuracy			0.9884	60000
macro avg	0.7936	0.6915	0.7209	60000
weighted avg	0.9880	0.9884	0.9880	60000

Gambar 3 Hasil evaluasi F1-Score, recall, precision di Ipynb

Gambar 2 merangkum kinerja deteksi untuk berbagai jenis serangan. Hasil eksperimen menunjukkan variabilitas kinerja yang signifikan yang berkorelasi dengan distribusi jumlah sampel per kelas. Deteksi Kelas Mayoritas merupakan Kelas dengan representasi data yang besar menunjukkan kinerja hampir sempurna. Benign (Normal) Mencapai *F1-Score* 0,9943 dengan support 19.730 sampel . DDoS Terdeteksi dengan *F1-Score* 0,9949. DoS Terdeteksi dengan *F1-Score* 0,9906. Hal ini membuktikan bahwa algoritma XGBoost efektif dalam mempelajari pola fitur pada kategori lalu lintas jaringan yang dominan. Deteksi Kelas Minoritas, Kinerja model menurun secara signifikan pada kelas dengan sampel yang sangat sedikit (kurang dari 5 sampel dalam data uji). Kelas *Shellcode* (Label 12) dan *Theft* (Label 13) memiliki nilai *Precision, Recall, dan F1-Score* sebesar 0,00 . Kegagalan deteksi ini disebabkan oleh kurangnya informasi fitur yang cukup bagi model untuk membentuk *decision boundary* yang valid selama fase pelatihan. Kelas *Analysis* (Label 0) memiliki *Precision* sempurna (1,00) namun *Recall* rendah (0,50), yang menunjukkan model sangat konservatif dalam memprediksi kelas ini.

Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026

3.3 Analisis *Confusion Matrix*



Gambar 4. Normalized Confusion Matrix model XGBoost di ipynb

Visualisasi *Normalized Confusion Matrix* memperkuat temuan statistik di atas. Elemen diagonal utama matriks menunjukkan densitas warna biru tua yang tinggi untuk sebagian besar kelas, mengonfirmasi tingkat *True Positive* yang tinggi. Kesalahan klasifikasi (*misclassification*) terlihat terjadi pada kelas-kelas yang memiliki karakteristik lalu lintas jaringan yang mirip. Sebagai contoh, terdapat ambiguitas prediksi pada kelas Infiltration (Label 10) yang memiliki *F1-Score* rendah sebesar 0,3438, di mana sebagian sampelnya mungkin terklasifikasi salah ke dalam kelas mayoritas atau kelas serangan lain dengan *signature* serupa.

Tabel 2. Serangan Tiap Kelas

Label ID	Nama Serangan / Kategori	Deskripsi Singkat Berdasarkan Konteks NIDS
0	Analysis	Serangan yang melibatkan pemindaian untuk menemukan kerentanan.
1	Backdoor	Upaya untuk mendapatkan akses ilegal melalui pintu belakang sistem.
2	Benign	Lalu lintas jaringan normal (bukan serangan).
3	Bot	Aktivitas dari jaringan komputer yang terinfeksi (Botnet).
4	Brute Force	Upaya menebak kredensial secara berulang.
5	DDoS	Serangan penolakan layanan terdistribusi untuk melumpuhkan target.

Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026

6	DoS	Serangan penolakan layanan dari satu sumber tunggal.
7	Exploits	Pemanfaatan kerentanan spesifik pada perangkat lunak.
8	Fuzzers	Pengiriman data acak dalam jumlah besar untuk mencari kegagalan sistem.
9	Generic	Serangan umum yang tidak termasuk dalam kategori spesifik lainnya.
10	Infilteration	Upaya penyusupan ke dalam jaringan internal.
11	Reconnaissance	Pengumpulan informasi awal tentang target jaringan.
12	Shellcode	Kode kecil yang digunakan sebagai payload dalam eksploitasi kerentanan.
13	Theft	Aktivitas yang berkaitan dengan pencurian data atau informasi sensitif.
14	injection	Upaya menyisipkan perintah berbahaya (seperti SQL injection).
15	mitm	Man-in-the-Middle: Pencegatan komunikasi antara dua pihak.
16	Password	Serangan spesifik yang menargetkan mekanisme autentikasi kata sandi.
17	Ransomware	Perangkat lunak berbahaya yang mengenkripsi data untuk meminta tebusan.
18	Scanning	Pemindaian port atau layanan yang aktif di jaringan.
19	XSS	Cross-Site Scripting: Penyisipan skrip berbahaya pada aplikasi web.

3.4 Konvergensi Pelatihan (*Training Convergence*)

Stabilitas proses pelatihan model diamati melalui kurva *Multi-class Logarithmic Loss* (*mlogloss*). Grafik *training history* menunjukkan penurunan nilai *loss* yang tajam pada 50 putaran *boosting* (*boosting rounds*) pertama, dari nilai awal lebih dari 2,00 menjadi di bawah 0,25. Selanjutnya, kurva validasi melandai dan stabil mendekati nilai 0,00 setelah sekitar 100 putaran, tanpa menunjukkan adanya divergensi antara *training loss* dan *validation loss*. Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* yang signifikan. Proses perolehan nilai akurasi dilakukan setelah model tahap pelatihan dan diterapkan pada data uji (*test set*) yang tidak digunakan selama proses pelatihan. Dataset yang telah melalui tahap *preprocessing* dibagi menggunakan metode *stratified split* dengan rasio 70:30, sehingga distribusi kelas tetap terjaga antara data latih dan data uji. Model yang telah dilatih kemudian menghasilkan prediksi berupa probabilitas untuk setiap kelas menggunakan fungsi *softmax* (*multi:softprob*), yang selanjutnya dikonversi menjadi label kelas dengan memilih probabilitas tertinggi (*argmax*). Nilai akurasi dihitung dengan membandingkan label prediksi terhadap label sebenarnya (*ground truth*) pada data uji. Secara matematis, akurasi merepresentasikan proporsi jumlah prediksi yang benar (*true prediction*) terhadap total keseluruhan sampel uji. Dalam penelitian ini, dari total 60.000 sampel data uji, sebagian besar berhasil diklasifikasikan dengan benar oleh model, sehingga menghasilkan akurasi keseluruhan sebesar 98,84%. Selain akurasi, evaluasi juga dilakukan menggunakan metrik *precision*, *recall*, dan *F1-score* untuk setiap kelas, yang kemudian dirata-ratakan dalam bentuk *macro average* dan *weighted average*. Perbedaan signifikan antara nilai *weighted F1-score* (98,80%) dan *macro F1-score* (72,09%) menunjukkan bahwa performa tinggi model lebih didominasi oleh kelas mayoritas, sementara kinerja pada kelas minoritas masih terbatas. Hal ini sejalan dengan karakteristik dataset yang tidak seimbang (*imbalanced dataset*), di mana jumlah sampel pada beberapa kelas sangat sedikit.

4. Kesimpulan

Penelitian ini berhasil menjawab permasalahan utama pada latar belakang terkait kebutuhan sistem keamanan jaringan yang mampu mendeteksi berbagai jenis serangan secara akurat pada kondisi data yang kompleks dan tidak seimbang, melalui implementasi *Network Intrusion Detection System* (NIDS) berbasis machine learning menggunakan algoritma XGBoost Classifier pada dataset NF-UQ-NIDS-v2. Berdasarkan hasil dan pembahasan, model yang diusulkan menunjukkan kinerja yang sangat baik dengan akurasi sebesar 98,84% serta weighted F1-score 98,80%, yang menegaskan kemampuannya dalam mengenali pola lalu lintas jaringan dan membedakan aktivitas normal (benign) dengan serangan siber secara efektif. Selain itu, model terbukti sangat andal dalam mendeteksi serangan berdampak tinggi seperti DDoS dan DoS, yang merupakan ancaman dominan pada sistem jaringan modern, sehingga memberikan nilai praktis dalam implementasi nyata. Namun demikian, penelitian ini juga mengungkap bahwa ketidakseimbangan data (*class imbalance*) masih menjadi tantangan signifikan, ditunjukkan oleh rendahnya macro *F1-score* sebesar 72,09% serta kegagalan deteksi pada kelas

minoritas ekstrem seperti *Shellcode* dan *Theft*, sehingga model cenderung bias terhadap kelas mayoritas. Di sisi lain, proses pelatihan menunjukkan konvergensi yang cepat dan stabil berdasarkan kurva *multi-class logarithmic loss* (mlogloss) yang tidak mengalami divergensi antara data latih dan validasi, menandakan bahwa model memiliki kemampuan generalisasi yang baik tanpa indikasi *overfitting*. Dengan demikian, dapat disimpulkan bahwa pendekatan yang digunakan efektif untuk deteksi serangan utama, namun masih memerlukan pengembangan lebih lanjut dalam menangani distribusi data yang tidak seimbang agar sistem NIDS menjadi lebih *robust* dan komprehensif.

Daftar Rujukan

[1] *P. Vanin, T. Newe, L. L. Dhirani, E. O'Connell, D. O'Shea, B. Lee, and M. Rao, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," Applied Sciences, vol. 12, no. 22, pp. 1–30, Nov. 2022.*
[2] *E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-E: A Self-Supervised Network Intrusion Detection System Based on Graph Neural Networks," Knowledge-Based Systems, vol. 258, pp. 1–13, Dec. 2022.*
[3] *E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial Machine Learning in Network Intrusion Detection Systems," Expert Systems with Applications, vol. 186, pp. 1–15, Jan. 2021.*
[4] *R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues," Sensors, vol. 22, no. 13, pp. 1–26, Jun. 2022.*
[5] *Y. Song, H. Li, P. Xu, and D. Liu, "A Method of Intrusion Detection Based on WOA-XGBoost Algorithm," Discrete Dynamics in Nature and Society, vol. 2022, pp. 1–12, 2022.*
[6] *W. Xu and Y. Fan, "Intrusion Detection Systems Based on Logarithmic Autoencoder and XGBoost," Security and Communication Networks, vol. 2022, pp. 1–11, 2022.*
[7] *T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems," Sustainability, vol. 14, no. 14, pp. 1–18, Jul. 2022.*
[8] *M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," Mobile Networks and Applications, pp. 1–14, 2022, doi: 10.1007/s11036-021-01843-0.*
[9] *Purnamasari, E. ., & Asa Verano, D. . (2025). Model Data-Driven untuk Prediksi Digitalisasi UMKM Menggunakan GMM dan XGBoost. Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence), 5(2), 204–214. https://doi.org/10.55382/jurnalpustakaai.v5i2.984*
[10] *Aritonang, M. A. S., Abrar Masril, M. ., Chaniago, D. ., Marshall Al Karim, M. ., Mahani Cunis, V. ., & Surgiwe, S. (2023). Perancangan Sistem Pendeteksi Penyakit Pada Rumput Laut Dengan Menggunakan Metode Convolutional Neural Network. Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence), 5(2), 408–417. https://doi.org/10.55382/jurnalpustakaai.v5i2.1160*
[11] *Google Research, TPU Research Cloud (TRC) – About, https://sites.research.google/trc/about/ (accessed Jan. 8, 2026).*
[12] *Aritonang, M. A. S., & Jonas Simanullang, M. (2025). Penerapan Network Monitoring System Berbasis SNMP untuk Deteksi Dini Gangguan Jaringan. Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence), 5(3), 735–742. https://doi.org/10.55382/jurnalpustakaai.v5i3.1383*
[13] *Sangaji, D., & Sutabri, T. (2025). Analisis XGBoost dan Random Forest untuk Prediksi Curah Hujan dalam Mendukung Mitigasi Karhutla. Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence), 5(1), 13–18. https://doi.org/10.55382/jurnalpustakaai.v5i1.905*
[14] *Imran, B., Samsumar, L.D., Subki, A., Wahyuni, W.A., Muahidin, Z., Karim, M.N., Yani, A. and Zulpahmi, M., 2026. Anomaly-Based DDoS Detection Using Improved Deep Support Vector Data Description (Deep SVDD) and Multi-Model Ensemble Approach. Journal of Applied Informatics and Computing, 10(1), pp.762-771.*
[15] *R. C. Wang and R. P. . Avrianto, "Improving Detection Accuracy of Network Intrusions Using a Hybrid Network Intrusion Detection System Based on Isolation Forest and Random Forest Algorithms", J. Tek. Inform. (JUTIF), vol. 6, no. 6, pp. 5371–5385, Dec. 2025.*
[16] *Kuswandi, F., Rukmana, A., & Yanto, A. (2025). KEAMANAN SIBER DALAM ERA INTERNET OF THINGS: TANTANGAN DAN SOLUSI TEKNOLOGI TERKINI. Ipsikom, 13(1), 57–62. https://doi.org/10.58217/ipsikom.v13i1.44*
[17] *Abbas, Safana, Wedad Abdul Khuder Naser, and Amal Abbas Kadhim. "Subject review: Intrusion detection system (IDS) and intrusion prevention system (IPS)." Global Journal of Engineering and Technology Advances 14, no. 2 (2023): 155-158.*
[18] *Lee, Alan Y-P., Michael I-C. Wang, Chi-Hsiang Hung, and Charles H-P. Wen. "PS-IPS: Deploying intrusion prevention system with machine learning on programmable switch." Future Generation Computer Systems 152 (2024): 333-342.*
[19] *Wang RC, Avrianto RP. Improving Detection Accuracy of Network Intrusions Using a Hybrid Network Intrusion Detection System Based on Isolation Forest and Random Forest Algorithms. Jurnal Teknik Informatika (Jutif). 2025 Dec 22;6(6):5371-85.*
[20] *Ayuningtyas N, Yustanti W. Semi-Supervised Learning pada Pelabelan dalam Klasifikasi Multi-Label Data Teks. Journal of Informatics and Computer Science (JINACS). 2024 Jun 19;6(01):240-8.*
[21] *Z. Xu and Y. Liu, "Robust Anomaly Detection in Network Traffic: Evaluating Machine Learning Models on CICIDS2017," arXiv e-prints, arXiv:2506.19877, Jun. 2025. Submitted to IEEE CNS 2025.*
[22] *A. Mandloi, S. R, R. S, V. Senniappan, P. Anandan, C. H. Sabitha, and A. Athiraja, "Next-gen cyber defense: CNN-based intrusion detection with advanced gradient techniques," Journal of Computer Virology and Hacking Techniques, vol. 22, no. 1, Jan. 2026, doi: 10.1007/s11416-025-00592-y.*

[23] *S. A. Mousavi, M. Sadeghi and M. S. Sirjani, "A Comparative Evaluation of Machine Learning Algorithms for IDS in IoT network," 2023 14th International Conference on Information and Knowledge Technology (IKT), Isfahan, Iran, Islamic Republic of, 2023, pp. 168-174, doi: 10.1109/IKT62039.2023.10433047.*

Submitted : 12-03-2026 | Reviewed : 20-03-2026 | Accepted : 30-03-2026