 ISSN:2302-8149
e-ISSN:2540-9719

Perbandingan Algoritma *Machine Learning* untuk *Intrusion Detection System* pada Dataset NSL-KDD

A Comparative Study of Machine Learning Algorithms for Intrusion Detection Systems using the NSL-KDD Dataset

[1]Rulyansyah Permata Putra, [2]Amarudin*
[1]Program Studi Teknologi Informasi, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia
[2]Program Studi Ilmu Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia
[1,2]Jl. Zainal Abidin Pagaralam No.9-11 Labuhan Ratu, Bandar Lampung, Indonesia
*e-mail: rulyansyah_permata_putra@teknokrat.ac.id, amarudin@teknokrat.ac.id

(***received*:** 25 April 2025, ***revised*:** 24 May 2025, ***accepted*:** 25 May 2025)

Abstrak

Di era digital saat ini, serangan siber berkembang semakin kompleks, sehingga sistem deteksi intrusi (IDS) berbasis aturan statis sering kali gagal mengenali pola serangan baru. Tujuan utama penelitian adalah merancang dan mengimplementasikan model *machine learning* untuk mendeteksi intrusi jaringan komputer secara efektif dengan meminimalkan latensi, melalui analisis komparatif antara beberapa algoritma, yaitu *Decision Tree, Random Forest, Support Vector Machine* (SVM), dan *Boosting*. Metode penelitian yang digunakan meliputi pengumpulan dataset NSL-KDD, proses transformasi data, pembersihan, normalisasi, dan pembagian data menjadi set pelatihan dan pengujian. Masing-masing algoritma dilatih dengan parameter yang telah disesuaikan, dan performanya dievaluasi menggunakan metrik akurasi, presisi, recall, f1-score, serta analisis waktu pelatihan dan prediksi. Hasil penelitian menunjukkan bahwa algoritma *Boosting* menonjol dengan tingkat akurasi mencapai 99,36%. Boosting juga terbukti lebih handal dalam mendeteksi kelas minoritas meski memerlukan waktu pelatihan lebih lama. Penerapan metode *machine learning*, khususnya *Boosting*, merupakan pendekatan efektif untuk meningkatkan deteksi intrusi yang dapat dijadikan dasar bagi pengembangan sistem keamanan siber lebih adaptif dan andal.

Kata kunci: *intrusion detection system*, *boosting*, *machine learning*, NSL-KDD, keamanan siber

Abstract

In today's digital era, cyberattacks are becoming increasingly complex, rendering traditional rule-based Intrusion Detection Systems (IDS) often ineffective in recognizing new attack patterns. The primary objective of this study is to design and implement a machine learning model for detecting network intrusions efficiently while minimizing latency, through a comparative analysis of several algorithms: Decision Tree, Random Forest, Support Vector Machine (SVM), and Boosting. The research methodology includes the collection of the NSL-KDD dataset, followed by data transformation, cleaning, normalization, and partitioning into training and testing sets. Each algorithm was trained using tuned parameters, and performance was evaluated using metrics such as accuracy, precision, recall, F1-score, and an analysis of training and prediction time. The results indicate that the Boosting algorithm stands out, achieving an accuracy rate of 99.36%. Boosting also demonstrated greater reliability in detecting minority classes, despite requiring longer training times. The application of machine learning methods—particularly Boosting—proves to be an effective approach to enhancing intrusion detection and can serve as a foundation for developing more adaptive and reliable cybersecurity systems.

Keywords: *intrusion detection system, boosting, machine learning, NSL-KDD, cyber security*

 ISSN:2302-8149
e-ISSN:2540-9719

1 Pendahuluan

Keamanan sistem informasi menjadi aspek krusial dalam dunia digital yang terus berkembang pesat. Ancaman siber, seperti serangan peretasan, pencurian data, dan penyusupan jaringan, semakin kompleks dan sulit dideteksi. Seiring dengan meningkatnya jumlah data yang mengalir dalam suatu sistem, metode konvensional dalam mendeteksi serangan siber sering kali kurang efektif dalam mengenali pola serangan yang baru dan canggih. Oleh karena itu, diperlukan solusi yang lebih adaptif dan cerdas untuk meningkatkan efektivitas sistem keamanan jaringan.

Intrusion Detection System (IDS) merupakan salah satu solusi yang umum digunakan untuk mendeteksi aktivitas mencurigakan dalam suatu jaringan. IDS bekerja dengan menganalisis lalu lintas jaringan untuk mengidentifikasi pola yang mengindikasikan potensi serangan. Namun, sistem IDS berbasis aturan statis sering kali mengalami keterbatasan dalam mendeteksi serangan baru yang belum terdefinisi dalam database aturan yang ada. Oleh karena itu, integrasi teknologi kecerdasan buatan, khususnya *machine learning*, menjadi strategi yang menjanjikan dalam meningkatkan kemampuan IDS untuk mengenali ancaman yang terus berkembang [1].

Pendekatan ini memiliki potensi untuk meningkatkan keamanan secara keseluruhan dengan mengurangi jumlah *false positive* dan mendeteksi serangan yang sebelumnya tidak teridentifikasi. Desain sistem deteksi intrusi (IDS) yang memanfaatkan pembelajaran mesin dalam keamanan siber jaringan melibatkan beberapa elemen kunci. Pertama, diperlukan dataset yang kuat dan menyeluruh untuk melatih serta mengevaluasi model pembelajaran mesin. Dataset yang diusulkan harus mencakup berbagai pola lalu lintas jaringan, baik yang normal maupun yang berbahaya, agar proses pembelajaran dapat berlangsung dengan efektif. Kedua, pemilihan fitur dan teknik ekstraksi yang tepat sangat penting untuk menangkap informasi relevan dari data jaringan [2].

Dengan menerapkan *machine learning* memungkinkan sistem untuk menganalisis pola lalu lintas jaringan dan mendeteksi potensi intrusi secara otomatis [3]. Fokus utama dari penelitian ini adalah mengidentifikasi algoritma *machine learning* yang paling efektif dalam mendeteksi berbagai jenis serangan siber. Algoritma yang digunakan dalam penelitian ini meliputi *Decision Tree, Random Forest, Support Vector Machine (SVM),* dan *Boosting*. Selain itu, penelitian ini juga akan mengkaji performa model berdasarkan metrik evaluasi seperti akurasi, presisi, recall, dan waktu pemrosesan. Dengan demikian, pendekatan ini menjadi solusi yang menjanjikan dalam menghadapi tantangan keamanan jaringan modern [4]. Adapun yang menjadi pokok permasalahan yang diungkap pada penelitina ini antara lain:

a. Adanya pola serangan baru dan kompleks pada jaringan komputer yang tidak mudah dideteksi dengan aturan statis.
b. Belum diketahuinya performa algoritma Boosting dalam meningkatkan akurasi dan sensitivitas deteksi intrusi pada dataset NSL-KDD, dibandingkan dengan jenis algoritma *machine learning* yang lain seperti Decision Tree, Random Forest, dan SVM.
c. Belum diketahuinya *trade-off* antara peningkatan akurasi deteksi dan kebutuhan komputasi (waktu pelatihan dan prediksi) pada penerapan metode Boosting dalam IDS.

Berdasarkan beberapa masalah penelitian yang telah dipaparkan di atas, maka dalam penelitian ini disusun beberapa tujuan penelitian sebagai berikut:

a. Merancang dan mengimplementasikan model *machine learning* berbasis Boosting untuk sistem deteksi intrusi jaringan.
b. Melakukan analisis komparatif performa Boosting dengan Decision Tree, Random Forest, dan SVM menggunakan metrik akurasi, presisi, recall, f1-score, serta waktu pelatihan dan prediksi.
c. Mengevaluasi kemampuan Boosting dalam mendeteksi serangan, khususnya pada kelas minoritas yang sulit dikenali, sekaligus menilai efisiensi komputasi untuk aplikasi IDS real-time.

2 Tinjauan Literatur

Intrusion Detection System (IDS) merupakan salah satu komponen utama dalam keamanan siber yang berfungsi untuk mendeteksi dan merespons ancaman terhadap suatu sistem atau jaringan. IDS dapat diklasifikasikan ke dalam dua kategori utama, yaitu *Host-Based Intrusion Detection System (*HIDS) dan *Network-Based Intrusion Detection System* (NIDS). HIDS bekerja dengan menganalisis

 ISSN:2302-8149
e-ISSN:2540-9719

log sistem pada perangkat individu untuk mendeteksi aktivitas mencurigakan, sementara NIDS mengawasi lalu lintas jaringan untuk mengidentifikasi pola anomali. Keduanya memiliki keunggulan masing-masing, namun tantangan seperti tingkat false positive yang tinggi dan keterbatasan dalam menangani serangan yang lebih kompleks masih menjadi kendala dalam implementasinya [5].

Seiring dengan berkembangnya teknologi dan meningkatnya kompleksitas serangan siber, metode deteksi berbasis tanda tangan (*signature-based*) dalam sistem deteksi intrusi (IDS) menghadapi keterbatasan dalam mengenali ancaman baru atau serangan yang telah dimodifikasi. Oleh karena itu, pendekatan berbasis anomali dengan *machine learning* semakin banyak diterapkan untuk meningkatkan efektivitas IDS. Teknik *machine learning* dan *deep learning* telah terbukti mampu mengenali pola serangan anomali yang menyimpang dari perilaku jaringan normal, sehingga mempercepat proses deteksi dan meningkatkan akurasi sistem [6].

Amarudin, et al., [7] memperkenalkan model B-DT yang mengombinasikan teknik bagging dengan *Decision Tree* untuk sistem deteksi intrusi, dan melaporkan bahwa model tersebut berhasil menurunkan jumlah *false detection* pada dataset NSL-KDD dari 11.305 menjadi hanya 243, serta mencapai akurasi 99,45% dan kappa-score 99,08%.

Lama, et al., [8] melakukan studi komprehensif pada dataset CSE-CIC-IDS2018 dengan tahapan data *cleaning, feature engineering*, dan seleksi fitur, lalu membandingkan *Logistic Regression, Random Forest*, dan *Gradient Boosting*. Dalam penelitian tersebut melaporkan bahwa *Gradient Boosting* mencapai *precision, recall*, dan *f1-score* sebesar 0,98 pada data uji yang belum pernah dilihat sebelumnya, hal ini menunjukkan keandalannya dalam deteksi intrusi jaringan secara biner.

Salah satu pendekatan yang semakin populer dalam implementasi *machine learning* untuk IDS adalah metode *ensemble learning*, khususnya *Boosting*. *Boosting* merupakan teknik yang menggabungkan beberapa model lemah (*weak learners*) untuk membentuk model yang lebih kuat dan akurat [9]. Dalam konteks IDS, algoritma *Boosting* seperti *AdaBoost, Gradient Boosting*, dan *XGBoost* telah diterapkan untuk meningkatkan kinerja deteksi intrusi. Metode ini bekerja dengan cara melatih model secara iteratif, di mana setiap model baru berfokus pada memperbaiki kesalahan yang dibuat oleh model sebelumnya, sehingga menghasilkan prediksi yang lebih akurat. *Real AdaBoost* menunjukkan performa tinggi tetapi rentan terhadap *overfitting*, sementara *Gentle AdaBoost* lebih stabil dalam menghadapi data dengan *noise* tinggi [10]. *XGBoost* dikembangkan untuk mengatasi kelemahan *Real AdaBoost* dalam menangani dataset yang tidak seimbang [11].

Algoritma *machine learning* yang digunakan memiliki kelebihan dan kekurangannya masing-masing sebagai berikut:

a. *Decision Tree* (DT): Berupa pohon keputusan yang menawarkan berbagai keuntungan seperti efisiensi, kemudahan penggunaan, dan kompleksitas yang rendah. Akan tetapi, DT juga memiliki kekurangan, termasuk kerentanan terhadap *overfitting* dan ketidakstabilan dengan perubahan data kecil, yang dapat menyebabkan struktur dan klasifikasi pohon yang berbeda [12].
b. *Random Forest* (RF): Terdiri dari banyak pohon keputusan yang bekerja secara bersamaan. Dengan menggabungkan hasil prediksi dari berbagai pohon, algoritma ini mampu mengurangi risiko *overfitting* dan menghasilkan prediksi yang lebih akurat [13]. Namun, kinerjanya bisa sangat bervariasi tergantung pada pemilihan hiperparameternya, sehingga perlu dilakukan penyetelan yang cermat untuk mendapatkan hasil yang optimal [14].
c. *Support Vector Machine* (SVM) : Algoritma ini paling banyak digunakan dengan akurasi keseluruhan terbaik dan mampu menangani data berdimensi tinggi, namun kinerjanya sangat bergantung pada optimisasi parameter. Selain itu, SVM juga memerlukan komputasi yang tinggi, terutama dengan kumpulan data yang besar, yang dapat membatasi skalabilitasnya [15].
d. *Boosting*: Algoritma ini mampu menangani ketidakseimbangan data serta meningkatkan akurasi klasifikasi tanpa menyebabkan *overfitting*. Selain itu, juga mampu meningkatkan efisiensi deteksi terhadap serangan yang lebih kompleks. Dengan *Boosting* memungkinkan *Intrusion Detection System* (IDS) dapat terus menyempurnakan kemampuan deteksi, merespons pola serangan baru secara efektif melalui pembelajaran adaptif berbasis analisis lanskap ancaman yang dinamis [16]. Namun, Boosting memiliki satu kendala utama yaitu kebutuhan daya komputasi yang tinggi, terutama saat digunakan dalam sistem *real-time*. *Boosting* juga dalam

 ISSN:2302-8149
e-ISSN:2540-9719

mendeteksi serangan *zero-day* masih perlu ditingkatkan, karena model hanya dapat belajar dari pola yang telah diketahui dalam data pelatihan [10].

Secara keseluruhan, penggunaan *machine learning*, khususnya metode *Boosting*, telah membawa kemajuan signifikan dalam deteksi intrusi. Dengan kemampuan adaptif dan akurasinya yang tinggi, metode ini menjadi solusi yang menjanjikan untuk mengatasi tantangan yang dihadapi oleh IDS tradisional. Namun, penelitian dan pengembangan lebih lanjut tetap diperlukan untuk meningkatkan efisiensi, keamanan, dan interpretabilitas model dalam menghadapi ancaman siber yang terus berkembang.

3 Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan tujuan untuk membandingkan kinerja algoritma *machine learning* dalam mendeteksi serangan siber. Setiap algoritma menggunakan dataset yang sama, sehingga perbandingan hasil dapat dilakukan secara adil. Fokus utama adalah mengevaluasi efektivitas metode *Boosting* dibandingkan algoritma lain.

3.1. Pengumpulan Data

Pengumpulan data pada penelitian ini menggunakan dataset NSL-KDD. Dataset ini terdiri dari 41 fitur yang mencakup informasi seperti durasi koneksi, jenis protocol, dan jumlah paket yang dikirim. Gambar 1 menampilkan deskripsi fitur data dan jenis dari dataset yang digunakan. Dataset ini terbagi menjadi dua bagian utama, yaitu data pelatihan dan data pengujian, yang mencakup berbagai jenis serangan (seperti DoS, Probe, U2R, dan R2L) serta data normal.

```
Data columns (total 43 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   duration                     148515 non-null  int64
 1   protocol_type                148515 non-null  object
 2   service                      148515 non-null  object
 3   flag                         148515 non-null  object
 4   src_bytes                    148515 non-null  int64
 5   dst_bytes                    148515 non-null  int64
 6   land                         148515 non-null  int64
 7   wrong_fragment               148515 non-null  int64
 8   urgent                       148515 non-null  int64
 9   hot                          148515 non-null  int64
 10  num_failed_logins            148515 non-null  int64
 11  logged_in                    148515 non-null  int64
 12  num_compromised              148515 non-null  int64
 13  root_shell                   148515 non-null  int64
 14  su_attempted                 148515 non-null  int64
 15  num_root                     148515 non-null  int64
 16  num_file_creations           148515 non-null  int64
 17  num_shells                   148515 non-null  int64
 18  num_access_files             148515 non-null  int64
 19  num_outbound_cmds            148515 non-null  int64
...
 41  attack                       148515 non-null  object
 42  level                        148515 non-null  int64
dtypes: float64(15), int64(24), object(4)
```

Gambar 1. Fitur dataset NSL-KDD

3.2. Pemprosesan Data

Data yang telah diperoleh kemudian dilakukan pemprosesan data sehingga data dapat digunakan. Langkah Langkah yang dilakukan selama pemprosesan sebagai berikut:

a. Transformasi Data: Fitur non-numerik dalam dataset dikonversi ke dalam format numerik menggunakan metode seperti *One-Hot Encoding* atau *Label Encoding* agar dapat dipahami oleh algoritma *machine learning*.

 ISSN:2302-8149
e-ISSN:2540-9719

b. Pembersihan Data: Menghapus data yang duplikat serta menangani nilai yang hilang, baik dengan teknik imputasi maupun penghapusan data yang tidak relevan.
c. Normalisasi Data: Untuk memastikan bahwa perbedaan skala antar fitur tidak memengaruhi performa model, dilakukan normalisasi menggunakan *Min-Max Scaling* atau standarisasi dengan *Standard Scaling*.
d. Pembagian Data: Dataset dibagi menjadi dua bagian, yaitu 70% data pelatihan yang digunakan untuk melatih model dan 30% data pengujian yang digunakan untuk mengevaluasi performa model.
e. Distribusi Jumlah *Class* sebagai data pelatihan dan data pengujian: Pembagian data ke dalam kelas-kelas untuk pelatihan dan pengujian sangat penting agar setiap kelas tetap terwakili. Seperti ditunjukkan pada Table 1, distribusi jumlah sampel untuk masing-masing kelas pada data pelatihan (70%) dan data pengujian (30%) dipertahankan secara stratifikasi untuk memastikan representasi kelas yang seimbang.

Table 1. Distribusi jumlah *class* data pelatihan dan data pengujian

No	Nama Kelas	Pelatihan (70%)	Pengujian (30%)
1	DoS	37.354	16.009
2	Normal	53.937	23.116
3	Probing	9.848	4.221
4	R2L	2.750	1.179
5	U2R	63	26
Total Data		103.952	44.551

3.3. Pemilihan Model *Machine Learning*

Pemilihan algoritma ini didasarkan pada kemampuannya dalam menangani masalah deteksi intrusi dengan mempertimbangkan karakteristik dari dataset NSL-KDD. Meskipun penelitian ini lebih menekankan pada penerapan metode *Boosting*, model-model lainnya juga digunakan sebagai perbandingan untuk mengevaluasi kelebihan dan kekurangan masing-masing pendekatan, termasuk dalam hal kemudahan penyesuaian parameter, dan efisiensi komputasi. Algoritma yang digunakan dalam penelitian ini ada empat: yaitu

a. *Decision Tree* (DT): DT merupakan algoritma yang dapat memilih fitur dan paling berpengaruh dalam membagi data. Selanjutnya data dibagi berdasarkan nilai fitur tersebut. Proses ini diulangi secara bertahap hingga mencapai kondisi penghentian, dan menggunakan jalur dari akar ke daun untuk melakukan prediksi [12]. Pada algoritma Decision Tree ini menggunakan parameter *random_state*=0 yang berfungsi untuk menjamin replikasi hasil pelatihan model dengan pola pemilihan data yang konsisten. Parameter ini digunakan karena merupakan default dari algoritma *Decision Tree*.
b. *Random Forest* (RF): RF merupakan algoritma yang menggunakan metode *ensemble* dengan menggabungkan banyak pohon keputusan. Setiap pohon dibangun dengan memilih subset acak dari dataset. Setiap pohon keputusan dilatih pada subset data yang berbeda. Hasil prediksi dari semua pohon digabungkan dengan mengklasifikasikannya [4]. Pada algoritma *Random Forest* ini menggunakan parameter *max_depth*=10 dan *random_state*=50. Parameter *max_depth* membatasi kedalaman maksimum pohon-pohon keputusan yang dibangun, sehingga dapat mengurangi kompleksitas model dan risiko *overfitting*, serta mempercepat proses pelatihan. Sementara *random_state* memastikan bahwa proses pemilihan subset data dan fitur dilakukan secara deterministik. Parameter ini digunakan karena menghasilkan persentasi performa tertinggi nya.

 ISSN:2302-8149
e-ISSN:2540-9719

c. *Support Vector Machine* (SVM): SVM menganalisis data dengan membangun batas keputusan yang memisahkan kelas-kelas yang berbeda dalam data dan kernel melakukan komputasi dalam ruang input, memungkinkan pemisahan data yang tidak linier [4]. Pada SVM ini menggunakan kernel RBF (*Radial Basis Function*) yang memungkinkan model untuk melakukan pemisahan *non-linier* dengan memetakan data ke ruang berdimensi lebih tinggi. Parameter *random_state*=0 disertakan untuk menjaga konsistensi dalam proses pelatihan, khususnya pada tahap optimisasi internal yang melibatkan elemen acak. Meningkatkan parameter tidak memberikan peningkatan pada keamanan.
d. *Boosting*: *Boosting* merupakan salah satu teknik dalam *ensemble learning* yang bekerja dengan cara menggabungkan beberapa model lemah (*weak learners*) untuk menghasilkan sebuah model yang lebih kuat dan akurat [10]. Adapun pada algoritma **Boosting**, yang menjadi fokus utama dalam penelitian ini, menggunakan beberapa parameter penting, yaitu *n_estimators*=50, *learning_rate*=1.0, *max_depth*=15, dan *random_state*=0. Parameter *n_estimators* menunjukkan jumlah model lemah (*weak learners*) yang akan dilatih secara bertahap, di mana setiap model baru berfokus pada memperbaiki kesalahan prediksi model sebelumnya. Parameter *learning_rate* sebesar 1.0 menunjukkan bahwa setiap model memberikan kontribusi penuh terhadap prediksi akhir. Adapun *max_depth* yang diset pada *Boosting* ini sebesar 15. Sehingga, memungkinkan setiap pohon dalam *Boosting* dapat menangkap kompleksitas data dengan lebih baik, meskipun berisiko menyebabkan *overfitting* jika tidak diimbangi dengan teknik regularisasi yang tepat. Sama seperti algoritma lain, *random_state*=0 digunakan untuk menjaga replikasi hasil. Adapun nilai parameter tersebut diambil berdasarkan dari beberapa percobaan yang telah dilakukan. Mulai dari merubah nilai *n_estimators* dan *max_depth.* Dimana dari hasil percobaan tersebut didapatkan nilai maksimal pada *n_estimators=50,* dan *max_depth*=15.

3.4. Implementasi Metode

Data diimplementasikan dengan cara melatih dan menguji model menggunakan dataset yang telah disiapkan sebelumnya. Data pelatihan digunakan untuk membangun model dengan masing-masing algoritma, dengan penyesuaian parameter. Setelah model terlatih, dilakukan pengujian dengan data pengujian untuk mengukur performa secara menyeluruh.

3.5. Evaluasi

Evaluasi performa dilakukan dengan menghitung metrik seperti akurasi, presisi, recall, dan f1-score, serta dengan analisis *confusion matrix*. Di samping itu, pengukuran waktu proses untuk pelatihan dan prediksi juga dilakukan untuk menilai efisiensi tiap algoritma.

4 Hasil dan Pembahasan

Evaluasi dilakukan terhadap empat algoritma *machine learning* menggunakan dataset NSL-KDD, diperoleh hasil evaluasi performa dari empat algoritma yang dibandingkan, yaitu Decision Tree, Random Forest, Support Vector Machine (SVM), dan Boosting. Metrik evaluasi yang digunakan dalam penelitian ini mencakup akurasi, presisi, recall, dan f1-score. Table 2 menunjukkan hasil evaluasi model berdasarkan algoritma yang digunakan.

Table 2. Hasil pengukuran performa model

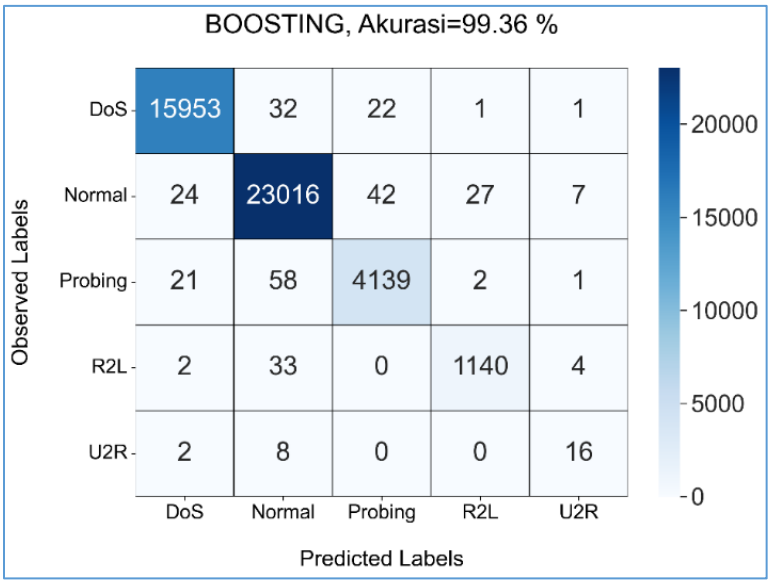
Algoritma	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)	Waktu Latih (s)	Waktu Prediksi (s)
Decision Tree	99.20	99.21	99.20	99.20	0.9	0.012
Random Forest	98.81	98.81	98.81	98.79	13	0.66
SVM	98.12	98.12	98.12	98.11	68	35
Boosting	**99.36**	**99.36**	**99.36**	**99.36**	**171**	**0.65**

 ISSN:2302-8149
e-ISSN:2540-9719

Table 2 menunjukkan bahwa semua algoritma menghasilkan akurasi di atas 98%, yang berarti bahwa algoritma tersebut cukup baik dalam mengenali pola serangan dan lalu lintas normal menggunakan dataset NSL-KDD. Namun, terdapat perbedaan dari setiap agoritma yang perlu untuk dicermati.

4.1. *Boosting*

Implementasi algoritma *Boosting* pada dataset NSL-KDD mencatat performa terbaik pada semua metrik evaluasi, dengan akurasi 99.36%, f1-score yang seimbang, dan kemampuan mengklasifikasikan serangan secara konsisten. Hal ini menunjukkan bahwa teknik *Boosting* efektif dalam mendeteksi serangan dan membedakan lalu lintas normal secara konsisten, namun *Boosting* membutuh waktu pelatihan yang cukup lama. Adapun *confusion matrik Boosting* hasil dari penelitian ini disajikan pada Gambar 2.



Gambar 2. *Confusion matrix boosting*

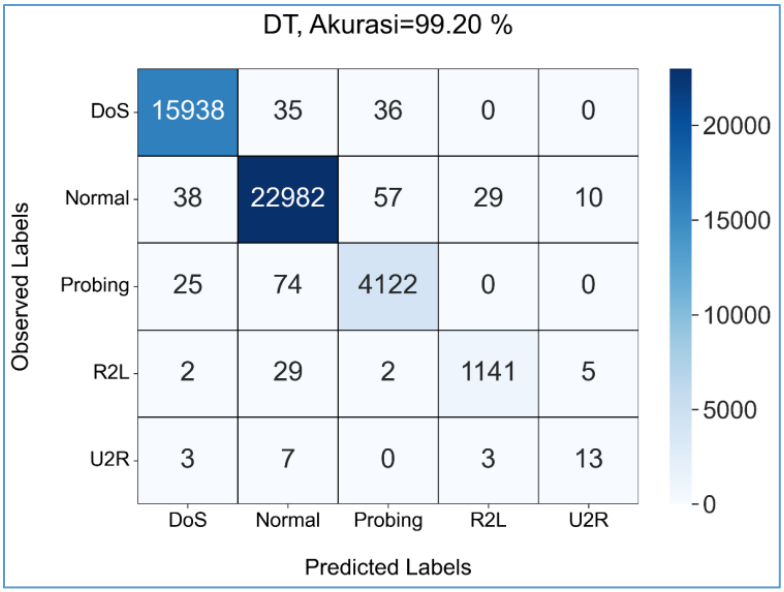
Confusion matrix menunjukkan bahwa model *Boosting* mampu mengklasifikasikan data dengan akurat, termasuk kelas minoritas yang biasanya sulit dikenali oleh model lain. Ini menunjukkan bahwa *Boosting* tidak hanya memiliki performa yang baik secara keseluruhan, tetapi juga mampu mengenali pola-pola kompleks dalam data serangan.

Secara keseluruhan, *Boosting* mampu menjadi solusi unggul untuk sistem deteksi intrusi karena ketepatan dan konsistensinya yang tinggi. Namun, jika waktu pelatihan menjadi kendala utama, perlu dipertimbangkan optimasi atau pendekatan lain untuk mempercepat proses tanpa mengorbankan akurasi secara signifikan.

4.2. *Decision Tree*

Algoritma *Decision Tree* (DT) menunjukkan performa klasifikasi yang sangat baik pada dataset NSL-KDD dengan tingkat akurasi mencapai 99.20%. dan salah satu model tercepat dalam pengujian ini. Berdasarkan *confusion matrix* yang ditampilkan, terlihat bahwa *Decision Tree* mampu mengklasifikasikan data dengan tingkat ketepatan yang tinggi, khususnya pada kelas mayoritas seperti *DoS* (15.938 data diklasifikasikan benar dari total 16.009) dan *Normal* (22.982 benar dari total 23.116), menunjukkan keakuratan deteksi terhadap serangan yang lebih umum. Adapun *confusion matrix Decision Tree* dari hasil penelitian ini disajikan pada Gambar 3.

 ISSN:2302-8149
e-ISSN:2540-9719

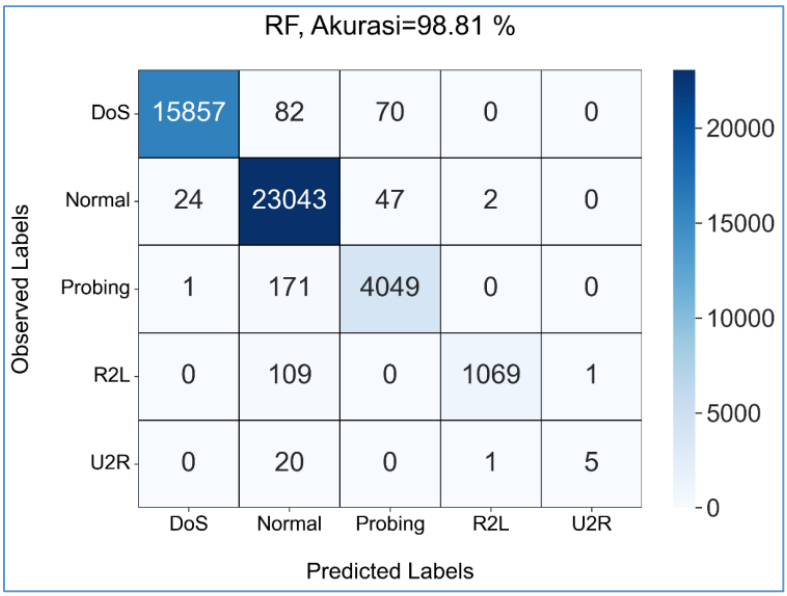


Gambar 3. *Confusion matrix decision tree*

Berdasarkan Gambar 3, dapat diketahui bahwa model ini juga menunjukkan performa cukup baik pada kelas *Probing*, dengan 4.122 dari 4.521 data berhasil diklasifikasikan dengan benar. Meskipun terdapat sedikit kesalahan klasifikasi pada kelas *R2L* dan *U2R*, yang merupakan kelas minoritas dan lebih sulit dideteksi, model ini tetap dapat mengidentifikasi sebagian besar serangan dengan benar. Hal ini mencerminkan bahwa *Decision Tree* tetap mampu menangani data yang tidak seimbang dengan cukup baik. Dengan kecepatan proses yang tinggi dan kompleksitas yang relatif rendah dibandingkan model lain, algoritma ini sangat cocok untuk implementasi sistem keamanan jaringan *real-time* yang membutuhkan respons cepat dan akurat.

4.3. *Random Forest*

Berdasarkan tabel evaluasi, *Random Forest* menghasilkan akurasi sebesar 98.81% dengan nilai presisi dan recall yang hampir identik, yang menandakan bahwa model ini secara keseluruhan sangat efektif dalam mengklasifikasikan data intrusi. Waktu pelatihan yang dibutuhkan tercatat sekitar 13 detik, dan waktu prediksi hanya 0.66 detik, menunjukkan bahwa *Random Forest* tidak hanya unggul dalam hal akurasi, tetapi juga efisien secara komputasi, yang sangat penting untuk aplikasi *real-time* dalam sistem deteksi intrusi. Adapun *confusion matrix Random Forest* dari hasil penelitian ini disajikan pada Gambar 4.



Gambar 4. *Confusion matrix random forest*

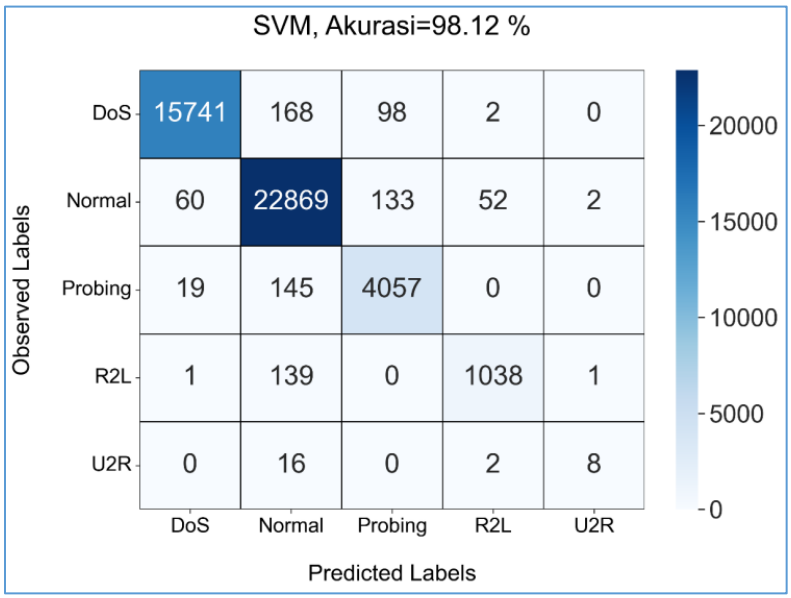
 ISSN:2302-8149
e-ISSN:2540-9719

Berdasarkan Gambar 4, dapat diketahui bahwa model *Random Forest* mampu mengidentifikasi pola-pola serangan dengan tepat, terutama pada kelas-kelas mayoritas seperti DoS dan Normal, yang mendominasi dataset NSL-KDD. Namun, seperti yang terlihat pada analisis *confusion matrix*, terdapat beberapa tantangan dalam menangani kelas minoritas seperti R2L dan U2R. Kesalahan klasifikasi pada kelas minoritas ini menunjukkan bahwa meskipun nilai metrik keseluruhan sangat tinggi, masih ada ruang untuk optimasi, misalnya dengan penerapan teknik *oversampling* atau penyesuaian bobot kelas agar model dapat mengenali serangan yang lebih jarang terjadi dengan lebih baik.

Secara keseluruhan, performa *Random Forest* yang stabil dengan nilai metrik yang tinggi serta efisiensi waktu pelatihan dan prediksi yang baik, menjadikannya pilihan yang menarik untuk diimplementasikan dalam *Intrusion Detection System* (IDS). Namun, agar kinerjanya semakin optimal, penyesuaian lebih lanjut pada model, khususnya dalam mengatasi ketidakseimbangan data, perlu dipertimbangkan agar deteksi terhadap semua jenis serangan dapat dilakukan secara merata dan akurat.

4.4. *Support Vector Machine* (SVM)

SVM menunjukkan akurasi paling rendah 98.12% diantara algoritma lainnya. Meskipun performa klasifikasinya masih tinggi, kelemahan utamanya terletak pada sensitivitas terhadap kelas minoritas dan waktu komputasi yang tinggi, sehingga kurang ideal untuk penggunaan *real-time* atau ketika sumber daya terbatas. Adapun *confusion matrix SVM* dari hasil penelitian ini disajikan pada Gambar 5.



Gambar 5. ***Confusion matrix support vector machine***

Berdasarkan Gambar 5, dapat diketahui bahwa model *Support Vector Machine* (SVM) menunjukkan kinerja klasifikasi yang cukup baik. Dengan demikian, SVM berhasil mengklasifikasikan sebagian besar data intrusi dan lalu lintas normal dengan tepat. Namun, perlu dicermati bahwa waktu pelatihan yang dibutuhkan mencapai 68 detik dan waktu prediksi mencapai 35 detik, yang menandakan bahwa meskipun performa klasifikasi yang dicapai cukup tinggi, model ini memiliki kebutuhan komputasi yang signifikan.

Di sisi lain, perbedaan waktu komputasi yang cukup besar dibandingkan dengan algoritma lain (misalnya *Decision Tree* dan *Random Forest*) menunjukkan potensi keterlambatan dalam pengambilan keputusan, sehingga hal tersebut dapat menjadi kendala dalam implementasi *real-time* pada sistem deteksi intrusi. Untuk meningkatkan performa SVM, diperlukan optimasi lebih lanjut melalui *tuning* parameter atau penerapan teknik peningkatan sensitivitas terhadap kelas minoritas agar dapat mengurangi kesalahan klasifikasi, terutama pada jenis serangan yang jarang muncul namun krusial dalam konteks keamanan jaringan.

 ISSN:2302-8149
e-ISSN:2540-9719

4.5 Analisis Hasil

Penelitian ini menunjukkan bahwa penerapan algoritma *machine learning*, yaitu *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)*, dan *Boosting*, dapat memberikan performa tinggi dalam mendeteksi intrusi pada jaringan komputer berbasis dataset NSL-KDD. Semua algoritma menunjukkan akurasi di atas 98%, yang mengindikasikan bahwa pendekatan ini sangat efektif untuk mengenali pola serangan siber maupun lalu lintas jaringan yang normal. Di antara algoritma tersebut, *Boosting* unggul secara akurasi, sementara *Decision Tree* dan *Random Forest* menonjol dalam efisiensi waktu, menjadikannya lebih cocok untuk implementasi sistem *real-time*.

Perbandingan performa antar algoritma memperlihatkan adanya *trade-off* antara akurasi dan efisiensi komputasi. *Boosting* menunjukkan keunggulan akurasi dan sensitivitas terhadap kelas minoritas, namun membutuhkan waktu pelatihan yang lebih lama. Sebaliknya, *Decision Tree* dan *Random Forest* menawarkan kecepatan pelatihan dan prediksi yang tinggi dengan akurasi yang kompetitif. Sementara itu, SVM memberikan hasil cukup baik namun memiliki keterbatasan pada waktu komputasi, sehingga perlu dioptimalkan lebih lanjut. Perbedaan ini memberikan dasar penting bagi pemilihan algoritma berdasarkan kebutuhan spesifik dari sistem deteksi intrusi yang dikembangkan.

5 Kesimpulan

Penelitian ini berhasil melakukan analisis komparatif terhadap Decision Tree, Random Forest, Support Vector Machine (SVM), dan Boosting menggunakan dataset NSL-KDD. Hasil menunjukkan bahwa metode Boosting unggul dengan akurasi 99,36% serta f1-score yang seimbang, terutama dalam mendeteksi kelas minoritas. Meskipun memerlukan waktu pelatihan lebih lama, Boosting menawarkan sensitivitas dan konsistensi tinggi yang dapat meningkatkan efektivitas dan keandalan Intrusion Detection System (IDS) dalam menghadapi serangan kompleks.

Kontribusi penelitian ini terhadap bidang keamanan siber terletak pada pembuktian bahwa *ensemble learning* (khususnya Boosting) dapat mengatasi keterbatasan model berbasis aturan statis, memperbaiki deteksi data tidak seimbang, serta mengurangi risiko *false negative*. Temuan ini menyediakan referensi praktis bagi pengembang IDS untuk memilih algoritma yang sesuai, menyeimbangkan antara akurasi deteksi dan efisiensi komputasi. Di sisi implementasi, pengoptimalan parameter *learning-rate* dan *n_estimators*, serta pemanfaatan komputasi terdistribusi, dapat mempercepat proses pelatihan agar model ini lebih layak digunakan dalam aplikasi *real-time*.

Saran penelitian selanjutnya meliputi pengujian pada dataset yang lebih mutakhir seperti CSE‑CIC‑IDS2018 untuk melihat keandalan model dalam kondisi ancaman dan volume data terbaru, serta implementasi dan uji coba nyata (*real-time testing*) dengan integrasi pada platform keamanan jaringan untuk menilai dampak pada skalabilitas dan latensi. Selain itu, eksplorasi pendekatan *hybrid* menggabungkan Boosting dengan *deep learning* atau *graph‑based feature extraction* diharapkan dapat meningkatkan akurasi dan interpretabilitas model, sehingga memberikan IDS yang lebih adaptif dan responsif terhadap lanskap ancaman siber yang terus berkembang.

Referensi

[1] B. Dhanunjay, E. Sanjay, A. K. Raj, and M. Dholvan, "*Intrusion Detection System using Machine Learning*," *Quantum Comput. Model. Cybersecurity Wirel. Commun.*, Vol. 6, No. 3, pp. 279–292, 2020, doi: 10.1002/9781394271429.ch19.

[2] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "*Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity*," *Appl. Sci.*, Vol. 13, No. 13, 2023, doi: 10.3390/app13137507.

[3] Amarudin, Styawati, Syaifuddin, and M. Iqbal, "*Improving Intrusion Detection System on Servers using Machine Learning-Based RFERF Technique*," *Proceeding 2024 9th Int. Conf. Inf. Technol. Digit. Appl. ICITDA 2024*, pp. 1–8, 2024, doi: 10.1109/ICITDA64560.2024.10809789.

[4] K. N. Jaya Varshini R, Sifa Thahasin F, Jayasri S, "*Intrusion Detection System using Machine Learning Algorithm*," *Quantum Comput. Model. Cybersecurity Wirel. Commun.*, pp. 279–292, 2023, doi: 10.1002/9781394271429.ch19.

[5] A. Efe and İ. N. Abaci, "*Comparison of the Host based Intrusion Detection Systems and*

 ISSN:2302-8149
e-ISSN:2540-9719

Network Based Intrusion Detection Systems," *Celal Bayar Üniversitesi Fen Bilim. Derg.*, Vol. 18, No. 1, pp. 23–32, 2022, doi: 10.18466/cbayarfbe.832533.

[6] A. Elhanashi, K. Gasmi, A. Begni, P. Dini, Q. Zheng, and S. Saponara, "*Machine Learning Techniques for Anomaly-based Detection System on CSE-CIC-IDS2018 Dataset*," *Lect. Notes Electr. Eng.*, vol. 1036 LNEE, No. April, pp. 131–140, 2023, doi: 10.1007/978-3-031-30333-3_17.

[7] Amarudin, R. Ferdiana, and Widyawan, "B-DT Model: *A Derivative Ensemble Method to Improve Performance of Intrusion Detection System*," *J. Adv. Inf. Technol.*, Vol. 15, No. 1, pp. 87–103, 2024, doi: 10.12720/jait.15.1.87-103.

[8] A. Lama and D. P. Savant, "*Network-based Intrusion Detection Systems using Machine Learning Algorithms*," *Int. J. Eng. Appl. Sci. Technol.*, Vol. 6, No. 11, pp. 145–155, 2022, doi: 10.33564/ijeast.2022.v06i11.028.

[9] O. D. Okey *et al.*, "BoostedEnML: *Efficient Technique for Detecting Cyberattacks in IoT Systems using Boosted Ensemble Machine Learning*," *Sensors*, Vol. 22, No. 19, pp. 1–26, 2022, doi: 10.3390/s22197409.

[10] A. Shahraki, M. Abbasi, and Ø. Haugen, "*Boosting Algorithms for Network Intrusion Detection: A Comparative Evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost*," *Eng. Appl. Artif. Intell.*, Vol. 94, No. February, p. 103770, 2020, doi: 10.1016/j.engappai.2020.103770.

[11] M. Hernowo and E. Sugiharti, "*XGBoost Algorithm on Intrusion Detection System in Detecting Network Intrusions*," *Endang Sugiharti Innov. J. Soc. Sci. Res.*, Vol. 4, pp. 10572–10588, 2024.

[12] A. M. Sani, A. S. Ben-musa, and M. Haladu, "I*n-Depth Study of Decision Tree Model*," *Int. J. SCI. Res.*, Vol. 10, No. 11, pp. 705–709, 2021, doi: 10.21275/MR211102051237.

[13] D. P. Sari, Z. Halim, Irlon, B. Waseso, and Saromah, "Implementasi *Machine Learning* untuk Deteksi Intrusi pada Jaringan Komputer," Vol. 13, No. September, pp. 1389–1394, 2024.

[14] T. Zhu, "*Analysis on the Applicability of the Random Forest*," *J. Phys. Conf. Ser.*, Vol. 1607, No. 1, 2020, doi: 10.1088/1742-6596/1607/1/012123.

[15] M. Aljanabi, M. A. Ismail, and A. H. Ali, "*Intrusion Detection Systems, Issues, Challenges, and Needs*," *Int. J. Comput. Intell. Syst.*, Vol. 14, No. 1, pp. 560–571, 2021, doi: 10.2991/ijcis.d.210105.001.

[16] M. Farooq, M. H. Khan, and R. A. Khan, "*Dynamic Threat Landscape Analysis and Adaptive Response Strategies for Intrusion Detection and Prevention Systems using Advance Gradient Boosting Algorithms*," *Ijarcce*, Vol. 13, No. 3, pp. 251–264, 2024, doi: 10.17148/ijarcce.2024.13243.