Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

Evaluating Ensemble Versus Non-Ensemble Machine Learning Performance with Preprocessing Techniques for IoT Intrusion Detection on CICIoT2023

Febrian Sabila Firdaus[1], Puspanda Hatta[*2], Basori[3]

[1,2,3]Informatic and Computer Engineering Education, Universitas Sebelas Maret, Indonesia

Email: [2]hatta.puspanda@staff.uns.ac.id

Received : Oct 29, 2025; Revised : Nov 1, 2025; Accepted : Nov 3, 2025; Published : Feb 15, 2026

Abstract

The rapid expansion of the Internet of Things (IoT) introduces significant security vulnerabilities, exposing networks to sophisticated attacks. Developing effective Intrusion Detection Systems (IDS) is critical, yet many machine learning benchmarks rely on outdated datasets. This study provides a comprehensive comparative evaluation of ensemble and non-ensemble machine learning models for multiclass attack classification using the modern and complex CICIoT2023 dataset. The methodology involves robust preprocessing, including random undersampling to address extreme class imbalance and a hybrid feature selection approach combining Mutual Information (MI) and Random Forest Feature Importance (RFFI). Models, including Naive Bayes, Logistic Regression, SVM, Random Forest, and XGBoost, were evaluated using stratified 5-fold cross-validation (K=5) with default hyperparameters. The results demonstrate that ensemble models consistently and significantly outperform non-ensemble models. XGBoost achieved the highest and most stable performance, yielding a mean F1-score of 0.8889 ± 0.0008 across the K-folds, and a final macro F1-score of 0.8891 on the test set. This research confirms the superiority of ensemble methods for complex IoT traffic and quantitatively highlights the critical role of preprocessing. Notably, scaling was proven essential for non-ensemble models, drastically improving Logistic Regression's F1-score from an unstable 0.6280 to 0.7691.

Keywords : *CICIoT2023, Ensemble Learning, Feature Selection, Intrusion Detection, Machine Learning, Preprocessing*

1. PENDAHULUAN

Internet of Things (IoT) telah berkembang menjadi visi komputasi di mana berbagai perangkat fisik, mulai dari sensor sederhana hingga aktuator, saling terhubung untuk berbagi data dan mengoordinasikan keputusan [1]. Konsep ini memungkinkan perangkat untuk berkomunikasi melalui jaringan internet tanpa intervensi manusia [2]. Didorong oleh kemajuan teknologi sensor dan komputasi awan, jumlah perangkat IoT yang terhubung diproyeksikan akan mencapai puluhan miliar unit pada tahun 2030, menandakan perannya yang semakin integral di berbagai sektor.

Seiring dengan adopsi yang masif ini, isu keamanan dan privasi menjadi tantangan utama [3]. Sistem IoT yang heterogen, melibatkan beragam protokol dan perangkat keras, menciptakan permukaan serangan yang luas dan menyulitkan penerapan standarisasi keamanan [4] Ancaman siber terhadap IoT terus berkembang dan menjadi semakin canggih [5]. Serangan-serangan ini dapat berkisar dari pencurian data personal [6] hingga pembentukan botnet masif, seperti *Mirai*, yang mampu melumpuhkan infrastruktur internet melalui serangan *DDoS* [7].

Menghadapi tantangan ini, komunitas riset telah berfokus pada pendekatan komputasi cerdas, terutama *Machine Learning* (ML) dan *Deep Learning* (DL). Banyak survei dan tinjauan sistematis telah memetakan penggunaan ML dan DL untuk Sistem Deteksi Intrusi Jaringan (NIDS) [8][9][10].

Pendekatan DL telah diteliti secara ekstensif untuk mendeteksi anomali jaringan atau serangan *botnet* yang canggih [11][12]. Di sisi lain, model ML klasik, khususnya *ensemble learning*, juga menunjukkan hasil yang sangat menjanjikan. Tinjauan komprehensif oleh [13] menyoroti berbagai solusi dan tantangan ML dalam keamanan IoT. Penelitian lain oleh [14] secara spesifik menunjukkan bahwa metode *ensemble* seperti *Random Forest* dan *XGBoost* serta metode *stacking* [15] seringkali mengungguli algoritma ML tunggal dalam tugas klasifikasi serangan [16].

Meskipun bermanfaat, penelitian-penelitian yang bergantung pada dataset lama ini memiliki keterbatasan. Sebagai contoh, studi oleh [17] melakukan *benchmarking* beberapa model ML pada dataset *CIC-IDS2017*, memberikan metrik performa dasar yang penting. Demikian pula, penelitian lain [18][19] telah membandingkan model-model seperti SVM, RF, dan NB untuk deteksi serangan IoT, dan menyoroti dampak kritis dari seleksi fitur terhadap performa. Untuk merangkum kondisi s*tate-of-the-art* dan memposisikan penelitian ini dengan jelas, Tabel 1 menyajikan gambaran komparatif dari studi-studi relevan terbaru.

Tabel 1. Tinjauan pustaka studi komparatif terkait

Penulis	Dataset	Metode / Model yang dibandingkan	Temuan Kunci
[18]	Dataset IoT (N-BaIoT, BoT-IoT)	Perbandingan model ML (SVM, RF, LR, k-NN).	Random Forest (RF) secara konsisten memberikan hasil akurasi tertinggi.
[19]	NSL-KDD	Evaluasi ML (NB, RF, DT) dengan feature selection.	Feature selection secara signifikan meningkatkan akurasi deteksi (terutama untuk RF).
[24]	NSL-KDD, UNSW-NB15	Metode hybrid: Feature Selection (RFE) + Data Reduction (LOF) dengan DT.	Preprocessing hybrid (seleksi & reduksi) secara signifikan meningkatkan performa deteksi.
[29]	BoT-IoT	Random Forest (RF) menunjukkan performa terbaik pada dataset yang tidak seimbang.	Random Forest (RF) menunjukkan performa terbaik pada dataset yang tidak seimbang.

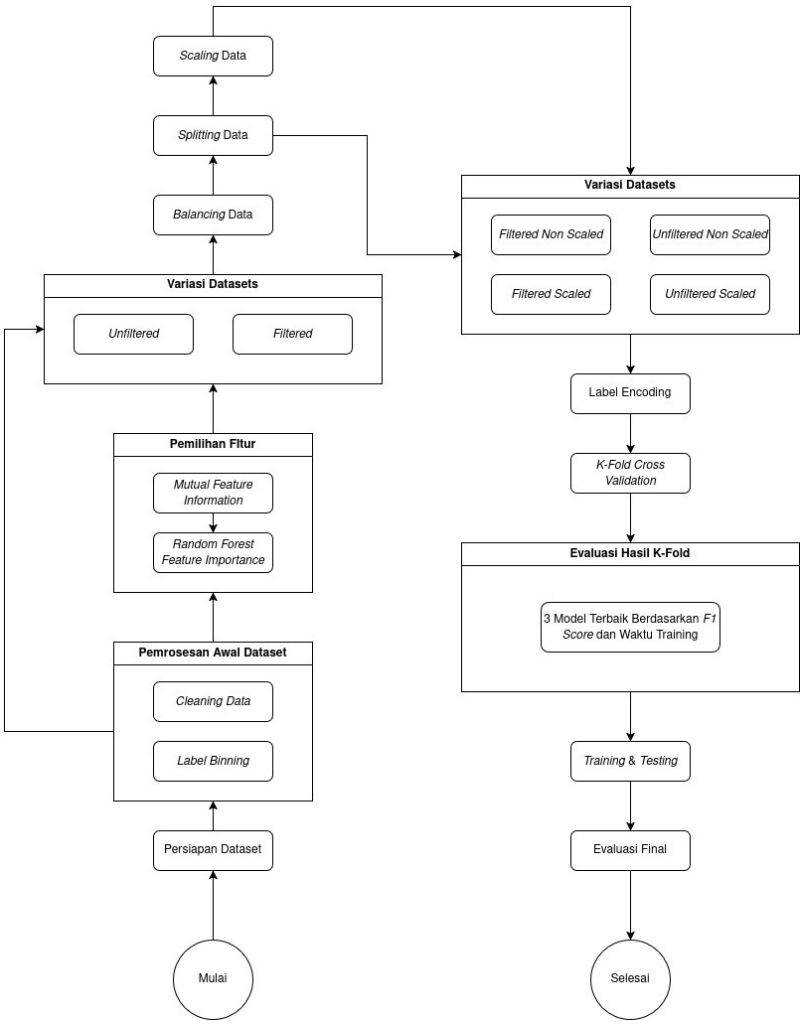
Sebuah tinjauan oleh [20] juga secara spesifik mengidentifikasi adanya fitur yang tidak valid pada dataset BoT-IoT yang dapat menyebabkan *overfitting* pada model. Keterbatasan ini menyoroti dua kebutuhan penting yang ditangani oleh penelitian ini yaitu pentingnya *preprocessing* data yang cermat [21], termasuk penyeimbangan data [22] dan seleksi fitur yang kuat [23] [24] dan kebutuhan akan evaluasi model pada dataset yang lebih baru dan lebih realistis.

Sebagai respons terhadap keterbatasan ini, dataset *CICIoT2023* diperkenalkan sebagai *benchmark* yang lebih modern, dirancang untuk merepresentasikan skenario serangan nyata dengan kompleksitas yang lebih tinggi [25]. Meskipun demikian, masih terdapat kesenjangan penelitian, dimana studi komparatif yang sistematis antara model *ensemble* dan *non-ensemble* sebagai perwakilan ML klasik pada dataset baru ini masih sangat terbatas. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan tersebut. Tujuan utama dari penelitian ini adalah untuk melakukan evaluasi komparatif yang sistematis terhadap model *ensemble* (*Random Forest*, *XGBoost*) dan *non-ensemble* (*Naive Bayes*, *Logistic Regression*, SVM) guna menentukan pendekatan yang paling efektif untuk deteksi intrusi IoT pada dataset *CICIoT2023*. Secara khusus, kebaruan dari penelitian ini terletak pada investigasi eksplisit mengenai dampak dari variasi *preprocessing*, termasuk seleksi fitur (MI+RFFI) dan normalisasi data, terhadap performa dan efisiensi komputasi dari model-model tersebut.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

2. METODE PENELITIAN

Penelitian ini merupakan penelitian kuantitatif dengan metode eksperimen. Alur penelitian dirancang secara sistematis, seperti yang dirangkum dalam Gambar 1, untuk mengevaluasi dan membandingkan performa model *machine learning*. Tahapan utama meliputi deskripsi dataset, *preprocessing* data yang komprehensif, seleksi fitur, definisi model, dan skenario evaluasi model.



Gambar 1. Flowchart penelitian

2.1. Dataset Penelitian: CICIoT2023

Data yang digunakan merupakan data sekunder, yaitu dataset *CICIoT2023* yang dikembangkan oleh *Canadian Institute for Cybersecurity* [25]. Dataset ini merupakan sekumpulan data *real-time* yang dirancang khusus untuk mengevaluasi serangan skala besar dalam lingkungan IoT. *CICIoT2023* dipilih karena secara spesifik dirancang untuk mengatasi keterbatasan dataset IoT sebelumnya, seperti BoT-IoT [20], yang sering dikritik karena kurangnya keragaman serangan dan potensi fitur yang tidak valid. Dataset ini mencakup topologi jaringan yang luas dan skenario lalu lintas yang realistis [25]. Secara bawaan, dataset ini memiliki 39 fitur, seperti yang dirinci pada Tabel 2. Fitur-fitur ini mencerminkan karakteristik *flow-based* dari lalu lintas jaringan, yang mencakup informasi statistik (seperti *Min*, *Max*, AVG), detail protokol (seperti TCP, UDP, HTTP), dan metadata paket (seperti *Header_Length* dan f*lag counts*). Dataset ini juga memiliki 33 jenis serangan spesifik yang dikelompokkan ke dalam 7 kelas utama, termasuk DDoS, *Mirai*, *Spoofing*, DoS, dan *Recon*. Keragaman fitur dan label serangan inilah yang membuatnya menjadi *benchmark* yang relevan untuk penelitian ini.

Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

Tabel 2. Fitur pada dataset CIC IotT 2023

No	Fitur	Tipe Data	No	Fitur	Tipe Data	No	Fitur	Tipe Data
1	Header_Length	float	16	HTTP	float	31	LLC	float
2	Protocol Type	int	17	HTTPS	float	32	Tot sum	int
3	Time_To_Live	float	18	DNS	float	33	Min	int
4	Rate	float	19	Telnet	float	34	Max	int
5	fin_flag_number	float	20	Variance	float	35	AVG	float
6	syn_flag_number	float	21	SMTP	float	36	Std	float
7	rst_flag_number	float	22	SSH	float	37	Tot size	float
8	psh_flag_number	float	23	IRC	float	38	IAT	float
9	ack_flag_number	float	24	TCP	float	39	Number	int
10	ece_flag_number	float	25	UDP	float	40	Label	object
11	cwr_flag_number	float	26	DHCP	float			
12	ack_count	int	27	ARP	float			
13	syn_count	int	28	ICMP	float			
14	fin_count	int	29	IGMP	float			
15	rst_count	int	30	IPv	float			

2.2. Preprocessing Data

Preprocessing data merupakan langkah krusial dalam *supervised learning* untuk meningkatkan kualitas data dan efektivitas model [21]. Tahap ini diawali dengan pembersihan data untuk menghilangkan nilai tidak valid. Selanjutnya, dilakukan teknik label *binning* untuk mengelompokkan 33 label serangan yang ada di dataset asli ke dalam lima kelas utama yang digunakan dalam penelitian ini, yaitu DoS, *Mirai*, *Spoofing*, *Recon*, dan *Normal*.

Langkah selanjutnya adalah menangani ketidakseimbangan kelas. Dataset deteksi intrusi jaringan, termasuk CICIoT2023, dikenal sering mengalami ketidakseimbangan kelas yang ekstrem [22], di mana data serangan sebagai kelas mayoritas jauh melampaui data normal sebagai kelas minoritas. Jika tidak ditangani, model akan cenderung sangat bias dan memiliki performa buruk dalam mengenali kelas minoritas. Untuk mengatasi masalah ini, penelitian ini menerapkan proses *balancing* data menggunakan teknik *random undersampling*. Pendekatan ini dipilih karena dua alasan utama: pertama, terbukti efektif dalam meningkatkan kemampuan model untuk mengenali pola pada kelas minoritas dengan mengurangi dominasi prediktif dari kelas mayoritas [22]. Kedua, mengingat volume dataset yang sangat besar, *undersampling*, yang merupakan teknik mengurangi sampel mayoritas, jauh lebih efisien secara komputasi untuk proses pelatihan dibandingkan *oversampling*, yaitu teknik menambah sampel minoritas, seperti SMOTE [26].

Terakhir, data numerik dinormalisasi menggunakan *StandardScaler*. Langkah ini mutlak diperlukan karena fitur-fitur dalam dataset CICIoT2023 memiliki rentang nilai dan unit yang sangat beragam; sebagai contoh, fitur *rate* bernilai kecil sementara fitur *tot_size* bernilai besar. Model *non-ensemble* yang dievaluasi dalam penelitian ini, khususnya *Logistic Regression* yang berbasis optimasi gradien dan *Support Vector Machine* yang berbasis perhitungan jarak, sangat sensitif terhadap perbedaan skala ini. *StandardScaler* dipilih karena kemampuannya mengubah skala setiap fitur agar memiliki rata-rata 0 dan standar deviasi 1. Normalisasi ini memastikan bahwa tidak ada satu fitur pun yang mendominasi proses pembelajaran hanya karena memiliki variansi numerik yang lebih besar, sehingga memungkinkan model untuk konvergen secara efisien dan stabil.

Seluruh tahapan *preprocessing* ini, mulai dari pembersihan data, *undersampling*, hingga *scaling*, diimplementasikan menggunakan pustaka *scikit-learn* dalam lingkungan *python*. Implementasi ini menggunakan parameter default yang disediakan oleh *scikit-learn* untuk setiap fungsi, seperti

StandardScaler dan *RandomUnderSampler* dari pustaka *imbalanced-learn*, untuk memastikan objektivitas dan reproduktibilitas penelitian.

2.3. Seleksi Fitur

Tahap seleksi fitur bertujuan untuk mengidentifikasi dan mengeliminasi fitur-fitur yang dianggap tidak berpengaruh signifikan terhadap hasil akhir model. Proses ini menggunakan gabungan teknik *Mutual Information* (MI) dan *Random Forest Feature Importance* (RFFI).

MI dipilih sebagai metode filter karena kemampuannya yang kuat dalam mengukur ketergantungan non-linear antara sebuah fitur dengan variabel target [23]. Teknik ini berfokus pada pemilihan fitur yang memiliki relevansi maksimal terhadap kelas target [27].

Secara matematis, MI mengukur jumlah informasi yang diperoleh dari satu variabel acak (fitur X) melalui variabel acak lainnya (kelas target Y). Formula MI didefinisikan sebagai berikut:

$$I(X,Y) = \sum_{X} \sum_{Y} p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) (1)$$

Di mana p(x,y) adalah probabilitas gabungan (*joint probability*) dari fitur X dan kelas Y, sementara p(x) dan p(y) adalah probabilitas marginal (*marginal probability*) independen dari X dan Y. Nilai MI yang tinggi menunjukkan ketergantungan yang kuat antara fitur dan kelas target.

Sementara itu, RFFI dipilih sebagai metode *embedded* yang efektif untuk deteksi intrusi di IoT [28]. Teknik ini mengevaluasi pentingnya fitur berdasarkan kontribusi aktualnya dalam pengambilan keputusan model, biasanya dengan mengukur seberapa besar fitur tersebut berkontribusi pada pengurangan *impurity* seperti *Gini impurity* di seluruh pohon dalam *forest*.

Pendekatan gabungan ini digunakan untuk mendapatkan perspektif yang lebih komprehensif. MI bertindak sebagai penyaring awal untuk relevansi statistik, sedangkan RFFI memvalidasi kontribusi praktis fitur tersebut terhadap kinerja model. Kombinasi ini diyakini menghasilkan subset fitur yang lebih optimal dan stabil. Tahap ini menghasilkan dua varian dataset utama untuk eksperimen: pertama, dataset *unfiltered* yang memuat seluruh 39 fitur asli, dan kedua, dataset *filtered* yang hanya memuat 21 fitur hasil seleksi.

2.4. Model Machine Learning

Penelitian ini mengadopsi pendekatan *supervised learning*, di mana model dilatih menggunakan data berlabel untuk mengklasifikasikan jenis serangan. Model yang dievaluasi dibagi menjadi dua kelompok: *non-ensemble* dan *ensemble*.

2.4.1. Model Non-Ensemble

Merupakan algoritma tunggal yang membuat prediksi berdasarkan satu fungsi pemisah atau probabilistik. Model-model ini cenderung lebih ringan dan cepat dalam proses pelatihan, namun memiliki keterbatasan dalam menangani kompleksitas data yang tinggi [29]. Model *non-ensemble* yang digunakan, dipilih karena sering dijadikan sebagai *baseline* perbandingan dalam penelitian deteksi intrusi [29][30]. Untuk meningkatkan reproduktibilitas penelitian, implementasi model-model ini menggunakan pustaka *scikit-learn* pada lingkungan *python*. Seluruh model *non-ensemble* dijalankan menggunakan parameter *default* yang disediakan oleh pustaka tersebut untuk memastikan objektivitas perbandingan.

- *Naive Bayes* (NB): Model ini adalah pengklasifikasi probabilistik yang didasarkan pada Teorema Bayes dengan asumsi independensi antar fitur. NB dipilih karena kesederhanaannya, kecepatan komputasinya yang sangat tinggi, dan kemampuannya menangani data dalam jumlah besar.

Model ini sering digunakan sebagai *baseline* yang cepat dalam berbagai tugas klasifikasi, termasuk deteksi intrusi [30].

- *Logistic Regression* (LR): Ini adalah model statistik yang memprediksi probabilitas suatu kelas dengan menggunakan fungsi logistik. LR dipilih karena merupakan model linear yang fundamental, mudah diinterpretasi, dan umum digunakan sebagai *baseline* untuk perbandingan performa. Meskipun sederhana, LR bisa sangat efektif pada masalah yang dapat dipisahkan secara linear [29].
- *Support Vector Machine* (SVM): SVM bekerja dengan menemukan *hyperplane* optimal yang paling baik memisahkan titik data dari kelas yang berbeda dalam ruang berdimensi tinggi. SVM dipilih karena landasan teoretisnya yang kuat pada prinsip *Structural Risk Minimization* (SRM) [31]. Berbeda dengan metode tradisional yang sering berfokus pada *Empirical Risk Minimization* (ERM) atau meminimalkan kesalahan pada data latih, SRM bertujuan untuk meminimalkan batas atas dari risiko yang diharapkan [31]. Secara praktis, hal ini memberikan SVM keunggulan teoretis dalam generalisasi dan kemampuan untuk menghindari *overfitting*, menjadikannya pilihan yang kuat untuk data berdimensi tinggi seperti fitur-fitur pada deteksi intrusi.

2.4.2. Model Ensemble

Merupakan prosedur yang melatih beberapa model *machine learning* dan menggabungkan hasil keluaran mereka untuk mendapatkan prediksi yang lebih baik dan lebih stabil [32]. Pendekatan ini bertujuan untuk meningkatkan akurasi dan kemampuan generalisasi model, dan telah terbukti secara konsisten mengungguli model ML tunggal dalam berbagai tugas deteksi serangan [15][16]. Implementasi model *Random Forest* menggunakan pustaka *scikit-learn*, sedangkan *XGBoost* menggunakan pustaka aslinya pada linkungan *python*. Selaras dengan pendekatan pada model *non-ensemble*, kedua model *ensemble* ini juga dijalankan menggunakan parameter *default* yang disediakan oleh masing-masing pustaka untuk menjamin perbandingan yang objektif.

- *Random Forest* (RF): Model ini dipilih sebagai perwakilan dari metode *Bagging* (*Bootstrap Aggregating*). RF membangun sejumlah besar *Decision Tree* secara paralel selama pelatihan. Setiap pohon dilatih pada sampel data yang diambil secara acak (*bootstrap*) dan hanya mempertimbangkan *subset* fitur secara acak untuk setiap percabangan. Prediksi akhir diambil berdasarkan voting mayoritas dari semua pohon. RF dipilih karena ketahanannya terhadap *overfitting* dan performa tingginya yang telah terbukti dalam dataset deteksi intrusi IoT sebelumnya, seperti BoT-IoT [29]. Relevansinya juga didukung oleh penelitian lain yang menggabungkan RF dengan seleksi fitur untuk keamanan IoT [28].
- *XGBoost* (*Extreme Gradient Boosting*): Model ini dipilih sebagai perwakilan dari metode *Boosting*. Berbeda dengan RF, *boosting* membangun model secara berurutan. Setiap model baru dilatih untuk memperbaiki kesalahan atau sisa dari model sebelumnya. *XGBoost* secara khusus mengoptimalkan proses *gradient boosting* ini dengan regularisasi untuk mencegah *overfitting* dan pemrosesan paralel, sehingga menghasilkan performa yang sangat tinggi. Model ini dipilih karena popularitasnya sebagai model pemenang dalam banyak kompetisi dan kinerjanya yang terbukti unggul dalam klasifikasi lalu lintas jaringan [14].

2.5. Skenario Eksperimen dan Evaluasi Model

Setiap model, baik *ensemble* maupun *non-ensemble*, dilatih dan dievaluasi menggunakan empat variasi dataset. Variasi ini merupakan kombinasi dari dua skenario seleksi fitur, yaitu dataset *filtered* dan *unfiltered*, serta dua skenario normalisasi data, yaitu *scaled* dan *unscaled*. Evaluasi performa model menggunakan pendekatan *Stratified K-Fold Cross Validation*. Teknik *Stratified K-Fold* ini dipilih karena kemampuannya menjaga distribusi label yang seimbang di setiap *fold*, sehingga mengurangi bias

Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

evaluasi pada data tidak seimbang [33]. Jumlah *fold* (K) ditetapkan sebanyak lima. Nilai ini dipilih sebagai titik tengah yang optimal untuk menyeimbangkan antara keandalan hasil evaluasi dan efisiensi sumber daya komputasi, terutama untuk mengelola waktu pelatihan model yang intensif. Performa model diukur berdasarkan nilai rata-rata dari beberapa metrik kuantitatif, yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan ROC-AUC *score*, serta perbandingan lama waktu pelatihan.

3. HASIL PENELITIAN

Bab ini menyajikan hasil eksperimen klasifikasi serangan siber pada dataset CICIoT2023. Fokus utama adalah membandingkan performa model *non-ensemble* (*Naive Bayes, Logistic Regression, Support Vector Machine*) dan model *ensemble* (*Random Forest, XGBoost*) berdasarkan empat variasi skenario *preprocessing*.

3.1. Hasil Preprocessing dan Seleksi Fitur

Proses preprocessing data diawali dengan seleksi fitur untuk mengidentifikasi fitur yang paling relevan dan mengurangi redundansi. Analisis korelasi awal menunjukkan bahwa beberapa fitur statistik, seperti *Min*, *Max*, *Avg*, dan *Tot Size*, memiliki korelasi sangat kuat dengan nilai koefisien (r) lebih dari 0.90. Untuk memilih subset fitur terbaik, penelitian ini menggunakan gabungan teknik *Mutual Information* (MI) dan *Random Forest Feature Importance* (RFFI). Dari total 39 fitur awal, proses ini mengidentifikasi 21 fitur final yang dianggap paling berkontribusi pada klasifikasi, seperti yang dirinci pada Tabel 3.

Tabel 3. Fitur Terpilih dan Distribusi Kelas Akhir

No	Fitur	No	Fitur	No	Fitur
1	Max	8	IAT	15	TCP
2	Number	9	Std	16	syn_flag_number
3	AVG	10	Variance	17	syn_count
4	Min	11	Time_To_Live	18	psh_flag_number
5	Tot sum	12	ack_flag_number	19	ack_count
6	Rate	13	HTTPS	20	Protocol Type
7	Tot size	14	Header_Length	21	UDP

Setelah dataset *filtered* yang berisi 21 fitur dan dataset *unfiltered* yang berisi 39 fitur disiapkan, langkah selanjutnya adalah penanganan data. Dataset asli CICIoT2023 yang memiliki 33 label serangan, pertama-tama melalui proses label *binning* untuk dikelompokkan menjadi lima kelas utama: DoS, *Mirai*, *Recon*, *Spoofing*, dan *Normal*. Untuk mengatasi ketidakseimbangan kelas yang signifikan [13], penelitian ini menerapkan strategi *random undersampling* pada kelas mayoritas. Proses ini memastikan setiap kelas memiliki jumlah sampel yang sama, sehingga menghasilkan dataset yang seimbang untuk pelatihan model, seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Distribusi Kelas Akhir Setelah Proses Balancing

Kelas	Jumlah Sampel
Recon	142.616
DoS	142.616
Spoofing	142.616
Normal	142.616
Mirai	142.616
Total	**713.080**

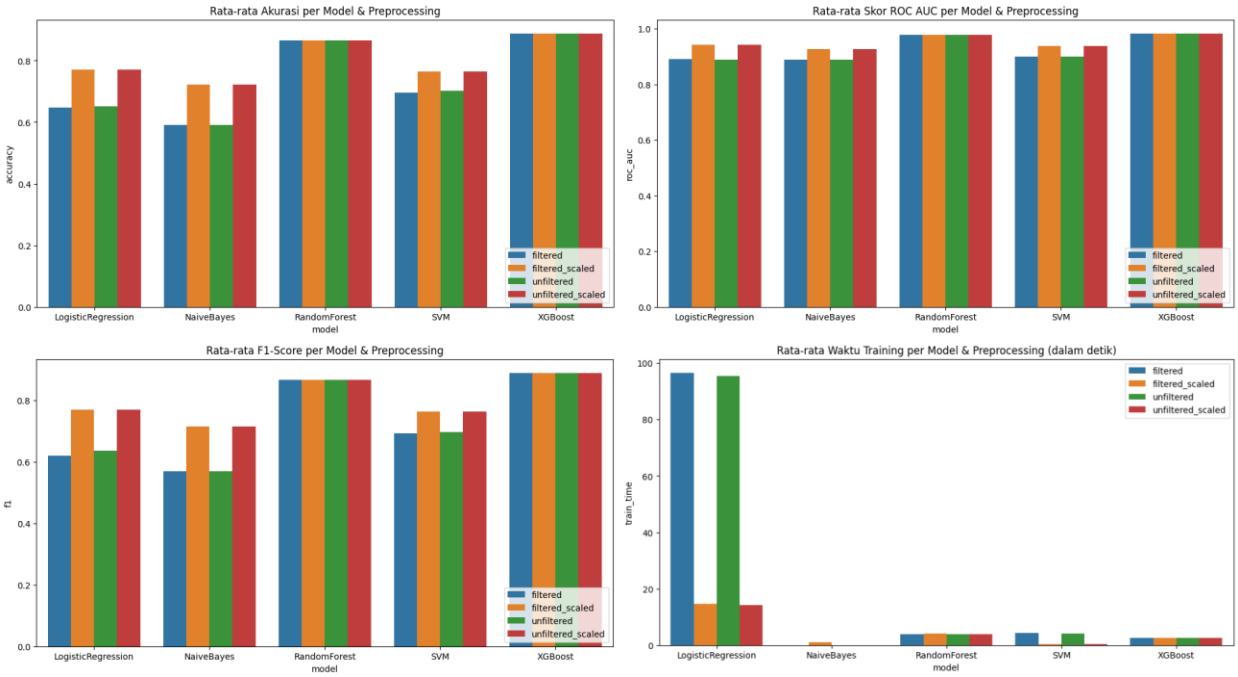
Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

3.2. Evaluasi Model Berdasarkan Validasi Silang (K-Fold)

Setiap model, baik *ensemble* maupun *non-ensemble*, dilatih dan dievaluasi menggunakan empat variasi dataset. Variasi ini merupakan kombinasi dari dua skenario seleksi fitur, yaitu dataset *filtered* dan *unfiltered*, serta dua skenario normalisasi data, yaitu *scaled* dan *unscaled*.

3.2.1. Pengaruh Preprocessing terhadap Performa Model

Seperti yang ditunjukkan pada Gambar 2, hasil evaluasi menunjukkan dua pola yang berbeda secara signifikan antara model *ensemble* dan *non-ensemble*.



Gambar 2. Hasil Evaluasi Model Setelah Proses K-Fold

Model *ensemble*, yaitu *Random Forest* (RF) dan *XGBoost*, menunjukkan performa yang sangat tinggi dan stabil di semua variasi dataset. *XGBoost* secara konsisten mencapai *F1-score* rata-rata tertinggi, yaitu 0.8889 ±0.0008. RF menyusul dengan *F1-score* rata-rata yang juga sangat stabil pada 0.8667 ±0.0007. Performa yang konsisten ini, baik pada data yang telah di-*scaling* maupun tidak, menunjukkan bahwa model berbasis pohon keputusan tidak sensitif terhadap skala fitur.

Sebaliknya, performa model *non-ensemble*, yang mencakup *Logistic Regression* (LR), *Naive Bayes* (NB), dan *Support Vector Machine* (SVM), terbukti sangat bergantung pada proses *scaling*. Model LR menunjukkan peningkatan *F1-score* paling drastis. Nilainya meningkat dari rata-rata 0.6280 pada data *unscaled* menjadi 0.7691 pada data *scaled*. Penting untuk dicatat, performa LR pada data *unscaled* juga sangat tidak stabil, yang ditunjukkan oleh standar deviasi tinggi, mencapai ±0.0388 pada varian *filtered*. Demikian pula, *F1-score* pada model NB meningkat signifikan dari 0.5682 menjadi 0.7153 setelah di-*scaling*. Model SVM juga mengalami peningkatan serupa, dari rata-rata 0.6945 pada dataset *unfiltered* menjadi 0.7624 pada dataset *scaled*, dengan performa yang jauh lebih stabil, dimana standar deviasi turun dari ±0.0101 menjadi ±0.0008. Temuan ini menegaskan bahwa normalisasi fitur adalah langkah krusial untuk algoritma yang berbasis perhitungan jarak atau optimasi gradien.

3.2.2. Efisiensi Waktu Pelatihan

Seperti yang diilustrasikan pada Gambar 2, analisis waktu pelatihan juga menunjukkan dampak signifikan dari *scaling*. Model LR membutuhkan waktu latih rata-rata yang sangat tinggi, berkisar 95.8

Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

hingga 96.9 detik, pada data *unscaled*. Namun, setelah dilakukan *scaling*, waktu latihnya turun drastis menjadi rata-rata 14.6 hingga 15.0 detik. Hal serupa terjadi pada SVM, yang waktu pelatihannya turun secara signifikan dari kisaran 4.5 hingga 4.6 detik untuk dataset *unscaled* menjadi hanya sekitar 0.69 detik pada dataset *scaled*.

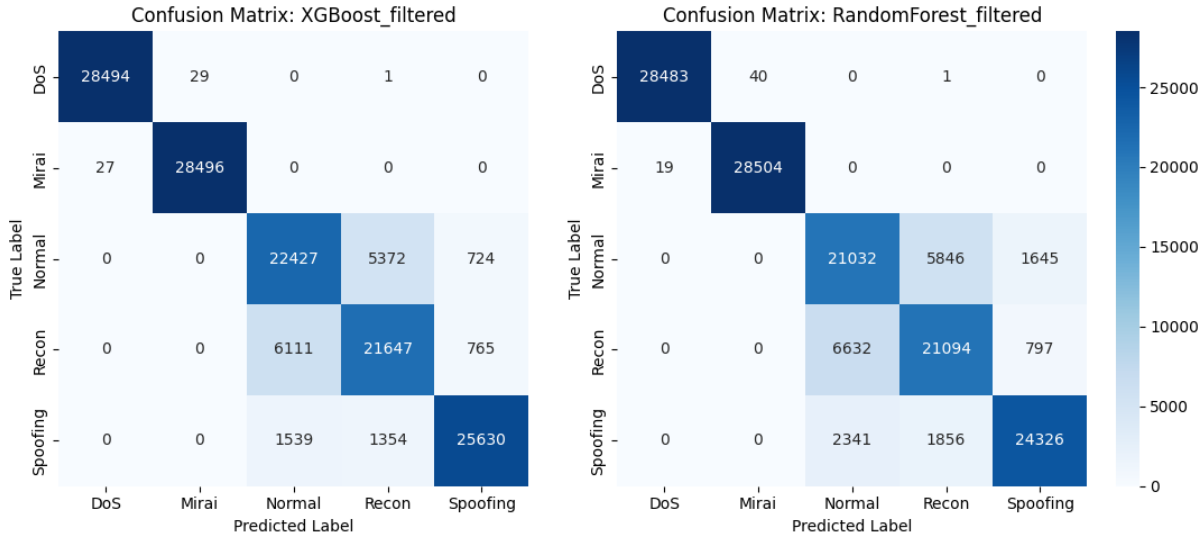
Model *ensemble*, yaitu RF dan *XGBoost*, menunjukkan waktu latih yang konsisten dan tidak terpengaruh oleh *scaling*. RF membutuhkan waktu latih rata-rata yang konsisten sekitar 6.0 hingga 6.2 detik pada semua variasi dataset. Sementara *XGBoost* secara konsisten lebih cepat, dengan waktu latih rata-rata sekitar 2.9 detik. Model NB menjadi yang paling ringan secara komputasi pada data *unscaled* dengan rata-rata waktu latih di bawah 0.5 detik, dan tetap efisien pada data *scaled* dengan rata-rata waktu latih berkisar antara 0.4 hingga 1.75 detik, meskipun dengan performa prediktif terendah.

3.3. Analisis Kinerja Model Terbaik

Berdasarkan nilai F1-score tertinggi dan waktu pelatihan yang efisien, tiga model terbaik dipilih untuk analisis final: *XGBoost* (*ensemble*), *Random Forest* (*ensemble*), dan *Logistic Regression* (*non-ensemble* terbaik).

3.3.1. XGBoost (Varian Filtered)

Model ini mencapai akurasi dan *F1-score* secara keselurhan dengan nilai masing-masing 0.8884 dan 0.8891. Seperti yang ditunjukkan oleh *classification report* pada Tabel 5, model ini menunjukkan performa sempurna dengan *F1-score* 1.00 untuk klasifikasi kelas DoS dan *Mirai*. Namun, seperti yang ditunjukkan secara kuantitatif pada Gambar 3, model ini masih menunjukkan kebingungan signifikan antara kelas-kelas yang mirip dimana 6.111 sampel *Recon* salah diklasifikasikan sebagai *Normal*, dan 5.372 sampel *Normal* salah diklasifikasikan sebagai *Recon*.



Gambar 3. Confusion Matrix XGBoost & Random Forest

3.3.2. Random Forest (Varian Filtered)

Model ini mencapai *F1-score* 0.8665 dan akurasi 0.8655. Seperti *XGBoost*, model RF juga sempurna dalam mendeteksi DoS dan *Mirai*, dengan *F1-score* 1.00. Namun, performanya sedikit lebih rendah dalam klasifikasi *Spoofing* yang menghasilkan *F1-score* sebesar 0.88, kelas *Normal* pada 0.72, dan kelas *Recon* pada 0.74. Hal ini juga terkonfirmasi secara kuantitatif pada Gambar 2, di mana model ini salah mengklasifikasikan 6.632 sampel *Recon* sebagai *Normal* dan 5.846 sampel *Normal* sebagai *Recon*.

3.3.3. Logistic Regression (Varian Filtered-Scaled)

Sebagai model *non-ensemble* terbaik, LR mencapai *F1-score* 0.7686 dan akurasi 0.7700. Performa model ini sangat timpang; meskipun sempurna untuk DoS dan *Mirai* dengan *F1-score* 1.00, model ini sangat kesulitan mengklasifikasikan kelas *Normal* yang hanya mencapai *F1-score* 0.56 dan kelas *Recon* pada 0.63. Rincian performa ini terdapat pada *classification report* di Tabel 5.

Tabel 5. Classification Report

Kelas	XGBoost – Filtered			Random Forest – Filtered			Logistic Regression – Filtered Scaled		
	Precission	Recall	F1	Precission	Recall	F1	Precission	Recall	F1
DoS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Mirai	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Normal	0.75	0.79	0.77	0.70	0.74	0.72	0.60	0.52	0.56
Recon	0.76	0.76	0.76	0.73	0.74	0.74	0.63	0.62	0.63
Spoofing	0.95	0.90	0.92	0.91	0.85	0.88	0.62	0.72	0.66

4. PEMBAHASAN

Penelitian ini menyajikan beberapa temuan penting. Pertama, model gabungan (*ensemble*) seperti *XGBoost* dan *Random Forest* secara jelas bekerja lebih baik daripada model tunggal (*non-ensemble*) seperti *Logistic Regression*, SVM, dan *Naive Bayes*. Model gabungan ini menang dalam hal nilai *F1-score* dan kinerja yang stabil saat diuji pada dataset CICIoT2023. Hasil ini mendukung penelitian sebelumnya oleh [15] dan [16] yang juga menemukan bahwa metode gabungan (*ensemble*) memang menunjukkan hasil yang lebih unggul dibandingkan algoritma ML biasa untuk tugas deteksi intrusi.

Kinerja *Random Forest* (RF) yang baik dalam studi ini, yang mencapai *F1-score* 0.87, juga memperkuat temuan [29]. Penelitian mereka sebelumnya menemukan RF adalah model paling akurat untuk dataset *BoT-IoT* yang tidak seimbang. Ini membuktikan bahwa kemampuan RF dalam menangani variasi data juga sama baiknya saat diterapkan pada dataset *CICIoT2023* yang lebih baru. Meskipun demikian, *XGBoost* terbukti sedikit lebih unggul dengan *F1-score* 0.89 berkat cara kerjanya yang fokus memperbaiki kesalahan secara bertahap.

Temuan penting kedua adalah peran besar dari *preprocessing* untuk model tunggal. Seperti yang terlihat pada Gambar 2, hasil eksperimen menunjukkan bahwa proses *scaling* data tidak hanya menaikkan nilai *F1-score* pada model *Logistic Regression* dan *SVM* secara drastis, tetapi juga memotong waktu latih mereka secara signifikan. Ini membuktikan bahwa model yang mengandalkan hitungan jarak, seperti SVM dan LR, memang harus menggunakan data yang sudah dinormalisasi agar dapat bekerja dengan baik dan cepat.

Menariknya, proses pemilihan fitur yang menggunakan MI dan RFFI ternyata tidak banyak meningkatkan kinerja model gabungan, yaitu RF dan *XGBoost*. Hal ini menandakan bahwa cara kerja internal model RF yang menggunakan sistem *voting* dari banyak pohon dan *XGBoost* yang bisa memberi bobot pada fitur, sudah cukup baik dalam menyaring sendiri fitur-fitur yang tidak penting.

Tantangan terbesar yang ditemukan dalam penelitian ini adalah semua model sulit membedakan secara akurat antara kelas *Recon* dan lalu lintas *Normal*, seperti yang dirinci pada Tabel 5. Walaupun model gabungan bisa mengenali serangan bervolume tinggi seperti DoS dan *Mirai* dengan sempurna, dengan *F1-score* mencapai 1.00, kesalahan prediksi (*false positive*) yang cukup tinggi antara kelas *Recon* dan *Normal* menandakan bahwa ciri-ciri lalu lintas data dari kedua kelas ini memang sangat mirip. Ini menjadi catatan untuk penelitian selanjutnya, yang mungkin butuh teknik fitur yang lebih canggih atau model yang lebih rumit untuk membedakan aktivitas mencurigakan yang samar ini dari lalu lintas biasa.

Temuan kami juga selaras dengan *benchmarking* pada dataset *CIC-IDS2017* pada penelitian [17], di mana model berbasis pohon seperti *Random Forest* terbukti lebih unggul daripada model linear. Demikian pula, temuan kami tentang efektivitas RF dan SVM sejalan dengan perbandingan oleh [18] pada dataset IoT lainnya. Pentingnya *preprocessing* yang cermat, seperti yang kami temukan pada model LR dan SVM, juga didukung oleh penelitian [24], yang menunjukkan bahwa pendekatan *hybrid* dalam *preprocessing* yaitu seleksi fitur dan reduksi data, sangat penting untuk meningkatkan kinerja deteksi. Perbandingan dengan [14] juga mengkonfirmasi keunggulan *XGBoost* dalam tugas klasifikasi trafik jaringan.

Secara implikatif untuk ilmu komputer, temuan ini memiliki dua dampak utama. Pertama, efisiensi komputasi *XGBoost* dengan waktu latih 2.68 detik dan RF dengan waktu latih 3.88 detik dibandingkan dengan LR *unscaled* yang lebih dari 90 detik, menunjukkan bahwa model *ensemble* tidak hanya lebih akurat tetapi juga lebih efisien secara praktis, yang sangat penting untuk melatih ulang model IDS secara berkala. Kedua, skalabilitas penerapan model ini pada perangkat *edge computing* IoT menjadi pertimbangan penting. Meskipun *XGBoost* dan RF sangat akurat, model yang lebih ringan seperti *Logistic Regression* setelah di-*scaling* dengan benar mungkin menawarkan keseimbangan yang lebih baik antara performa deteksi dan penggunaan sumber daya yang terbatas pada perangkat *edge*.

5. KESIMPULAN

Penelitian ini menyimpulkan bahwa model *ensemble learning*, khususnya *XGBoost* dan *Random Forest,* secara konsisten menunjukkan performa yang jauh lebih unggul dalam hal akurasi, *F1-score*, dan stabilitas dibandingkan model *non-ensemble* seperti *Logistic Regression*, SVM, dan *Naive Bayes* untuk tugas klasifikasi serangan *multiclass* pada dataset *CICIoT2023*. *XGBoost* terbukti menjadi model paling efektif secara keseluruhan dalam menangani dataset yang kompleks ini.

Temuan ini juga menegaskan peran ganda dari *preprocessing* data. Di satu sisi, normalisasi data terbukti menjadi langkah krusial bagi model *non-ensemble* seperti *Logistic Regression* dan SVM. Proses ini tidak hanya meningkatkan akurasi mereka secara drastis, tetapi juga mempercepat waktu pelatihan secara signifikan. Di sisi lain, model *ensemble* terbukti tidak terlalu bergantung pada proses seleksi fitur, yang menandakan kemampuan internal mereka untuk menangani fitur yang tidak relevan.

Meskipun model-model terbaik mampu mendeteksi serangan bervolume tinggi seperti DoS dan *Mirai* dengan nyaris sempurna, tantangan utama tetap ada. Seluruh model menunjukkan kesulitan dalam membedakan secara akurat antara lalu lintas *Normal* dan serangan subtil seperti *Recon*. Temuan ini memberikan dampak spesifik bagi bidang informatika, menyoroti kebutuhan akan *framework* IDS yang tidak hanya akurat tetapi juga efisien secara komputasi untuk skenario IoT. Oleh karena itu, penelitian di masa depan disarankan untuk berfokus pada beberapa area pengembangan. Pertama, eksplorasi *model* hybrid yang mengintegrasikan *Deep Learning* (DL) dengan *ensemble* ML untuk meningkatkan kemampuan pembedaan kelas yang mirip. Kedua, penelitian lanjutan dapat berfokus pada implementasi dan pengujian model-model ini untuk skenario deteksi *real-time*, dengan mempertimbangkan potensi pengembangan *framework open-source* untuk IDS IoT guna memberikan kontribusi praktis yang lebih luas.

ACKNOWLEDGEMENT

Penelitian ini didanai oleh Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Sebelas Maret melalui pendanaan Non APBN UNS dengan Nomor Kontrak 369/UN27.22/PT.01.03/2025.

REFERENCES

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, “Internet

Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, 2013, doi: 10.1016/j.future.2013.01.010.

[2] S. W. Mudjanarko, S. Winardi, and A. D. Limantara, "Pemanfaatan Internet of Things (IoT) sebagai solusi manajemen transportasi kendaraan sepeda motor," Prosiding Seminar Nasional Aplikasi Teknologi Prasarana Wilayah X, 2017.

[3] E. D. Meutia, "Internet of Things – Keamanan dan Privasi," in Seminar Nasional dan Expo Teknik Elektro, vol. 1, no. 1, pp. 85–89, 2015.

[4] Joseph Jose Anthraper, and J. Kotak, "Security, Privacy and Forensic Concern of MQTT Protocol," Social Science Research Network, 2019, doi: 10.2139/ssrn.3355193.

[5] Maad M. Mijwil, Omega John Unogwu, Y. Filali, I. Bala, and Humam Al-Shahwani, "Exploring the Top Five Evolving Threats in Cybersecurity: An In-Depth Overview," Mesopotamian Journal of Cyber Security, 2023, doi: 10.58496/mjcs/2023/010.

[6] Warsun Najib, Selo Sulistyo, and Widyawan, "Tinjauan Ancaman dan Solusi Keamanan pada Teknologi Internet of Things," Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI), vol. 9, no. 4, 2020, doi: 10.22146/jnteti.v9i4.539.

[7] Jiamin Hu, and Xiaofan Yang, "A cost-effective adaptive repair strategy to mitigate DDoS-capable IoT botnets," PLoS ONE, vol. 19, no. 12, 2024, doi: 10.1371/journal.pone.0301888.

[8] Patrick Vanin et al., "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," Applied Sciences, vol. 12, 2022, doi: 10.3390/app122211752.

[9] Hongyu Liu, and Bo Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," Applied Sciences, 2019, doi: 10.3390/app9204396.

[10] Zeeshan Ahmad, A. Khan, W. Cheah, J. Abdullah, and Farhan Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, 2020, doi: 10.1002/ett.4150.

[11] Tamara A. Al-Shurbaji et al., "Deep Learning-Based Intrusion Detection System For Detecting IoT Botnet Attacks: A Review," IEEE Access, 2025, doi: 10.1109/access.2025.3526711.

[12] Yung-Chung Wang, Yi-Chun Houng, Han-Xuan Chen, and S. Tseng, "Network Anomaly Intrusion Detection Based on Deep Learning Approach," Italian National Conference on Sensors, 2023, doi: 10.3390/s23042171.

[13] Fatima Hussain, Rasheed Hussain, Syed Ali Hassan, and Ekram Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," arXiv (Cornell University), 2019, doi: 10.48550/arxiv.1904.05735.

[14] Won-Ju Eom, Yeong-Jun Song, Chang-Hoon Park, Jeong-Keun Kim, Geon-Hwan Kim, and You-Ze Cho, "Network Traffic Classification Using Ensemble Learning in Software-Defined Networks," 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2021, doi: 10.1109/icaiic51459.2021.9415187.

[15] Edi Ismanto, Januar Al Amien, and Vitriani Vitriani, "A Comparison of Enhanced Ensemble Learning Techniques for Internet of Things Network Attack Detection," Matrik Jurnal Manajemen Teknik Informatika dan Rekayasa Komputer, 2024, doi: 10.30812/matrik.v23i3.3885.

[16] Wengang Zhang, Chongzhi Wu, Haiyi Zhong, Yongqin Li, and Lin Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," Geoscience frontiers, 2021, doi: 10.1016/j.gsf.2020.03.007.

[17] Ziadoon Kamil Maseer, R. Yusof, N. Bahaman, S. Mostafa, and Cik Feresa Mohd Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," IEEE Access, vol. 9, 2021, doi: 10.1109/access.2021.3056614.

[18] Yakub Kayode Saheed, Aremu Idris Abiodun, S. Misra, Monica Kristiansen Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," Alexandria Engineering Journal, 2022, doi: 10.1016/j.aej.2022.02.063.

[19] Raja Azlina Raja Mahmood, AmirHossien Abdi, and Masnida Hussin, "Performance Evaluation of Intrusion Detection System using Selected Features and Machine Learning Classifiers," Baghdad Science Journal, 2021, doi: 10.21123/bsj.2021.18.2(suppl.).0884.

[20] Jared M. Peterson, Joffrey L. Leevy, and T. Khoshgoftaar, "A Review and Analysis of the Bot-

Vol. 7, No. 1, February 2026, Page. 606-618
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5408

IoT Dataset," International Symposium on Service Oriented Software Engineering, 2021, doi: 10.1109/sose52839.2021.00007.

[21] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data Preprocessing for Supervised Leaning,"International Journal of Computer Science, 2007.

[22] Sikha Bagui, and Kunqi Li, "Resampling imbalanced data for network intrusion detection datasets," Journal of Big Data, 2021, doi: 10.1186/s40537-020-00390-x.

[23] Fei Zhao, Jiyong Zhao, Xinxin Niu, Shoushan Luo, and Yang Xin, "A Filter Feature Selection Algorithm Based on Mutual Information for Intrusion Detection," Applied Sciences, 2018, doi: 10.3390/app8091535.

[24] Achmad Akbar Megantara, and Tohari Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," Journal of Big Data, vol. 8, 2021, doi: 10.1186/s40537-021-00531-w.

[25] E. P. Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Italian National Conference on Sensors, 2023, doi: 10.3390/s23135941.

[26] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," arXiv: Artificial Intelligence, 2011, doi: 10.1613/jair.953.

[27] Hanchuan Peng, Fuhui Long, and Chris Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, doi: 10.1109/tpami.2005.159.

[28] Aulia Arif Wardana, Parman Sukarno, Setio Basuki, and Sasmito Budi Utomo, "Federated Random Forest with Feature Selection for Collaborative Intrusion Detection in Internet of Things," Procedia Computer Science, 2024, doi: 10.1016/j.procs.2024.09.193.

[29] Andrew Churcher et al., "An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks.," arXiv: Cryptography and Security, 2021, doi: 10.3390/s21020446.

[30] Jadel Alsamiri, and Khalid Alsubhi, "Internet of Things Cyber Attacks Detection using Machine Learning," International Journal of Advanced Computer Science and Applications, 2019, doi: 10.14569/ijacsa.2019.0101280.

[31] Y. LeCun, Y. Bengio, and M. A. Arbib, The Handbook of Brain Theory and Neural Networks. 1998.

[32] Gavin Brown, "Ensemble Learning," Encyclopedia of Machine Learning, 2011, doi: 10.1007/978-0-387-30164-8_252.

[33] Chen Chen et al., "Application of GA-WELM Model Based on Stratified Cross-Validation in Intrusion Detection," Symmetry, 2023, doi: 10.3390/sym15091719.

[34] Andrew P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognition, 1997, doi: 10.1016/s0031-3203(96)00142-2.