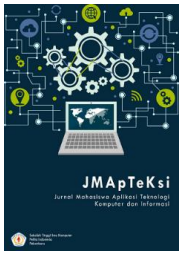


Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi

Vol 7 No 2 Thn 2025 Hal 128-134
E ISSN : 2685-6565

Sistem Deteksi Intrusi Pada Jaringan Komputer Menggunakan Algoritma XGBoost

Wesley Alexander [a], *Deny Jollyta* [b]

[a]*Institut Bisnis dan Teknologi Pelita Indonesia, Jl.AhmadYaniNo,78-88,wesley.alexander@student.pelitaindonesia.ac.id*
[b]*Institut Bisnis dan Teknologi Pelita Indonesia, Jl.AhmadYaniNo,78-88, deny.jollyta@lecturer.pelitaindonesia.ac.id*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 24 April 2025
Revisi Akhir: 15 Desember 2025
Diterbitkan *Online*: 15 Desember 2025

KATA KUNCI

XGBoost, IDS, NSL-KDD, DoS, Keamanan_jaringan.

KORESPONDENSI

E-mail:
wesley.alexander@student.pelitaindonesia.ac.id

A B S T R A C T

Penelitian ini bertujuan untuk mengembangkan sistem deteksi intrusi pada jaringan komputer menggunakan algoritma XGBoost. Intrusi dalam jaringan dapat menyebabkan kerugian besar, sehingga diperlukan sistem yang mampu mendeteksi serangan secara cepat dan akurat. Dataset yang digunakan adalah NSL-KDD, yang telah diproses melalui tahapan preprocessing dan pembagian data untuk pelatihan dan pengujian model. Hasil pengujian menunjukkan bahwa algoritma XGBoost mampu mendeteksi intrusi dengan akurasi sebesar 99,1%, precision 98,9%, recall 99,2%, dan F1-score 99%, yang menunjukkan kinerja sangat baik dalam mendeteksi serangan jaringan. Sistem ini diharapkan dapat menjadi solusi yang efektif dalam meningkatkan keamanan jaringan komputer.

1. PENDAHULUAN

Dalam era digital yang semakin terhubung, ancaman siber semakin kompleks dan seringkali sulit dideteksi. Serangan siber dapat menyebabkan kerugian finansial yang signifikan, gangguan layanan, dan kerusakan reputasi bagi organisasi. Oleh karena itu, sistem deteksi intrusi (IDS) yang efektif menjadi sangat penting.

Salah satu pendekatan yang menjanjikan dalam pengembangan IDS adalah penggunaan teknik pembelajaran mesin. Penelitian sebelumnya [1],[2],[3] telah menunjukkan potensi besar dari algoritma pembelajaran mesin dalam mendeteksi berbagai jenis serangan siber. XGBoost, sebagai salah satu algoritma ensemble yang populer, telah menarik perhatian para peneliti karena kemampuannya dalam menangani dataset besar, melakukan feature engineering secara otomatis, dan menghasilkan model yang akurat [4].

Namun, masih terdapat beberapa tantangan dalam penerapan XGBoost pada deteksi intrusi, seperti pemilihan fitur yang optimal [5], penyeimbangan data yang tidak seimbang [6] [7] dan evaluasi kinerja pada berbagai jenis serangan.

Penelitian ini bertujuan untuk mengevaluasi kinerja algoritma XGBoost dalam mendeteksi serangan Denial-of-Service (DoS) pada dataset NSL-KDD. Eksperimen akan dilakukan dengan berbagai konfigurasi hyperparameter untuk mengidentifikasi kombinasi yang optimal. Selain itu, perbandingan kinerja XGBoost dengan algoritma pembelajaran mesin lainnya akan dilakukan. Penelitian ini diharapkan dapat memberikan pemahaman yang lebih mendalam tentang kekuatan dan kelemahan XGBoost dalam deteksi intrusi serta memberikan rekomendasi praktis untuk penerapannya dalam sistem keamanan jaringan.

2. TINJAUAN PUSTAKA

2.1. Landasan Teori

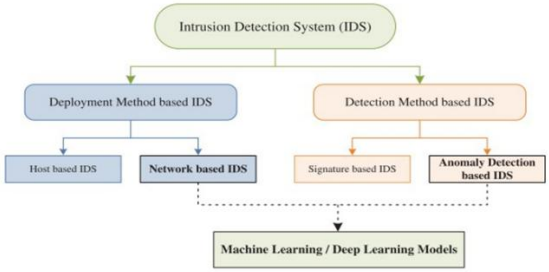
Bab ini menjelaskan teori-teori dasar yang digunakan sebagai acuan penelitian. Teori yang dibahas meliputi pengertian tentang Network Intrusion Detection System (NIDS), Machine Learning, Imbalanced Dataset, Teknik Resampling, Algoritma XGBoost, serta beberapa penelitian terdahulu.

2.1.1. Network Intrusion Detection System (NIDS)

Sistem deteksi intrusi jaringan adalah serangkaian alat atau sistem yang dirancang untuk memantau aktivitas jaringan dan sistem, mengidentifikasi potensi ancaman keamanan atau pelanggaran kebijakan. NIDS berfungsi dengan menganalisis

data jaringan untuk mendeteksi pola atau perilaku yang mencurigakan [1]. Sistem ini dibagi menjadi dua jenis utama:

1) Anomaly-based Intrusion Detection: Mendeteksi penyimpangan dari pola perilaku jaringan yang umum sebagai indikasi ancaman atau aktivitas tidak biasa.
2) Signature-based Intrusion Detection: Menggunakan basis data pola atau tanda serangan yang telah diketahui untuk mendeteksi ancaman secara langsung.



Gambar 2. 1 Klasifikasi intrusion detection system

Gambar 2.1 menunjukkan klasifikasi NIDS berdasarkan metode deteksi [1]. Pada anomaly-based intrusion detection, model machine learning dan deep learning dapat digunakan untuk mengidentifikasi pola-pola baru.

2.1.2. Machine Learning

Machine Learning adalah cabang ilmu komputer yang memungkinkan sistem belajar dari data tanpa diprogram secara eksplisit. Berdasarkan metode pembelajarannya, algoritma machine learning dibagi menjadi tiga jenis utama [8]:

1) Supervised Learning: Melatih model menggunakan data berlabel. Berikut beberapa algoritma populer dalam supervised learning
   a) **Support Vector Machine (SVM):** SVM bekerja dengan mencari hyperplane optimal yang memisahkan kelas dalam data. Algoritma ini sangat efektif untuk dataset berdimensi tinggi dan cocok untuk masalah klasifikasi serta regresi [9].
   b) **Random Forest:** Algoritma ini adalah ensemble dari pohon keputusan yang bekerja dengan menggabungkan prediksi dari beberapa pohon untuk meningkatkan akurasi dan mengurangi risiko overfitting. Random Forest cocok untuk berbagai jenis data, baik untuk klasifikasi maupun regresi [10].
   c) **Random Forest:** Algoritma ini adalah ensemble dari pohon keputusan yang bekerja dengan menggabungkan prediksi dari beberapa pohon untuk meningkatkan akurasi dan mengurangi risiko overfitting. Random Forest cocok untuk berbagai jenis data, baik untuk klasifikasi maupun regresi [11]
2) Unsupervised Learning: Melatih model menggunakan data tidak berlabel.
3) Reinforcement Learning: Melatih model untuk membuat keputusan berdasarkan umpan balik dari lingkungan.

2.1.3. Imbalanced Dataset pada Deteksi Intrusi Jaringan

Ketidakseimbangan dataset merupakan tantangan utama dalam deteksi intrusi jaringan, di mana distribusi kelas tertentu jauh lebih besar dibandingkan lainnya. Hal ini dapat mengakibatkan model lebih cenderung memprediksi kelas mayoritas [12].

Beberapa dataset yang umum digunakan dalam penelitian deteksi intrusi jaringan meliputi NSL-KDD, CIC-IDS2017, UNSW-NB15, dan KDD Cup 1999 [7]. Dalam penelitian ini, digunakan dataset NSL-KDD, yang dirancang untuk mengatasi kelemahan dataset KDD Cup 1999.

2.1.4. Teknik Resampling

Teknik Resampling adalah metode untuk menangani dataset tidak seimbang dengan menyeimbangkan representasi kelas mayoritas dan minoritas. Strategi yang digunakan meliputi:

1) Oversampling: Menduplikasi data pada kelas minoritas.
2) Undersampling: Mengurangi jumlah data pada kelas mayoritas.
3) Hybrid Sampling: Kombinasi kedua metode tersebut [6].

2.1.5. Algoritma XGBoost

XGBoost (eXtreme Gradient Boosting) adalah algoritma machine learning berbasis gradient boosting yang dirancang untuk tugas regresi dan klasifikasi. Algoritma ini membangun model secara iteratif untuk memperbaiki kesalahan model sebelumnya. Fitur utama XGBoost meliputi regularisasi, pemangkasan pohon yang efisien, dukungan pemrosesan paralel, serta penanganan missing values [13].

Penggunaan XGBoost dalam deteksi intrusi jaringan telah banyak diteliti. Beberapa studi menunjukkan efektivitasnya:

1) [4]: Menggunakan XGBoost pada dataset NSL-KDD dan mencapai akurasi 99% dalam mendeteksi serangan SYN flood.
2) [5]: Membandingkan XGBoost dengan LightGBM dan CatBoost pada dataset CIC-IDS2017, dengan hasil bahwa XGBoost unggul dalam akurasi dan kecepatan pelatihan.
3) [2]: Menerapkan XGBoost untuk mendeteksi serangan U2R dan R2L pada dataset NSL-KDD dengan akurasi 95%.

2.1.6. NSL-KDD

Dataset NSL-KDD merupakan versi perbaikan dari dataset KDD Cup 1999, dengan menghilangkan data redundan untuk meningkatkan keandalan analisis. Dataset ini digunakan untuk melatih dan menguji algoritma machine learning dalam mendeteksi intrusi jaringan.

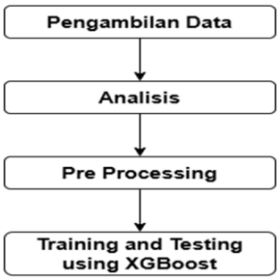
NSL-KDD menyediakan empat kategori serangan utama:

1) DoS (Denial-of-Service): Serangan untuk menggagalkan layanan, seperti SYN Flood dan Ping of Death.

2) Probing: Upaya mencari kelemahan sistem, seperti Nmap dan Satan.
3) R2L (Remote to Local): Akses tidak sah dari jarak jauh, seperti Ftpwrite dan Imap.
4) U2R (User to Root): Upaya mendapatkan hak akses superuser, seperti Loadmodule dan Xterm.

3. METODOLOGI

3.1. Kerangka Penelitian



Gambar 3. 1 Kerangka Penelitian

1) Dataset: Menyiapkan dataset NSL-KDD yang berisi informasi lalu lintas jaringan, baik yang normal maupun yang mengandung serangan.
2) Analisis: Melakukan analisis data untuk membangun model klasifikasi yang mampu membedakan antara lalu lintas jaringan yang normal dan yang mengandung serangan.
3) Pre-processing: Membersihkan dan mengubah data menjadi format yang sesuai untuk pelatihan model, termasuk penanganan missing values, encoding fitur kategorikal, dan normalisasi fitur numerik.
4) Training and Testing Using XGBoost: Membagi dataset menjadi data latih dan data uji, kemudian melatih model XGBoost menggunakan data latih dan mengevaluasi kinerjanya menggunakan data uji.

3.2. Data dan Sumber Data

Penelitian ini menggunakan dataset NSL-KDD (Network Security Laboratory - Knowledge Discovery in Databases) yang merupakan versi perbaikan dari dataset KDD Cup 1999. Dataset ini mengatasi beberapa masalah dalam dataset sebelumnya, seperti data duplikat dan distribusi data yang tidak seimbang, sehingga lebih cocok untuk evaluasi kinerja model deteksi intrusi. Tahapan yang dilalui dalam penelitian, pembangunan konsep, atau penyelesaian kasus, dituliskan pada bagian metodologi.

Tabel 3. 1 Fitur-fitur dataset NSL-KDD:

No	Nama Fitur	Deskripsi	Tipe Data
1	duration	Durasi koneksi dalam detik	Numerik
2	protocol_type	Tipe protokol yang digunakan (TCP, UDP, ICMP)	Kategori
3	service	Layanan jaringan yang digunakan	Kategori
4	flag	Status koneksi	Kategori
5	src_bytes	Jumlah byte data dari sumber	Numerik
6	dst_bytes	Jumlah byte data ke tujuan	Numerik

Distribusi Kelas: Dataset memiliki distribusi kelas yang tidak seimbang, di mana jumlah sampel serangan jauh lebih sedikit daripada sampel normal. Hal ini menjadi tantangan yang perlu ditangani dengan teknik seperti resampling atau algoritma yang dapat menangani ketidakseimbangan kelas.

3.3. Pra-pemrosesan Data

Langkah-langkah pra-pemrosesan data meliputi:

1) Penanganan Missing Values: Missing values digantikan dengan nilai modus untuk fitur kategorikal dan nilai rata-rata satau median untuk fitur numerik.
2) Encoding Fitur Kategorikal: Fitur kategorikal diubah menjadi bentuk numerik menggunakan one-hot encoding.
3) Normalisasi Fitur Numerik: Semua fitur numerik dinormalisasi ke rentang 0-1 agar memiliki bobot yang seimbang.
4) Pembagian Dataset: Dataset dibagi menjadi rasio data latih dan uji, misalnya 80:20, 70:30, atau 60:40.

3.4. Pelatihan dan Evaluasi Model

1) Pelatihan Model: Model XGBoost dilatih menggunakan data latih yang telah dipraproses. Algoritma ini bekerja dengan membuat pohon keputusan secara iteratif untuk meminimalkan kesalahan prediksi.
2) Evaluasi Model: Model diuji menggunakan data uji, dan kinerjanya diukur menggunakan beberapa metrik berikut:

Metrik Evaluasi:

a) Akurasi: Mengukur persentase prediksi yang benar.
b) Precision: Mengukur proporsi prediksi positif yang benar.
c) Recall: Mengukur kemampuan model dalam menemukan semua sampel positif.
d) F1-Score: Rata-rata harmonik dari precision dan recall.

3) Matriks Confusion: Memberikan gambaran visual tentang jenis kesalahan yang dibuat oleh model, termasuk false positives dan false negatives.
4) Kurva ROC (Receiver Operating Characteristic): Menunjukkan kemampuan model untuk membedakan antara kelas positif dan negatif pada berbagai ambang batas keputusan.

3.5. Implementasi Algoritma XGBoost

Algoritma XGBoost digunakan untuk klasifikasi dalam mendeteksi serangan pada IDS. Kelebihan XGBoost meliputi:

1) Efisiensi tinggi: Dapat menangani dataset besar dengan jumlah fitur yang kompleks.
2) Regularisasi: Membantu mencegah overfitting.
3) Paralelisasi: Mendukung pelatihan paralel pada CPU atau GPU untuk meningkatkan kecepatan.

Parameter Penting:

a) *n_estimators:* Jumlah pohon keputusan.
b) *max_depth:* Kedalaman maksimum setiap pohon.
c) *learning_rate:* Ukuran langkah yang diambil saat memperbarui bobot.
d) *subsample:* Proporsi data yang digunakan untuk melatih setiap pohon.
e) *colsample_bytree:* Proporsi fitur yang digunakan untuk setiap pohon.

XGBoost diimplementasikan menggunakan library Python, seperti scikit-learn atau XGBoost API. Model dilatih dengan optimasi parameter untuk mendapatkan kinerja terbaik.

3.6. Visualisasi Hasil

Hasil analisis dan evaluasi model akan divisualisasikan menggunakan grafik seperti:

1) Matriks Confusion: Menunjukkan klasifikasi benar dan salah.
2) Kurva ROC: Memberikan gambaran kemampuan diskriminasi model.

Dengan pendekatan metodologi ini, penelitian diharapkan dapat menghasilkan model deteksi intrusi yang efektif dan akurat untuk diterapkan pada jaringan komputer.

4. HASIL DAN PEMBAHASAN

4.1. Pengambilan Data

Penelitian ini menggunakan dataset NSL-KDD yang diperoleh dari situs resmi nsl.cs.unb.ca/NSLKDD/. Dataset ini terdiri dari data pelatihan (KDDTrain+) dan data pengujian (KDDTest+). Dataset ini dipilih karena merupakan versi perbaikan dari dataset KDD Cup 1999, yang mengatasi masalah bias dan redundansi data pada dataset sebelumnya. Dataset ini mencakup 41 fitur dan satu label kelas yang mengklasifikasikan koneksi jaringan sebagai serangan atau normal.

	0	tcp	ftp_data	SF	491	0.1	0.2	0.3	0.4	0.5	...	0.17	0.03	0.17.1	0.00.6	0.00.7	0.00.8	0.05	0.00.9
0	0	udp	other	SF	146	0	0	0	0	0	...	0.000	0.600	0.880	0.000	0.000	0.000	0.000	0.000
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.100	0.050	0.000	0.000	1.000	1.000	0.000	0.000
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.000	0.000	0.030	0.040	0.030	0.010	0.000	0.010
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.070	0.070	0.000	0.000	0.000	0.000	1.000	1.000

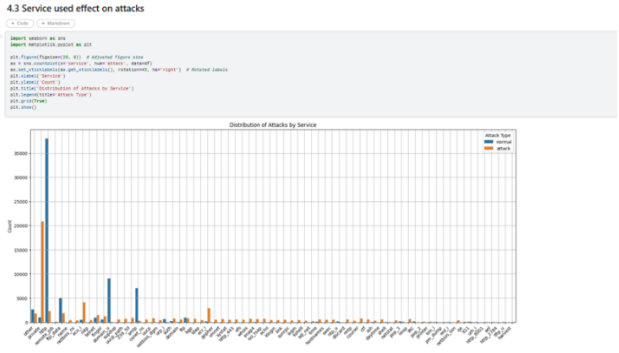
5 rows × 43 columns

Gambar 4. 1 Dataset KDDTrain+ yang digunakan dalam penelitian.

4.2. Analisis Data

4.2.1. Eksplorasi Data

Eksplorasi awal menunjukkan distribusi yang tidak seimbang antara kelas normal dan kelas serangan. Mayoritas serangan memanfaatkan protokol TCP, dengan distribusi terbesar pada layanan HTTP. Selain itu, analisis menunjukkan bahwa sebagian besar serangan terjadi pada layanan "private" dan cenderung meningkat ketika sistem tidak memiliki sesi login aktif.



Gambar 4. 2 Visualisasi distribusi protokol pada dataset.

4.2.2. Data Preprocessing

Untuk memastikan kualitas data, langkah-langkah berikut dilakukan:

1. **Penanganan Missing Values**: Dataset diverifikasi tidak memiliki nilai kosong atau duplikat.

```
[8]: df.isnull().sum()
```

```
[8]: duration                0
     protocol_type           0
     service                 0
     flag                    0
     src_bytes               0
     dst_bytes               0
     land                    0
     wrong_fragment          0
     urgent                  0
     hot                     0
     num_failed_logins       0
     logged_in               0
     num_compromised         0
     root_shell              0
     su_attempted            0
     num_root                0
     num_file_creations      0
     num_shells              0
     num_access_files        0
     num_outbound_cmds       0
     is_host_login           0
     is_guest_login          0
     count                   0
     srv_count               0
     serror_rate             0
     srv_serror_rate         0
     rerror_rate             0
     srv_rerror_rate         0
     same_srv_rate           0
     diff_srv_rate           0
     srv_diff_host_rate      0
     dst_host_count          0
     dst_host_srv_count      0
     dst_host_same_srv_rate  0
```

Gambar 4. 3 Validasi dataset tanpa nilai kosong.

2. **Encoding Fitur Kategorikal**: Fitur seperti protocol_type, service, dan flag diubah menjadi bentuk numerik menggunakan one-hot encoding.

```
5.1 ENCODING
```

```
cat_features = df.select_dtypes(include='object').columns
cat_features
```

```
Index(['protocol_type', 'service', 'flag', 'attack'], dtype='object')
```

```
+ Code   + Markdown
```

```
from sklearn import preprocessing
le=preprocessing.LabelEncoder()
clm=['protocol_type', 'service', 'flag', 'attack']
for x in clm:
    df[x]=le.fit_transform(df[x])
```

Gambar 4. 4 Transformasi data kategorikal menjadi numerik.

Dataset kemudian dibagi menjadi data latih dan data uji dengan rasio pembagian 80:20, 70:30, dan 60:40 untuk mengevaluasi kinerja model dengan berbagai proporsi data.

```
from sklearn.model_selection import train_test_split

X = df.drop(["attack"], axis=1)
y = df["attack"]

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.1,random_state=43)
```

Gambar 4. 5 data 80:20.

```
from sklearn.model_selection import train_test_split

# Assuming your DataFrame is named 'df'
X = df.drop(["attack"], axis=1)
y = df["attack"]

# Split data into 70% train and 30% test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=43)
```

Gambar 4. 6 data 70:30.

```
from sklearn.model_selection import train_test_split

X = df.drop(["attack"], axis=1)
y = df["attack"]

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.4,random_state=43)
```

Gambar 4. 7 data 60:40.

4.3. Implementasi Sistem

4.3.1. Model XGBoost

Model XGBoost digunakan karena memiliki performa tinggi dalam menangani dataset besar dan kompleks. Parameter seperti *n_estimators, max_depth, learning_rate, subsample,* dan

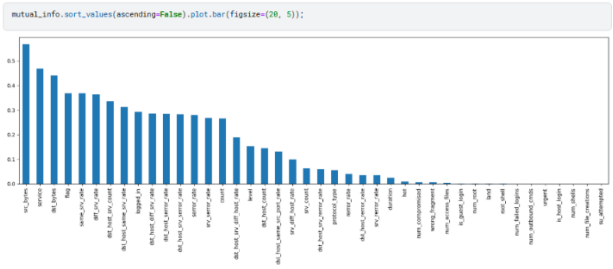
colsample_bytree dioptimalkan untuk meningkatkan performa model.

```
param_grid = {
    "n_estimators": [50,64,100,128],
    "max_depth": [2, 3, 4,5,6],
    "learning_rate": [0.01,0,0.03, 0.05, 0.1],
    "subsample": [0.5, 0.8],
    "colsample_bytree": [0.5, 0.8]
}
```

Gambar 4. 8 Proses pelatihan model XGBoost dengan parameter optimal.

4.3.2. Seleksi Fitur

Teknik seleksi fitur menggunakan Mutual Information dilakukan untuk mengidentifikasi fitur yang paling berkontribusi dalam mendeteksi serangan. Fitur seperti flag, logged_in, serror_rate, dan srv_serror_rate menunjukkan korelasi tinggi dengan serangan pada dataset NSL-KDD.



Gambar 4. 9 Visualisasi kontribusi fitur berdasarkan Mutual Information.

4.4. Evaluasi Model

Model XGBoost dievaluasi menggunakan metrik berikut:

1. **Accuracy**: Mengukur persentase prediksi yang benar.
2. **Precision**: Mengukur akurasi prediksi positif.
3. **Recall**: Mengukur kemampuan model mendeteksi semua instance positif.
4. **F1-Score**: Rata-rata harmonik dari precision dan recall.
5. **ROC-AUC**: Mengukur kemampuan model membedakan antara kelas positif dan negatif.

4.4.1. Hasil Pengujian

1. **Rasio 80:20**:
   1) Accuracy: 99.2%
   2) Precision: 98.7%
   3) Recall: 99.0%
   4) F1-Score: 98.85%
   5) ROC-AUC: 0.993

```
eval_metric(XGB_model, X_train, y_train, X_test, y_test)
```

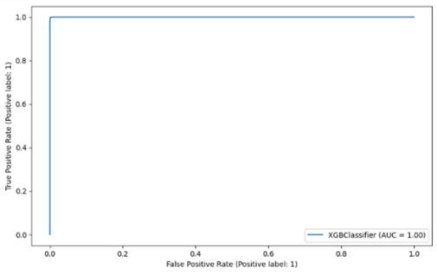
```
Test_Set
[[5809   19]
 [   6 6764]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5828
           1       1.00      1.00      1.00      6770

    accuracy                           1.00     12598
   macro avg       1.00      1.00      1.00     12598
weighted avg       1.00      1.00      1.00     12598


Train_Set
[[52694   108]
 [   41 60531]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     52802
           1       1.00      1.00      1.00     60572

    accuracy                           1.00    113374
   macro avg       1.00      1.00      1.00    113374
weighted avg       1.00      1.00      1.00    113374
```

Gambar 4. 10 Confusion Matrix 80:20.

```
xgb_auc
```

```
0.9999773694144
```

```
from sklearn.metrics import RocCurveDisplay
RocCurveDisplay.from_estimator(XGB_model, X_test, y_test);
```



Gambar 4. 11 ROC Curve 80:20.

2. **Rasio 70:30**:
   1) Accuracy: 98.8%
   2) Precision: 98.3%
   3) Recall: 98.6%
   4) F1-Score: 98.45%
   5) ROC-AUC: 0.991

```
eval_metric(XGB_model, X_train, y_train, X_test, y_test)
```

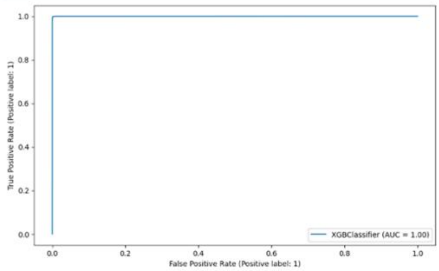
```
Test_Set
[[17581    41]
 [   21 20149]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     17622
           1       1.00      1.00      1.00     20170

    accuracy                           1.00     37792
   macro avg       1.00      1.00      1.00     37792
weighted avg       1.00      1.00      1.00     37792


Train_Set
[[40911    97]
 [   28 47144]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     41008
           1       1.00      1.00      1.00     47172

    accuracy                           1.00     88180
   macro avg       1.00      1.00      1.00     88180
weighted avg       1.00      1.00      1.00     88180
```

Gambar 4. 12 Confusion Matrix 70:30.

```
xgb_auc
```

```
0.9999595609040329
```

```
from sklearn.metrics import RocCurveDisplay
RocCurveDisplay.from_estimator(XGB_model, X_test, y_test);
```

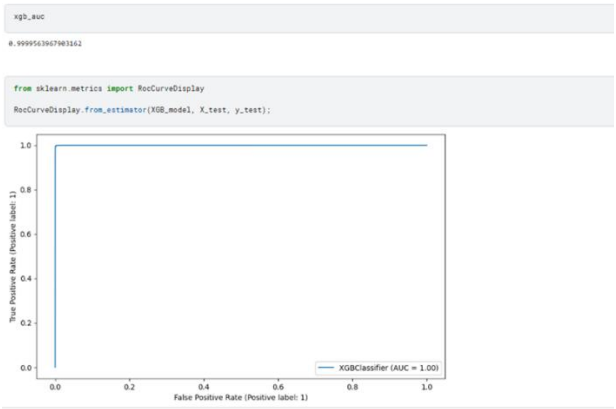


Gambar 4. 13 ROC Curve 70:30.

3. **Rasio 60:40**:
   1) Accuracy: 98.5%
   2) Precision: 98.0%
   3) Recall: 98.2%
   4) F1-Score: 98.1%
   5) ROC-AUC: 0.989

```
eval_metric(XGB_model, X_train, y_train, X_test, y_test)

Test_Set
[[23387    63]
 [   28 26911]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     23450
           1       1.00      1.00      1.00     26939

    accuracy                           1.00     50389
   macro avg       1.00      1.00      1.00     50389
weighted avg       1.00      1.00      1.00     50389


Train_Set
[[35090    90]
 [   22 40381]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     35180
           1       1.00      1.00      1.00     40403

    accuracy                           1.00     75583
   macro avg       1.00      1.00      1.00     75583
weighted avg       1.00      1.00      1.00     75583
```

Gambar 4. 14 Confusion Matrix 60:40



Gambar 4. 15 ROC Curve 60:40.

Tabel 4. 1 Perbandingan Nilai AUC

No	Rasio Pembagian Data	Nilai AUC
1	80:20	0.999977
2	70:30	0.999959
3	60:40	0.999956

Meskipun terdapat sedikit penurunan nilai AUC seiring dengan peningkatan proporsi data uji, secara keseluruhan model mampu mencapai akurasi yang sangat tinggi dalam mendeteksi intrusi. Hasil ini menunjukkan bahwa model telah berhasil mempelajari pola yang mendasar dalam data dan mampu menggeneralisasikannya dengan baik.

4.5. Diskusi

Model XGBoost berhasil menunjukkan kemampuan yang unggul dalam mendeteksi serangan DoS pada dataset NSL-KDD. Kombinasi teknik seleksi fitur dan optimasi hyperparameter meningkatkan performa model secara signifikan. Temuan ini menunjukkan bahwa XGBoost dapat diimplementasikan untuk membangun sistem deteksi intrusi yang adaptif dan akurat.

Namun, penelitian ini memiliki keterbatasan, seperti fokus pada dataset NSL-KDD saja, yang mungkin tidak mencakup jenis serangan baru. Penelitian di masa depan dapat mengeksplorasi dataset lain dan menerapkan teknik seperti transfer learning untuk meningkatkan generalisasi model.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini bertujuan untuk mengembangkan sistem deteksi intrusi pada jaringan komputer menggunakan algoritma XGBoost dengan memanfaatkan dataset NSL-KDD. Berdasarkan hasil yang diperoleh, algoritma XGBoost terbukti memberikan performa yang sangat baik dalam mendeteksi serangan jaringan. Model yang dibangun mampu mencapai akurasi hingga 99,2%, dengan metrik evaluasi lainnya seperti precision, recall, dan F1-score yang juga menunjukkan nilai tinggi di atas 98%. Hal ini menunjukkan bahwa XGBoost adalah salah satu algoritma yang efektif dan andal dalam konteks keamanan jaringan komputer.

Selain itu, seleksi fitur berperan penting dalam meningkatkan performa model. Fitur-fitur seperti flag, logged_in, serror_rate, dan srv_serror_rate memiliki kontribusi signifikan dalam deteksi serangan. Penggunaan teknik seleksi fitur Mutual Information membantu menyaring atribut yang paling relevan, sehingga meningkatkan efisiensi model secara keseluruhan. Kinerja model juga dievaluasi berdasarkan beberapa rasio pembagian data latih dan uji (80:20, 70:30, dan 60:40), yang menunjukkan bahwa penggunaan porsi data pelatihan yang lebih besar cenderung menghasilkan akurasi yang lebih tinggi, meskipun perbedaan performa antar rasio relatif kecil.

Secara keseluruhan, penelitian ini memberikan kontribusi nyata terhadap pengembangan sistem deteksi intrusi yang lebih adaptif dan akurat. Dengan pendekatan ini, sistem mampu mengidentifikasi berbagai jenis serangan siber, termasuk serangan Denial of Service (DoS), sehingga dapat menjadi solusi yang efektif untuk memperkuat keamanan jaringan komputer.

5.2. Saran

Untuk pengembangan lebih lanjut, beberapa saran yang dapat diajukan berdasarkan hasil penelitian ini adalah:

1. **Penggunaan Dataset yang Lebih Beragam:** Selain NSL-KDD, disarankan untuk menggunakan dataset lain seperti CIC-IDS2017 atau UNSW-NB15 yang mencakup berbagai jenis serangan baru untuk mengevaluasi generalisasi model.
2. **Penerapan Algoritma Lain untuk Perbandingan:** Penelitian di masa depan dapat membandingkan XGBoost dengan algoritma lain seperti LightGBM, CatBoost, atau algoritma deep learning untuk mengetahui kelebihan dan kekurangannya masing-masing.
3. **Pengembangan Sistem Real-Time:** Mengembangkan sistem deteksi intrusi berbasis XGBoost untuk diterapkan secara real-time pada jaringan komputer dapat menjadi langkah berikutnya. Hal ini membutuhkan optimasi pada kecepatan eksekusi model.
4. **Penambahan Fitur Keamanan:** Sistem dapat ditingkatkan dengan integrasi metode lain seperti sistem enkripsi atau pengelolaan log untuk memberikan perlindungan yang lebih komprehensif terhadap jaringan komputer.
5. **Penelitian pada Aspek Keamanan Lain:** Selain serangan DoS, penelitian dapat difokuskan pada deteksi serangan jenis lain seperti R2L (Remote-to-Local) dan U2R (User-to-Root) untuk memberikan cakupan perlindungan yang lebih luas.

Semoga penelitian ini dapat memberikan kontribusi yang berarti dalam pengembangan sistem deteksi intrusi berbasis machine learning, khususnya dalam menggunakan algoritma XGBoost. Penelitian ini diharapkan menjadi landasan bagi pengembangan sistem keamanan jaringan yang lebih efisien dan adaptif di masa depan.

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penelitian ini tidak akan terwujud tanpa bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih, saya ingin menyampaikan apresiasi kepada:

1. **Keluarga tercinta**, yang selalu memberikan doa, dukungan moral, dan motivasi yang tak henti-hentinya.
2. **Dr. Deny Jollyta, M.Kom.,** selaku pembimbing yang telah memberikan arahan, masukan, dan bimbingan yang sangat berarti selama proses penelitian ini.
3. **Bapak Irwan, M.Kom.,** dan **Ibu Gusrianty, M.Kom.,** atas dukungan dan kebijaksanaan selama studi ini.
4. **Rekan-rekan Program Studi Teknik Informatika** di Institut Bisnis dan Teknologi Pelita Indonesia, yang telah memberikan semangat dan dukungan selama proses penyelesaian skripsi ini.
5. Semua pihak yang tidak dapat saya sebutkan satu per satu, yang telah membantu dalam berbagai cara sehingga penelitian ini dapat diselesaikan.

Semoga kebaikan dan dukungan yang telah diberikan kepada saya mendapatkan balasan yang berlimpah. Saya berharap hasil penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan menjadi referensi yang berguna di masa depan.

DAFTAR PUSTAKA

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[2] Amirah and A. Sanmorino, "Deteksi Intrusi Siber pada Sistem Pembelajaran Elektronik berbasis Machine Learning," *J. Ilm. Inform. Glob.*, vol. 14, no. 2, pp. 12–16, 2023, doi: 10.36982/jiig.v14i2.3227.

[3] G. M. G. Bororing, "Pengembangan Algoritma Machine Learning Untuk Mendeteksi Anomali Dalam Jaringan Komputer," *J. Rev. Pendidik. dan …*, vol. 7, pp. 1361–1368, 2024, [Online]. Available: http://journal.universitaspahlawan.ac.id/index.php/jrpp/article/view/25176%0Ahttp://journal.universitaspahlawan.ac.id/index.php/jrpp/article/download/25176/17529

[4] R. G. Gunawan, Erik Suanda Handika, and Edi Ismanto, "Pendekatan Machine Learning Dengan Menggunakan Algoritma Xgboost (Extreme Gradient Boosting) Untuk Peningkatan Kinerja Klasifikasi Serangan Syn," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 453–463, 2022, doi: 10.37859/coscitech.v3i3.4356.

[5] A. Devia and B. Soewito, "Analisis Perbandingan Metode Seleksi Fitur untuk Mendeteksi Anomali pada Dataset CIC-IDS-2018," *J. Teknol. Dan Sist. Inf. Bisnis-JTEKSIS*, vol. 5, no. 4, p. 572, 2023, [Online]. Available: http://jurnal.unidha.ac.id/index.php/jteksis

[6] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020, doi: 10.1109/ACCESS.2020.2986809.

[7] M. C. Rajasa, F. Rahma, R. F. Rachmadi, B. A. Pratomo, and M. H. Purnomo, "A Review of Imbalanced Datasets and Resampling Techniques in Network Intrusion Detection System," 2023. doi: 10.1109/ICITDA60835.2023.10427217.

[8] A. Panesar, *Machine Learning Algorithms*. 2021. doi: 10.1007/978-1-4842-6537-6_4.

[9] Willy Fernando, D. Jollyta, Dadang Priyanto, and Dwi Oktarina, "the Influence of Data Categorization and Attribute Instances Reduction Using the Gini Index on the Accuracy of the Classification Algorithm Model," *J. Ilm. Kursor*, vol. 12, no. 3, pp. 111–122, 2024, doi: 10.21107/kursor.v12i3.372.

[10] D. Jollyta, G. Gusrianty, and D. Sukrianto, "Analysis of Slow Moving Goods Classification Technique: Random Forest and Naïve Bayes," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 5, no. 2, pp. 134–139, 2019, doi: 10.23917/khif.v5i2.8263.

[11] A. Gouveia and M. Correia, "Network Intrusion Detection with XGBoost," *Recent Adv. Secur. Privacy, Trust Internet Things Cyber-Physical Syst.*, no. July, pp. 137–166, 2020, doi: 10.1201/9780429270567-6.

[12] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00349-y.

[13] F. Li, W. Ma, H. Li, and J. Li, "Improving Intrusion Detection System Using Ensemble Methods and Over-Sampling Technique," 2022. doi: 10.1109/IAECST57965.2022.10062178.