**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

# Optimasi Deteksi Intrusi Jaringan Menggunakan Hybrid Model Autoencoder dan Random Forest

**Afri Nanda*, Torkis Nasution, Rahmaddeni, Herwin**

Fakultas Teknik dan Informatika, Prodi Teknik Informatika, Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia
Email: [1,*]afrinanda66@gmail.com, [2]torkisnasution@usti.ac.id, [3]rahmaddeni@usti.ac.id, [4]herwin@usti.ac.id
Email Penulis Korespondensi: afrinanda66@gmail.com
Submitted: **29/01/2026**; Accepted: **05/03/2026**; Published**: 06/03/2026**

**Abstrak**−Sistem deteksi intrusi konvensional sering kali mengalami degradasi performa akibat ketidakmampuan menangani kompleksitas data berdimensi tinggi dan ketidakseimbangan distribusi kelas (class imbalance) pada lalu lintas jaringan modern. Penelitian ini bertujuan untuk mengoptimalkan sistem Deteksi Intrusi Jaringan (Intrusion Detection System) dengan mengatasi keterbatasan algoritma Random Forest dalam menangani data berdimensi tinggi dan kurangnya transparansi model (black-box). Metode yang diusulkan adalah model Hybrid yang mengintegrasikan Autoencoder sebagai pengekstraksi fitur non-linier dan Random Forest sebagai pengklasifikasi. Autoencoder dilatih dengan strategi semi-terawasi untuk menghasilkan fitur laten dan Reconstruction Error (MSE) yang berfungsi sebagai indikator anomali yang kuat. Selain itu, teknik SMOTE diterapkan untuk menangani ketidakseimbangan kelas pada dataset NSL-KDD. Guna menjawab tantangan interpretabilitas, metode Explainable AI (XAI) berbasis SHAP diterapkan secara strategis untuk menjelaskan interaksi kompleks antara fitur laten hasil kompresi Autoencoder dengan keputusan akhir klasifikasi, sehingga mentransformasi arsitektur hibrida ini menjadi sistem yang transparan. Hasil evaluasi menunjukkan bahwa model Hybrid Autoencoder-Random Forest unggul dibandingkan Random Forest dengan peningkatan Akurasi sebesar 2,54% (menjadi 77,61%) dan Recall sebesar 3,96% (menjadi 62,31%). Peningkatan signifikan pada metrik Recall membuktikan efektivitas fitur hibrida, khususnya Reconstruction Error, dalam mendeteksi serangan Zero-Day yang memiliki pola asing. Visualisasi SHAP juga berhasil mengungkap kontribusi fitur laten, memberikan transparansi yang krusial bagi analisis forensik keamanan jaringan.

**Kata Kunci**: Intrusion Detection System; Hybrid Model; Autoencoder; Random Forest; SHAP; Zero-Day Attack

**Abstract**−Conventional Intrusion Detection Systems often suffer from performance degradation due to their inability to handle the complexity of high-dimensional data and class imbalance in modern network traffic. This study aims to optimize the Network Intrusion Detection System (IDS) by addressing the limitations of the Random Forest algorithm in handling high-dimensional data and its lack of model transparency (black-box). The proposed method is a Hybrid model integrating an Autoencoder as a non-linear feature extractor and Random Forest as a classifier. The Autoencoder is trained using a semi-supervised strategy to generate latent features and Reconstruction Error (MSE), which serves as a robust anomaly indicator. Additionally, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to address class imbalance in the NSL-KDD dataset. To address the challenge of interpretability, SHAP-based Explainable AI (XAI) is strategically implemented to elucidate the complex interactions between the Autoencoder-compressed latent features and the final classification decisions, thereby transforming this hybrid architecture into a transparent system. Evaluation results demonstrate that the Hybrid Autoencoder-Random Forest model outperforms the Random Forest Baseline, achieving an Accuracy increase of 2.54% (to 77.61%) and a Recall increase of 3.96% (to 62.31%). The significant improvement in the Recall metric empirically validates the effectiveness of hybrid features, specifically the Reconstruction Error, in detecting Zero-Day attacks characterized by unknown patterns. Furthermore, SHAP visualization successfully reveals the contribution of latent features, providing crucial transparency for network security forensic analysis.

**Keywords**: Intrusion Detection System; Hybrid Model; Autoencoder; Random Forest; SHAP; Zero-Day Attack

## 1. PENDAHULUAN

Integritas infrastruktur digital kini menghadapi tekanan yang belum pernah terjadi sebelumnya. Perkembangan arsitektur jaringan yang meluas ke ranah *Internet of Things* (*IoT*), *cloud*, dan *edge computing* telah memperluas permukaan serangan (*attack surface*) [1], [2]. Kondisi ini memungkinkan aktor ancaman untuk mengeksploitasi celah keamanan dengan metode yang semakin adaptif dan kompleks [3], [4]. Fenomena ini menegaskan bahwa pendekatan keamanan konvensional tidak lagi memadai untuk membendung laju serangan siber modern yang bervolume tinggi.

Peningkatan ancaman ini tercermin nyata dalam serangkaian insiden siber yang mengguncang infrastruktur vital Indonesia sepanjang tahun 2024. Serangan *ransomware* varian *LockBit* 3.0 atau "*Brain Cipher*" terhadap Pusat Data Nasional (PDN) pada bulan Juni telah melumpuhkan layanan publik di lebih dari 200 instansi, mengungkap kerentanan fatal dalam sistem pemerintahan [5], [6]. Kemudian pada bulan Agustus, situasi makin bertambah buruk dengan adanya dugaan kebocoran 4,7 juta data Aparatur Sipil Negara (ASN) di Badan Kepegawaian Negara (BKN) yang diperjualbelikan di salah satu forum ilegal [7], [8]. Tidak lama berselang, pada bulan September kembali muncul insiden kebocoran sekitar 6 juta data wajib pajak milik Direktorat Jenderal Pajak (DJP) [9], [10]. Rentetan kasus ini bukan sekadar statistik, melainkan indikator krisis yang menuntut implementasi sistem pertahanan yang lebih cerdas dan proaktif.

Eskalasi ancaman terhadap infrastruktur digital terus menunjukkan tren yang mengkhawatirkan. Berdasarkan data pemantauan keamanan siber sepanjang tahun 2024, tercatat total ancaman siber mencapai angka masif sebesar 9.463.546 insiden. Tingginya volume serangan ini terlihat berfluktuasi namun konsisten, dengan puncak intensitas tertinggi terjadi pada bulan April yang mencatat 1.147.195 ancaman, diikuti oleh lonjakan signifikan lainnya pada

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

bulan Agustus dan September. Besarnya skala serangan ini mengindikasikan bahwa metode deteksi konvensional tidak lagi memadai untuk menangani lalu lintas berbahaya yang begitu padat dan dinamis [11].

Untuk mencegah insiden serupa, pemerintah ataupun pihak yang terkait perlu menerapkan pendekatan keamanan yang komprehensif, salah satunya melalui penerapan *Intrusion Detection System* (IDS) berbasis *machine learning* modern yang mampu mengenali pola intrusi atau serangan secara cepat, akurat, dan otomatis, sehingga mampu mengenali ancaman lebih dini, termasuk serangan yang bersifat *unknown* atau *zero-day* [12], [13], [14]. Pendekatan ini memungkinkan sistem untuk mengenali karakteristik serangan baru berdasarkan pola data, bukan sekadar ketergantungan pada tanda tangan yang telah diketahui sebelumnya.

Dalam konteks deteksi intrusi jaringan, algoritma *Random Forest* sangat cocok sebagai metode klasifikasi standar (*benchmark*) karena karakteristiknya yang *robust* terhadap variasi data. Penelitian yang dilakukan oleh Nanda et al. [15] menunjukkan bahwa *Random Forest* mampu mencapai tingkat akurasi dan presisi yang sangat tinggi dalam mendeteksi serangan jaringan, bahkan melampaui performa algoritma klasik seperti *Support Vector Machine* (SVM) dan *Regresi Logistik*. Kemudian, ketangguhan *Random Forest* juga teruji pada skenario data yang tidak seimbang (*imbalanced*), sebagaimana pada penelitian yang menggunakan dataset Honeynet BSSN, di mana algoritma ini tetap mampu mempertahankan stabilitas akurasi di atas 90% meskipun dihadapkan pada distribusi kelas serangan yang sangat timpang sebagaimana dilaporkan oleh Inayah dan Ramli [16].

Namun, penerapan tunggal *Random Forest* memiliki keterbatasan fundamental, terutama saat dihadapkan pada trafik jaringan modern yang berdimensi tinggi (*high-dimensional data*). Hasil penelitian yang didapatkan oleh L. Mhamdi and Isa [17] sejalan dengan Wang et al. [18] menyoroti bahwa performa algoritma *tree-based* cenderung menurun ketika harus memproses fitur yang mengandung *noise* atau korelasi non-linier yang kompleks. Selain itu, V. Hassija et al. [19] dan Udofot et al. [20] menemukan bahwa sifat *Random Forest* sebagai model *black-box* menjadi hambatan tersendiri karena keputusan deteksi yang akurat sering kali sulit dijelaskan, padahal aspek transparansi sangat krusial dalam audit forensik keamanan jaringan.

Selain itu, permasalahan ketidakseimbangan data (*imbalanced data*) yang umum terjadi pada dataset deteksi intrusi jaringan juga berpotensi menurunkan performa model dan meningkatkan *false positive rate*, sehingga perlu menggunakan teknik berbasis *Synthetic Minority Over-sampling Technique* (SMOTE) untuk menyeimbangkan distribusi kelas sebelum proses pelatihan model [21]. Namun, penerapan SMOTE pada ruang fitur mentah berdimensi tinggi dapat menghasilkan sampel sintetis yang kurang representatif, terutama ketika data masih mengandung *noise* dan korelasi non-linier yang kompleks.

Evaluasi terhadap literatur terdahulu memperlihatkan kesenjangan penelitian yang perlu dijembatani. Meskipun Nanda et al. [15] serta Inayah dan Ramli [16] telah mengonfirmasi keunggulan *Random Forest* dalam hal akurasi, studi lanjutan oleh Mhamdi dan Isa [17] serta Wang et al. [18] mengidentifikasi perlunya integrasi Autoencoder untuk menangani kelemahan pada data dimensi tinggi. Sayangnya, pendekatan hibrida yang mereka usulkan masih berfokus pada peningkatan akurasi klasifikasi semata tanpa menyentuh aspek transparansi. Padahal, kebutuhan akan *Explainable AI* (XAI) untuk mengatasi masalah black-box telah divalidasi urgensinya oleh Udofot et al. [20].

Merespons tantangan tersebut, penelitian ini mengusulkan arsitektur hibrida baru yang mengintegrasikan *Autoencoder* sebagai pengekstraksi fitur non-linier dan *Random Forest* sebagai pengklasifikasi utama. Secara umum, Autoencoder dikenal mampu mempelajari representasi fitur laten (*latent representation*) yang efektif dalam menangkap pola non-linier pada data berdimensi tinggi [22], [23]. Dalam penelitian ini, *Autoencoder* tidak hanya dimanfaatkan sebagai teknik reduksi dimensi, tetapi dilatih secara *semi-supervised* menggunakan data normal untuk membangun model normalitas yang representatif. Strategi tersebut menghasilkan fitur *reconstruction error* sebagai sinyal anomali eksplisit. Trafik dengan pola yang tidak terdapat dalam data latih akan menghasilkan nilai *reconstruction error* yang lebih tinggi, sehingga meningkatkan kemampuan sistem dalam mendeteksi serangan varian baru atau *Zero-Day*.

Selain itu, untuk mengatasi ketidakseimbangan data, teknik SMOTE diterapkan pada ruang fitur laten yang telah terkompresi, bukan pada data mentah. Strategi ini bertujuan menghasilkan distribusi kelas yang lebih stabil serta meminimalkan pembentukan sampel sintetis yang mengandung *noise*.

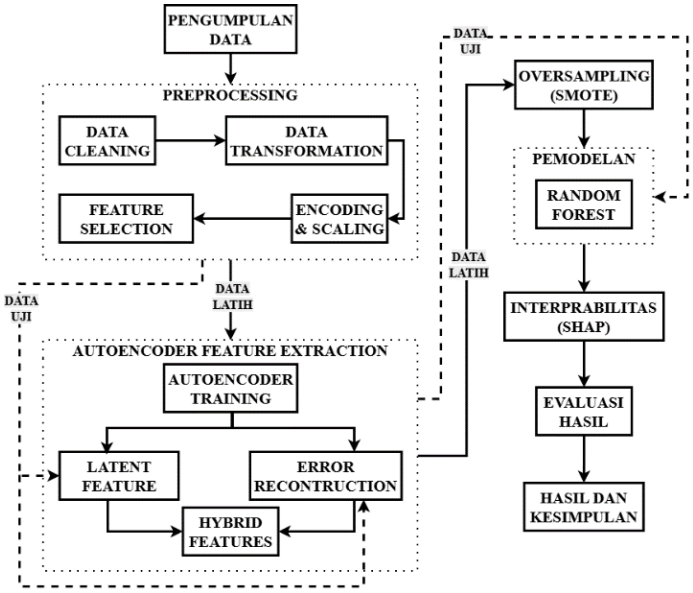
Sebagai penyempurnaan terhadap keterbatasan interpretabilitas *Random Forest*, penelitian ini mengintegrasikan pendekatan *Explainable AI* berbasis SHAP untuk memvisualisasikan kontribusi fitur laten terhadap keputusan deteksi. Dengan demikian, sistem yang diusulkan tidak hanya berorientasi pada peningkatan akurasi dan adaptivitas terhadap serangan baru, tetapi juga menghadirkan transparansi dan kepercayaan dalam proses pengambilan keputusan.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimen untuk menilai efektivitas model hybrid Autoencoder dan Random Forest dalam mendeteksi intrusi pada jaringan komputer. Tahapan dari penelitian ini ditampilkan pada Gambar 1.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309




**Gambar 1.** Tahapan Penelitian

Gambar 1 menampilkan alur penelitian yang dimulai dari pengolahan data latih sebagai basis pembelajaran model. Proses dimulai dari preprocessing, normalisasi, dan seleksi fitur menggunakan ANOVA. Selanjutnya, *Autoencoder* dilatih hanya menggunakan data normal untuk menghasilkan model ekstraksi fitur. Setelah model *Autoencoder* terbentuk, seluruh data latih (Normal dan Serangan) dikonversi menjadi fitur laten dan *reconstruction error*. Pada tahap inilah teknik SMOTE diterapkan, yaitu menyeimbangkan jumlah kelas pada ruang fitur laten sebelum data dimasukkan ke dalam pelatihan *Random Forest*. Pendekatan ini dipilih agar sintesis data dilakukan pada representasi fitur yang lebih padat dan bermakna. Sebaliknya, data uji digunakan murni untuk evaluasi eksternal tanpa melalui proses SMOTE maupun pembelajaran ulang, guna menjaga validitas pengujian serta mencegah terjadinya bias dan kebocoran informasi (*data leakage*).

### 2.2 Pengumpulan Data

Penelitian ini diawali dengan pengumpulan dataset NSL-KDD. Dataset NSL-KDD merupakan versi pengembangan dari dataset KDD Cup 1999. NSL-KDD terdiri dari data lalu-lintas jaringan yang dikategorikan menjadi dua kelas utama, yaitu normal dan serangan, di mana serangan dibagi ke dalam empat kategori yaitu *DoS*, *Probe*, *U2R*, dan *R2L* [24]. Seluruh proses pengolahan data dilakukan menggunakan *Google Colab*. Dataset NSL-KDD yang peneliti gunakan telah otomatis dibagi menjadi data training dan testing, sehingga proses pembagian data tidak perlu dilakukan kembali pada tahap *preprocessing*. Rincian distribusi dan karakteristik fitur dataset NSL-KDD dirangkum pada Tabel 1.

**Tabel 1.** Fitur Dataset NSL-KDD

| Kategori Fitur | Jumlah | Contoh Fitur Representatif |
|---|---|---|
| Basic Features | 9 | *duration, protocol_type, service, src_bytes, dst_bytes, flag* |
| Content Features | 13 | *logged_in, root_shell, num_failed_logins, is_guest_login, hot* |
| Time-based Traffic | 9 | *count, srv_count, serror_rate, rerror_rate, same_srv_rate* |
| Host-based Traffic | 10 | *dst_host_count, dst_host_srv_count, dst_host_same_src_port_rate* |
| Label & Difficulty | 2 | *class, difficulty_level* |

Tabel 1 merincikan ke-41 fitur yang terdapat dalam dataset NSL-KDD, yang dikelompokkan berdasarkan karakteristik ekstraksinya. Secara garis besar, fitur-fitur ini terbagi menjadi empat kategori utama yaitu fitur dasar yang mengambil informasi dari *header* TCP/IP, kemudian fitur konten yang menganalisis isi *payload* untuk mendeteksi perilaku mencurigakan serta fitur berbasis waktu dan fitur berbasis host yang menghitung statistik lalu lintas jaringan untuk mengidentifikasi pola serangan volumetrik seperti *Denial of Service* maupun serangan *Probe* yang lebih halus.

### 2.3 *Preprocessing*

Sebelum model dilatih, data terlebih dahulu melalui tahapan preprocessing untuk memastikan kualitas dan keseragaman data yang digunakan. Proses ini diawali dengan penyesuaian nama atribut sesuai struktur dataset NSL-KDD, kemudian dilakukan pembersihan data dengan menghapus nilai tidak valid, data duplikat, serta baris yang mengandung nilai hilang. Selanjutnya, atribut *class* dikonversi menjadi label biner, di mana trafik normal diberi nilai 0 dan trafik serangan diberi nilai 1. Pendekatan ini digunakan untuk menyederhanakan permasalahan menjadi klasifikasi dua kelas serta mendukung proses deteksi intrusi.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

Pada tahap *encoding & scaling*, atribut kategorikal seperti *protocol_type, service*, dan *flag* diubah menggunakan metode *One-Hot Encoding*, sedangkan atribut numerik dinormalisasi menggunakan *Standard Scaler* (*Z-Score Normalization*) agar seluruh fitur berada pada skala yang sebanding. Setelah proses tersebut, dilakukan seleksi fitur menggunakan ANOVA F-Test sebagai langkah reduksi dimensi tahap awal. Tujuannya adalah menyaring fitur-fitur yang memiliki relevansi statistik tertinggi terhadap label kelas dan membuang fitur yang bersifat *noise* atau redundan. Fitur-fitur terpilih dari tahap ini kemudian digunakan sebagai *input* untuk pelatihan model *Autoencoder*.

### 2.4 Autoencoder

Autoencoder merupakan arsitektur jaringan saraf tiruan (*Neural Network Architecture*) dan termasuk dalam kategori *unsupervised learning* [25]. Dalam penelitian ini, Autoencoder tidak hanya berfungsi untuk mereduksi dimensi, tetapi lebih utama sebagai *Feature Extractor* dan penghasil sinyal anomali (*reconstruction error*).

Penggunaan Autoencoder setelah tahap seleksi fitur ANOVA dilakukan dengan alasan dua tahap pemrosesan fitur:

a. Eliminasi noise (ANOVA), *Autoencoder* sensitif terhadap input yang kotor. Dengan menyaring fitur menggunakan ANOVA terlebih dahulu, *Autoencoder* dapat fokus mempelajari pola laten dari fitur-fitur yang terbukti relevan, tanpa terganggu oleh atribut yang tidak penting.
b. Ekstraksi pola non-linear (*Autoencoder*), ANOVA hanya melihat hubungan linear antar fitur secara individu. *Autoencoder* melengkapinya dengan mempelajari hubungan non-linear yang kompleks antar fitur tersebut dan memadatkannya menjadi vektor laten.

Pada penelitian ini, *Autoencoder* digunakan dengan strategi khusus seperti berikut:

a. *Training* pada Data Normal: *Autoencoder* hanya dilatih menggunakan data lalu lintas Normal. Hal ini memaksa model untuk mempelajari pola normalitas secara mendalam. Pendekatan ini bertujuan agar *Autoencoder* mempelajari pola normalitas secara representatif, sehingga data serangan akan menghasilkan pola laten yang berbeda ketika diproyeksikan ke ruang fitur.
b. *Latent Features*: Vektor di lapisan tengah (*bottleneck*) diambil sebagai fitur baru yang lebih padat (*compressed*).
c. *Reconstruction Erro*r (MSE): Selisih antara input asli dan hasil rekonstruksi dihitung menggunakan *Mean Squared Error* (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_1')^2 \quad (1)$$

Nilai MSE merepresentasikan rata-rata kuadrat perbedaan antara input asli dan hasil rekonstruksi, semakin kecil MSE maka semakin baik model mempelajari pola normal. Dengan cara ini, data serangan yang tidak sesuai pola normal akan menghasilkan nilai MSE lebih tinggi, sehingga dapat digunakan sebagai indikator anomali.

### 2.5 Oversampling

Ketidakseimbangan data (*class imbalance*) adalah masalah umum pada dataset keamanan siber, di mana jumlah data normal jauh lebih banyak daripada data serangan. Hal ini membuat model cenderung bias ke kelas mayoritas. SMOTE menangani hal ini dengan mensintesis sampel baru untuk kelas minoritas [21]. Dalam arsitektur *hybrid* ini, penerapan SMOTE dilakukan secara strategis setelah proses ekstraksi fitur oleh Autoencoder. Artinya, SMOTE tidak diterapkan pada data mentah yang berdimensi tinggi, melainkan pada fitur laten (*latent features*) dan nilai MSE. Hal ini bertujuan untuk menghasilkan sampel sintetis yang lebih berkualitas karena dibentuk dari representasi data yang telah terkompresi dan bebas dari noise. SMOTE menggunakan KNN untuk menghasilkan sampel sintetis dari kelas minoritas. Berikut rumus pembentukan sampel sintetis sederhananya:

$$x_{new} = x_i + \delta * (x_{i,NN} - x_i) \quad (2)$$

Dalam SMOTE, setiap sampel dari kelas minoritas $x_i$ dipilih, kemudian salah satu dari k-tetangga terdekatnya $x_{i,NN}$ digunakan untuk membuat titik baru. Titik baru $x_{new}$ dibentuk di sepanjang garis antara sampel asli dan tetangganya, dengan jarak ditentukan secara acak oleh $\delta$ antara 0 dan 1. Dengan demikian, SMOTE menghasilkan sampel sintetis baru yang meningkatkan keseimbangan antara kelas mayoritas dan minoritas, tanpa sekadar menyalin data lama.

### 2.6 Random Forest

*Random Forest* merupakan pendekatan ensemble learning yang membentuk banyak pohon keputusan berdasarkan sampel data latih yang diambil secara acak melalui teknik *bootstrap* [27, p. 35]. Setiap pohon dikonstruksi menggunakan kombinasi fitur yang berbeda-beda, sehingga variasi antar pohon dapat meningkatkan kestabilan model. Mekanisme penggabungan prediksi dari seluruh pohon tersebut memungkinkan *Random Forest* mengurangi kecenderungan *overfitting* serta menghasilkan kemampuan generalisasi yang lebih baik dibandingkan penggunaan satu pohon keputusan saja. Meskipun *Random Forest* dikenal sebagai salah satu metode *ensemble learning* yang kuat dan tahan terhadap *overfitting*, sejumlah literatur mencatat keterbatasan pada penggunaannya. Pertama, kompleksitas model ini membuat interpretasi hasil menjadi kurang transparan, sehingga kurang cocok untuk aplikasi yang membutuhkan penjelasan keputusan model secara eksplisit. Selain itu, pembentukan ratusan hingga ribuan pohon

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

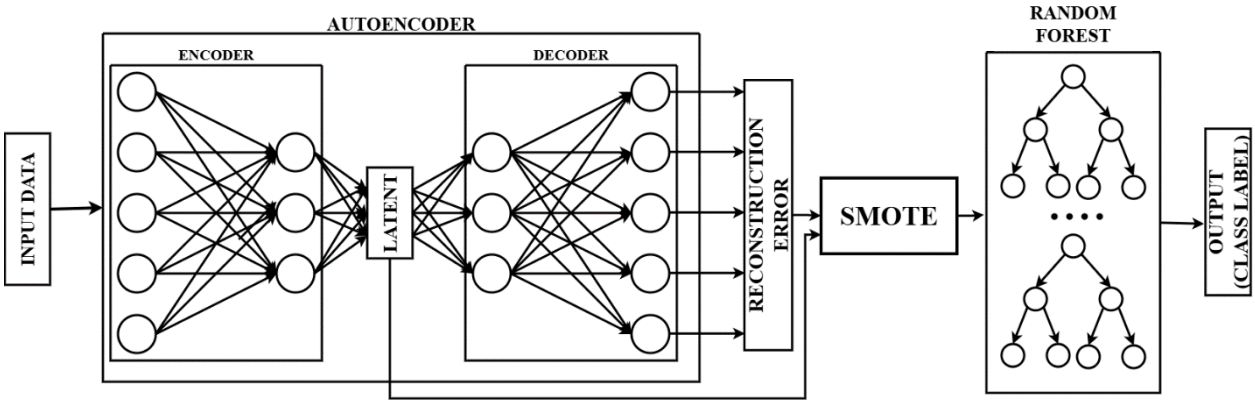
keputusan meningkatkan kebutuhan komputasi baik pada fase pelatihan maupun prediksi, yang dapat menjadi kendala pada dataset berukuran besar atau lingkungan komputasi terbatas [19], [20]. Secara matematis, indeks Gini dirumuskan sebagai berikut [27, p. 11]:

$$Gini(D) = 1 - (p_1^2 + p_2^1) \quad (3)$$

*Random Forest* menggunakan *Gini Impurity* untuk memilih atribut pemisah terbaik. Nilai *Gini Impurity* serendah mungkin (mendekati 0) menunjukkan tingkat kemurnian data yang tinggi, sedangkan nilai yang lebih tinggi mengindikasikan ketidakmurnian data pada node tersebut [26, p. 80].

### 2.7 *Hybrid Autoencoder* dan *Random Forest*

Untuk memberikan gambaran menyeluruh mengenai metode yang diusulkan, Gambar 2 memvisualisasikan skema arsitektur *Hybrid Autoencoder–Random Forest*.




**Gambar 2.** *Hybrid Autoencoder Random Forest*

Gambar 2 merupakan arsitektur *hybrid* yang mengintegrasikan keunggulan unsupervised dan supervised learning melalui tiga tahap utama yang saling berkesinambungan. Proses diawali dengan ekstraksi fitur menggunakan *Autoencoder*, di mana data berdimensi tinggi hasil seleksi ANOVA dikompresi ke dalam *latent space* untuk menghasilkan vektor fitur laten dan nilai *reconstruction error*. Selanjutnya, teknik SMOTE diterapkan secara spesifik pada representasi fitur laten dan *reconstruction error* tersebut guna menyeimbangkan distribusi kelas antara trafik normal dan serangan. Pada tahap akhir, *Random Forest* dilatih menggunakan data yang telah seimbang tersebut untuk menganalisis pola ruang laten yang diperkaya sinyal anomali, sehingga menghasilkan prediksi kelas yang akurat dan stabil.

### 2.8 *SHapley Additive exPlanations* (*SHAP*)

Dalam penelitian ini, metode *Explainable AI* yakni *SHapley Additive exPlanations* (SHAP) diterapkan untuk menginterpretasikan hasil prediksi model. SHAP memanfaatkan konsep nilai *Shapley* dari teori permainan untuk memberikan kontribusi fitur terhadap setiap prediksi secara lokal dan global. Hal ini memungkinkan pemahaman terhadap fitur laten hasil *Autoencoder* dan fitur dominan asli yang paling memengaruhi keputusan klasifikasi serangan atau normal. Penerapan SHAP juga konsisten dengan studi-studi terbaru dalam domain IDS yang menunjukkan bahwa interpretabilitas model meningkatkan transparansi dan kepercayaan model, serta membantu menganalisis pola kesalahan prediksi pada sistem deteksi intrusi berbasis machine learning. Secara matematis, untuk fitur $i$ dalam model dengan set fitur $p$ [26, p. 184] :

$$\phi_i = \sum \frac{|S|!*(|p|-|S|-1)!}{|p|!} * [v(S \cup \{i\}) - v(S)] \quad (4)$$

Nilai *SHAP* $\phi_i$ untuk sebuah fitur $i$ dihitung sebagai rata-rata kontribusi fitur tersebut terhadap prediksi model di semua kombinasi subset fitur yang mungkin. Setiap subset $S$ yang tidak memuat fitur $i$ dibandingkan prediksi model dengan dan tanpa fitur $i$. Faktor $\frac{|S|!*(|p|-|S|-1)!}{|p|!}$ merupakan bobot adil dari teori *Shapley* untuk memastikan setiap kontribusi fitur dihitung secara proporsional. Dengan cara ini, *SHAP* menunjukkan seberapa besar pengaruh setiap fitur terhadap prediksi model.

### 2.9 Evaluasi

Evaluasi ini dilakukan untuk mengukur dan membandingkan kinerja model *Hybrid Autoencoder-Random Forest* dan model Random Forest. Untuk mengukur kinerja sistem usulan, digunakan metrik evaluasi berdasarkan *Confusion Matrix*:

a. *True Positive* (TP): Serangan terdeteksi dengan benar sebagai serangan.
b. *True Negative* (TN): Normal terdeteksi dengan benar sebagai normal.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

c. *False Positive* (FP): Normal salah dideteksi sebagai serangan (Alarm Palsu).
d. *False Negative* (FN): Serangan salah dideteksi sebagai normal (Serangan Lolos).

Berdasarkan *confusion matrix* tersebut, metrik yang dapat dihitung sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$Presisi = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 - Score = 2 * \frac{Presisi * Recall}{Presisi + Recall} \quad (8)$$

# 3. HASIL DAN PEMBAHASAN

## 3.1 Pengumpulan Data

Tahapan inisiasi eksperimen dimulai dengan proses pengumpulan data, di mana dataset NSL-KDD yang didapatkan dari *Kaggle* akan dimuat ke dalam *Google Colab*. Dataset ini terdiri dari dua file terpisah, yaitu KDDTrain yang digunakan untuk melatih model dan KDDTest untuk evaluasi performa. Karena data mentah disimpan dalam repositori Google Drive, dilakukan proses mounting drive terlebih dahulu agar file dapat diakses secara langsung. Selanjutnya, pustaka Pandas digunakan untuk membaca file tersebut dan mengonversinya menjadi struktur Dataframe. Dikarenakan data asli tidak memiliki header, proses input ini juga mencakup pemberian nama atribut secara manual pada ke-43 kolom agar setiap fitur memiliki identitas yang jelas untuk pemrosesan selanjutnya.

**Tabel 2.** Data Latih

| | duration | protocol_type | … | dst_host_srv_rerror_rate | Class | difficulty_level |
|---|---|---|---|---|---|---|
| 1 | 0 | tcp | … | 0,00 | Normal | 21 |
| 2 | 0 | udp | … | 0,00 | Normal | 21 |
| … | … | … | … | … | … | … |
| 125972 | 0 | tcp | … | 0,00 | Neptune | 21 |
| 125973 | 0 | tcp | … | 0,00 | normal | 14 |

Tabel 2 adalah struktur dari data latih NSL-KDD yang terdiri dari 125.973 baris dan 43 kolom trafik jaringan yang mencakup fitur dasar seperti durasi dan protokol, serta fitur statistik trafik. Variabel target pada kolom terakhir menunjukkan klasifikasi trafik, di mana terlihat adanya variasi antara aktivitas 'normal' dan serangan spesifik seperti 'neptune'.

**Tabel 3.** Data Uji

| | duration | protocol_type | … | dst_host_srv_rerror_rate | class | difficulty_level |
|---|---|---|---|---|---|---|
| 1 | 0 | tcp | … | 1,00 | Neptune | 21 |
| 2 | 0 | tcp | … | 1,00 | Neptune | 21 |
| … | … | … | … | … | … | … |
| 22543 | 0 | udp | … | 0,00 | Normal | 21 |
| 22544 | 0 | tcp | … | 1,00 | mscan | 14 |

Tabel 3 adalah struktur dari data uji NSL-KDD yang terdiri dari 22.544 baris dan 43 kolom. Berbeda dengan data latih, data uji ini memuat varian serangan baru yang tidak diperkenalkan selama fase pelatihan, salah satunya terlihat pada entitas ke-22.544 dengan label *'mscan'*. Keberadaan varian serangan asing ini sangat vital untuk memvalidasi kemampuan model *Hybrid Autoencoder Random Forest* dalam mendeteksi serangan *Zero-Day* atau serangan yang polanya belum dikenali sebelumnya oleh sistem. Kedua data ini selanjutnya akan melalui tahap preprocessing.

## 3.2 *Preprocessing* Data

*Preprocessing* data dilakukan sebagai langkah krusial untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik serta relevan terhadap tujuan deteksi intrusi jaringan. Setiap tahapan dirancang untuk mengurangi noise, menyederhanakan representasi data, dan meningkatkan kemampuan model dalam mengenali pola serangan. Adapun tahapan *preprocessing* yang dilakukan ada pada Tabel 4.

**Tabel 4.** Hasil *Preprocessing*

| Tahapan | Contoh Hasil |
|---|---|
| *Data Cleaning* | Menghapus kolom *difficulty_level* (kolom ke-43) karena merupakan metadata kompetisi, bukan fitur jaringan. |

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

| | |
|---|---|
| *Data Transformation (Target Labeling)* | normal menjadi 0 dan semua jenis serangan menjadi 1. |
| *Encoding (One-Hot Encoding)* | *protocol_type* ('tcp') dipecah menjadi kolom baru sehingga total fitur naik jadi 122 kolom. |
| *Scaling (StandardScaler)* | Nilai *src_bytes* = 491 berubah menjadi -0.0078. Nilai *dst_bytes* = 0 berubah menjadi -0.0049 sehingga fitur besar tidak mendominasi fitur kecil. |
| *Feature Selection (ANOVA)* | Dari 122 fitur, dipilih 30 fitur terbaik sehingga mengurangi beban komputasi model. |

Tahapan awal dimulai dengan pembersihan data untuk memastikan bahwa hanya informasi yang benar-benar merepresentasikan karakteristik lalu lintas jaringan yang digunakan dalam proses pembelajaran. Penghapusan atribut *difficulty_level* dilakukan karena atribut tersebut tidak berkaitan langsung dengan perilaku jaringan, melainkan hanya berfungsi sebagai metadata dalam konteks kompetisi. Dengan menghilangkan atribut ini, model difokuskan pada pola trafik aktual tanpa terdistorsi oleh informasi tambahan yang tidak relevan.

Selanjutnya, proses pelabelan ulang terhadap data memberikan dampak signifikan dalam menyederhanakan permasalahan klasifikasi. Seluruh jenis serangan yang sebelumnya tersebar dalam banyak kelas digabungkan menjadi satu kategori serangan, sementara lalu lintas normal dipertahankan sebagai kelas tersendiri. Pendekatan ini memungkinkan model untuk lebih berkonsentrasi pada kemampuan membedakan antara aktivitas normal dan aktivitas mencurigakan, yang merupakan tujuan utama sistem deteksi intrusi. Hasilnya, distribusi kelas menjadi lebih jelas dan interpretasi performa model dapat dilakukan secara lebih terfokus.

Transformasi fitur kategorikal melalui *One-Hot Encoding* menyebabkan peningkatan jumlah atribut secara signifikan. Meskipun hal ini menambah kompleksitas dimensi data, representasi tersebut memungkinkan setiap kategori diperlakukan secara independen tanpa menimbulkan bias hubungan numerik semu. Peningkatan dimensi ini terbukti penting karena memungkinkan model menangkap variasi protokol, layanan, dan status koneksi jaringan secara lebih detail, yang sebelumnya tidak dapat direpresentasikan secara optimal dalam bentuk nilai tunggal.

Normalisasi data menggunakan *StandardScaler* memberikan kontribusi nyata dalam menyeimbangkan pengaruh antar fitur numerik. Sebelum proses ini dilakukan, fitur dengan nilai besar cenderung mendominasi proses pembelajaran model. Setelah normalisasi, seluruh fitur berada pada skala yang relatif seragam, sehingga model dapat mempelajari pola serangan berdasarkan hubungan antar fitur, bukan semata-mata pada besar kecilnya nilai tertentu. Kondisi ini sangat penting terutama pada model berbasis pembelajaran mendalam dan ensemble, yang sensitif terhadap perbedaan skala data.

Meningkatnya jumlah fitur akibat proses encoding kemudian diimbangi melalui seleksi fitur menggunakan metode ANOVA. Hasil seleksi menunjukkan bahwa tidak semua fitur hasil transformasi memiliki kontribusi yang signifikan dalam membedakan lalu lintas normal dan serangan. Dengan memilih sejumlah fitur paling diskriminatif, kompleksitas data berhasil ditekan tanpa mengorbankan informasi penting. Reduksi dimensi ini berdampak langsung pada efisiensi komputasi serta kestabilan model pada tahap pelatihan dan pengujian.

### 3.3 *Autoencoder Training*

Pada tahapan ini, *Autoencoder* dilatih dengan strategi *Semi-Supervised Anomaly Detection*, di mana model hanya dipaparkan pada data normal untuk mempelajari distribusi probabilitas fitur normal. Hal ini bertujuan agar *reconstruction error* dapat berfungsi sebagai fitur diskriminator yang kuat saat data serangan (*outliers*) diproses.

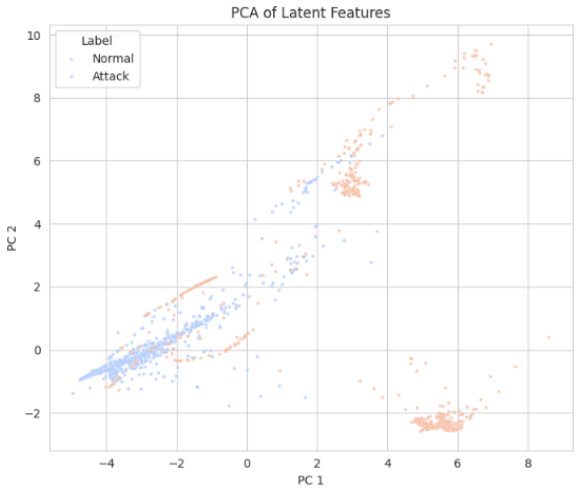

**Gambar 3.** Grafik *Training Loss Autoencoder*

Gambar 3 merupakan grafik *training loss* yang menunjukkan proses pelatihan *Autoencoder* yang dilakukan khusus pada data trafik normal selama 50 *epoch*. Terlihat penurunan nilai *loss* yang signifikan pada awal iterasi (*epoch* 0-5) dan mulai stabil (*konvergen*) pada nilai mendekati nol setelah *epoch* ke-10. Kedekatan jarak antara kurva *training* (biru) dan *validation* (oranye) mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting*. Rendahnya nilai *error* ini menegaskan bahwa *Autoencoder* berhasil mempelajari representasi fitur laten dari trafik normal dengan sangat akurat, yang menjadi prasyarat utama untuk mendeteksi anomali berdasarkan ambang batas *reconstruction error*.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

### 3.3.1 *Latent Feature*

Setelah model *Autoencoder* dilatih, langkah selanjutnya adalah mengekstrak representasi data terkompresi atau fitur laten. Guna memvalidasi efektivitas fitur tersebut dalam memisahkan pola trafik normal dan serangan, dilakukan pemetaan visual 2 dimensi yang dapat dilihat pada Gambar 4.

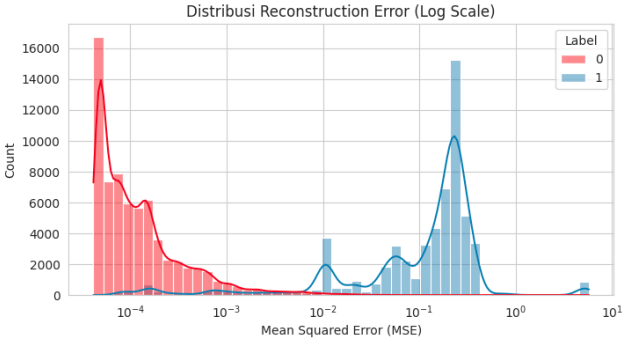



**Gambar 4.** *Latent Feature*

Gambar 4 memperlihatkan proyeksi 2 dimensi dari fitur laten yang diekstrak oleh *Autoencoder*. Visualisasi ini menegaskan bahwa transformasi fitur oleh *Autoencoder* mampu menciptakan separabilitas (keterpisahan) yang cukup jelas antara kelas normal dan serangan. Terbentuknya struktur atau klaster serangan (oranye) yang berbeda dari klaster normal (biru) membuktikan bahwa fitur baru ini membawa informasi diskriminatif yang krusial untuk dipelajari oleh algoritma Random Forest pada tahap selanjutnya.

### 3.3.2 *Reconstruction Error*

Pada penelitian ini, *reconstruction error* digunakan sebagai fitur tambahan (*input*) untuk membantu *Random Forest* membedakan antara trafik normal dan serangan dengan akurasi tinggi.




**Gambar 5.** Distribusi *Reconstruction Error*

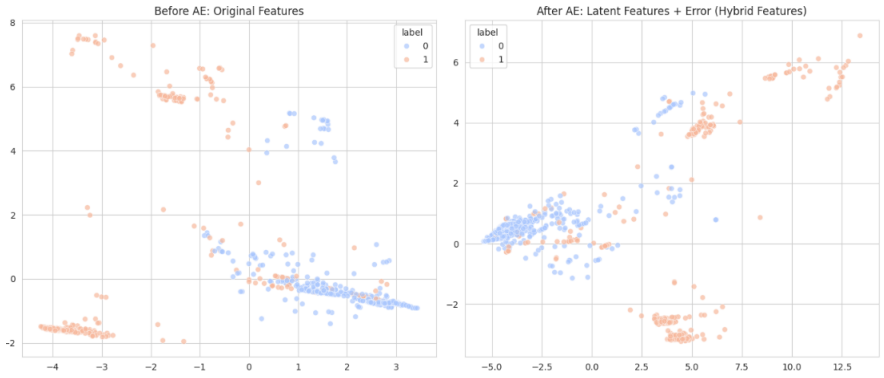
Gambar 5 memvisualisasikan distribusi nilai *Mean Squared Error* (MSE) antara trafik normal (merah) dan serangan (biru). Grafik ini memperlihatkan pemisahan kelas yang signifikan yang dihasilkan oleh model *Autoencoder*. Trafik normal memiliki kecenderungan nilai *reconstruction error* yang sangat rendah, terkonsentrasi di sisi kiri grafik (kisaran $10^{-4}$). Hal ini membuktikan bahwa *Autoencoder* berhasil mempelajari fitur laten dari aktivitas jaringan yang sah dengan sangat baik. Sebaliknya, trafik serangan menunjukkan distribusi *error* yang jauh lebih tinggi dan bergeser ke sisi kanan (kisaran $10^{-1}$), mengindikasikan ketidakmampuan model untuk merekonstruksi pola anomali yang tidak pernah dipelajari sebelumnya.

Perbedaan distribusi yang kontras ini memvalidasi efektivitas penggunaan *reconstruction error* sebagai fitur baru yang krusial. Tahapan selanjutnya adalah penggabungan dari *latent feature* dan *reconstruction error* (*Hybrid Features*).

### 3.3.3 *Hybrid Feature*

Guna memvalidasi efektivitas dari *Hybrid Features* (gabungan *latent feature* dan *reconstruction error*), dilakukan visualisasi perbandingan distribusi data. Gambar 6 di bawah ini menunjukkan transformasi signifikan pada struktur ruang fitur sebelum dan sesudah diproses oleh Autoencoder.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309



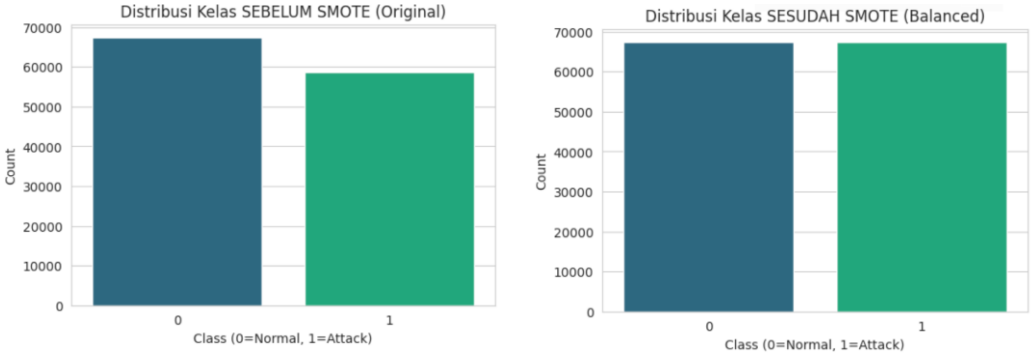

**Gambar 6.** *Hybrid Feature*

Gambar 6 menunjukkan grafik sebelah kanan (*After* AE) merepresentasikan ruang fitur hibrida (*Hybrid Feature Space*) yang terbentuk dari penggabungan fitur laten dan *reconstruction error*. Struktur linear yang terlihat pada visualisasi ini adalah manifestasi langsung dari fitur hibrida tersebut, yang membuktikan bahwa kombinasi kedua elemen fitur mampu menghasilkan pola data yang lebih distingtif dibandingkan fitur asli.

Secara spesifik, hasil visualisasi menunjukkan bahwa data trafik normal (titik biru) terkonsentrasi rapat di area kiri bawah, yang mengindikasikan nilai *reconstruction error* yang rendah dan pola laten yang konsisten. Sebaliknya, data serangan (titik oranye) terdistribusi menyebar secara diagonal ke arah kanan atas, menjauhi klaster normal. Pergeseran posisi ini menegaskan bahwa fitur tambahan berupa *reconstruction error* berfungsi efektif sebagai pembeda kuat (*discriminator*), meminimalkan area tumpang tindih (*overlap*) yang sebelumnya terlihat kompleks pada fitur asli. Keterpisahan yang lebih tegas ini secara langsung menyederhanakan kompleksitas ruang pencarian bagi algoritma *Random Forest*, sehingga memungkinkannya menarik batas keputusan (*decision boundary*) dengan akurasi yang lebih tinggi.

### 3.4 *Oversampling*

Guna mencegah terjadinya bias prediksi pada model akibat dominasi kelas mayoritas, dilakukan proses penyeimbangan data (*oversampling*) menggunakan metode SMOTE. Perbandingan distribusi sampel sebelum dan sesudah proses *oversampling* tersebut disajikan pada Gambar 7.




**Gambar 7.** SMOTE

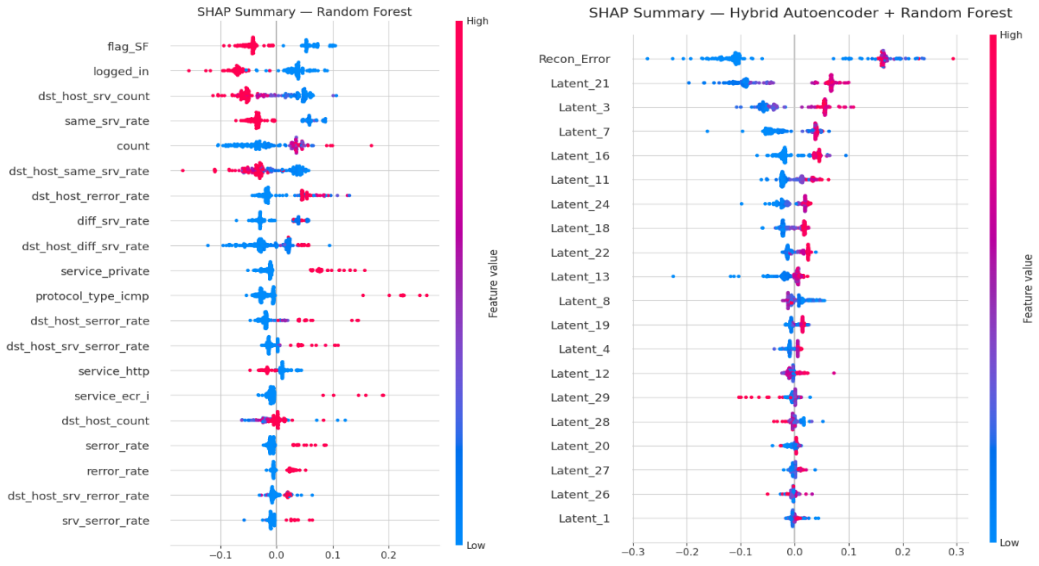
Gambar 7 merupakan visualisasi perbandingan distribusi kelas yang menunjukkan efektivitas penerapan SMOTE. Pada kondisi sebelum dilakukannya SMOTE, terlihat adanya ketidakseimbangan yang signifikan di mana kelas mayoritas mendominasi dataset. Setelah penerapan SMOTE, distribusi kelas menjadi seimbang dengan jumlah sampel yang setara antara kelas normal dan serangan. Penyeimbangan ini krusial untuk mencegah bias pada *Random Forest Classifier*, memastikan model mempelajari karakteristik kedua kelas dengan bobot yang adil, serta meningkatkan sensitivitas model dalam mendeteksi serangan yang sebelumnya merupakan kelas minoritas.

### 3.5 Perbandingan Model

#### 3.5.1 SHAP

Untuk memahami bagaimana model mengambil keputusan dalam mendeteksi intrusi, dilakukan analisis interpretabilitas menggunakan metode SHAP (*SHapley Additive exPlanations*). Analisis ini bertujuan untuk mengidentifikasi fitur-fitur yang memiliki kontribusi terbesar terhadap hasil prediksi, baik pada model *Hybrid Autoencoder* maupun model *Random Forest* standar. Visualisasi *SHAP Summary Plot* pada Gambar 8 memperlihatkan distribusi dampak setiap fitur terhadap kecenderungan model dalam memprediksi kelas serangan (*Attack*) atau normal.

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309




**Gambar 8.** SHAP

Gambar 8 adalah visualisasi dari analisis SHAP di mana terdapat perbedaan fundamental dalam cara kedua model mengambil keputusan. Model *Random Forest* bekerja sangat logis layaknya administrator jaringan, di mana ia mencurigai anomali berdasarkan status login, status koneksi (*flag*), dan volume trafik.

Di sisi lain, model *Hybrid Autoencoder* memiliki pendekatan yang lebih unik. Model ini tidak terlalu peduli pada detail teknis paket, melainkan berfokus penuh pada *reconstruction error*. Hal ini mengonfirmasi bahwa *Autoencoder* sukses berfungsi sebagai penyaring (filter) dan model ini menandai data yang 'sulit direkonstruksi' sebagai anomali, dan *Random Forest* menggunakan sinyal *error* tersebut sebagai fitur terkuat untuk memvonis adanya serangan.

### 3.5.2 *Confusion Matrix*

Berdasarkan hasil evaluasi menggunakan *Confusion Matrix*, diperoleh perbandingan kinerja antara *model Random Forest* dan model *Hybrid Autoencoder–Random Forest* sebagaimana ditunjukkan pada Tabel 5.

**Tabel 5.** *Confusion Matrix*

| Komponen | RF *Baseline* | *Hybrid* AE+RF | Penjelasan |
|---|---|---|---|
| *True Negative* | 9438 | 9501 | Lebih sedikit alarm palsu |
| *False Positive* | 273 | 210 | False Alarm berkurang |
| *False Negative* | 5327 | 4837 | Model lebih peka mendeteksi serangan |
| *True Positive* | 7506 | 7996 | Deteksi sukses bertambah |

Tabel 5 menunjukkan bahwa model *Hybrid Autoencoder Random Forest* lebih unggul dibandingkan model *Random Forest*. Keunggulan ini ditandai dengan peningkatan nilai *Recall*, yang mengindikasikan kemampuan lebih baik dalam mengidentifikasi ancaman, serta *Presisi* yang lebih optimal dalam meminimalkan tingkat peringatan palsu (*false positive*).

### 3.5.3 Evaluasi *Matrix*

Untuk menilai efektivitas metode yang diusulkan, dilakukan evaluasi kinerja dengan membandingkan model *Random Forest* standar sebagai *baseline* terhadap model hybrid yang mengombinasikan *Autoencoder* dan *Random Forest*. Perbandingan ini bertujuan untuk mengidentifikasi sejauh mana penambahan mekanisme pembelajaran representasi laten mampu meningkatkan kemampuan model dalam mendeteksi pola serangan, khususnya serangan yang tidak pernah muncul pada data pelatihan. Evaluasi dilakukan menggunakan beberapa metrik klasifikasi utama yang mencerminkan tidak hanya tingkat ketepatan prediksi, tetapi juga sensitivitas model dalam mengenali lalu lintas berbahaya. Ringkasan hasil perbandingan kinerja kedua pendekatan tersebut disajikan pada Tabel 6.

**Tabel 6.** Metrik Evaluasi

| Metrik Evaluasi | RF (Baseline) | *Hybrid* AE+RF |
|---|---|---|
| Akurasi | 75,07% | 77,61% |
| *Presisi* | 96,46% | 97,44% |
| *Recall* | 58,35% | 62,31% |
| *F1-Score* | 72,71% | 76,01% |

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

Peningkatan metrik *Recall* sebesar 3,96% memiliki implikasi krusial terhadap deteksi serangan *Zero-Day*. Pada model *Random Forest* (*baseline*), serangan jenis baru yang polanya tidak ada dalam data latih cenderung gagal dikenali dan dianggap sebagai trafik normal (*false negative*). Namun dalam metode usulan, serangan *Zero-Day* yang memiliki pola asing tersebut akan tetap menghasilkan nilai *reconstruction error* yang tinggi karena strukturnya menyimpang dari trafik normal. Tingginya nilai *error* inilah yang menjadi sinyal bagi model *Hybrid* untuk menangkap serangan tersebut, sehingga jumlah serangan yang lolos berkurang secara signifikan.

### 3.5.4 Analisis *Trade-off* Akurasi dan Sensitivitas Deteksi

Meskipun Akurasi keseluruhan tercatat pada angka 77,61% dengan *Recall* 62,31%, angka ini merepresentasikan realitas deteksi intrusi yang sesungguhnya di lingkungan produksi. Penting untuk digarisbawahi bahwa evaluasi ini menggunakan dataset KDDTest+, yang memiliki distribusi probabilitas yang berbeda secara fundamental dari data latih (KDDTrain+). Dataset uji ini memuat varian serangan spesifik seperti *'mscan'* dan *'saint'* yang sama sekali tidak diperkenalkan selama fase pelatihan (*Zero-Day Attack*). Pada kondisi ekstrim ini, model *Random Forest* mengalami penurunan performa yang wajar (*Recall* 58,35%) karena algoritma ini bekerja berdasarkan pencocokan pola (*signature-based*) dari data yang pernah dipelajari. Kegagalan *Random Forest* dalam mengenali serangan baru adalah bukti bahwa metode konvensional tidak cukup adaptif terhadap ancaman yang tidak diketahui.

Sebaliknya, kenaikan metrik *Recall* sebesar 3,96% (menjadi 62,31%) pada model *Hybrid AE + RF* bukanlah angka yang kecil. Kenaikan ini adalah indikator prestisius yang membuktikan bahwa fitur *reconstruction error* sukses berfungsi sebagai detektor anomali universal. Ketika dihadapkan pada serangan asing seperti *mscan*, *Autoencoder* gagal merekonstruksinya dengan baik sehingga menghasilkan *error* tinggi, yang kemudian ditangkap oleh *classifier* sebagai serangan. Tanpa mekanisme hibrida ini, serangan-serangan baru tersebut akan lolos begitu saja sebagai *false negative*.

## 4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, penelitian ini menyimpulkan bahwa integrasi model *Hybrid Autoencoder* dan *Random Forest* terbukti efektif dalam meningkatkan kinerja sistem deteksi intrusi jaringan. Penggunaan *Autoencoder* sebagai metode ekstraksi fitur berhasil mentransformasi data trafik jaringan yang kompleks menjadi fitur laten yang lebih representatif serta menghasilkan fitur *reconstruction error* yang berfungsi vital sebagai diskriminator anomali. Hal ini dibuktikan secara empiris melalui visualisasi ruang fitur yang menunjukkan separabilitas yang lebih tegas antara trafik normal dan serangan dibandingkan fitur asli. Secara kuantitatif, model usulan menunjukkan superioritas dibandingkan model *Random Forest* (*baseline*), dengan pencapaian Akurasi sebesar 77,61% dan F1-Score sebesar 76,01%. Peningkatan paling krusial terjadi pada metrik *Recall* yang naik sebesar 3,96%, yang mengindikasikan kemampuan model *hybrid* dalam meminimalisir tingkat serangan yang lolos (*false negative*), termasuk kemampuannya dalam mengenali serangan *Zero-Day* yang tidak ada dalam data latih. Selain itu, penerapan SMOTE berhasil menyeimbangkan distribusi kelas sehingga mencegah bias mayoritas, sementara integrasi *Explainable AI* menggunakan SHAP sukses mengatasi sifat *black-box* dari *Random Forest* dengan memberikan transparansi mengenai fitur-fitur yang memengaruhi keputusan deteksi. Dengan demikian, pendekatan ini tidak hanya menawarkan akurasi yang lebih tinggi, tetapi juga reliabilitas dan akuntabilitas yang lebih baik untuk implementasi keamanan siber modern, serta menjadi landasan strategis bagi pengembangan sistem pertahanan jaringan yang adaptif dan proaktif terhadap evolusi ancaman di masa depan.

## REFERENCES

[1] H. Sebestyen and D. E. Popescu, “A Literature Review on Security in the Internet of Things : Identifying and Analysing Critical Categories,” *Computers*, vol. 14, no. 2, p. 61, 2025, doi: 10.3390/computers14020061.
[2] E. Fazeldehkordi and T. Grønli, “A Survey of Security Architectures for Edge Computing-Based IoT,” *IoT*, vol. 3, no. 3, pp. 332–365, 2022, doi: 10.3390/iot3030019.
[3] T. R. Hadiningrum, R. Ayu, D. Talasari, and K. F. Ilham, “Survey on Risks Cyber Security in Edge Computing for The Internet of Things Understanding Cyber Attacks Threats and Mitigation,” *J. Ilm. Teknol. Inf.*, vol. 23, no. 1, pp. 29–50, 2025, doi: 10.12962/j24068535.v23i1.a1210.
[4] S. A. Alkadrie, “Keamanan Cloud Computing di Era Industri 4.0 : Systematic Literature Review,” *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 4, no. 2, pp. 1–15, 2024, doi: 10.24002/konstelasi.v4i2.10277.
[5] L. Noprizal, “BSSN: Gangguan Pusat Data Nasional Ulah Serangan Siber Ransomware,” CNN Indonesia. Accessed: Jan. 10, 2026. [Online]. Available: https://www.cnnindonesia.com/teknologi/20240624133250-192-1113404/bssn-gangguan-pusat-data-nasional-ulah-serangan-siber-ransomware
[6] BBC News Indonesia, “Pusat Data Nasional Sementara lumpuh akibat ransomware, mengapa instansi pemerintah masih rentan terhadap serangan siber?,” BBC Indonesia. Accessed: Jan. 10, 2026. [Online]. Available: https://www.bbc.com/indonesia/articles/cxee2985jrvo
[7] S. C. Arini, “Geger Data 4,7 Juta ASN Bocor dan Dijual Rp 159 Juta,” detikfinance. Accessed: Feb. 11, 2026. [Online]. Available: https://finance.detik.com/berita-ekonomi-bisnis/d-7484912/geger-data-4-7-juta-asn-bocor-dan-dijual-rp-159-juta
[8] I. Basyari and N. Harbowo, “BKN Diduga Diretas, Peretas Tawarkan Data ASN Rp 160 Juta,” kompas.id. Accessed: Jan. 10, 2026. [Online]. Available: https://www.kompas.id/artikel/bkn-diduga-diretas-peretas-tawarkan-data-asn-rp-160-juta

**Building of Informatics, Technology and Science (BITS)**
Volume 7, No 4, March 2026 Page: 2349−2360
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i4.9309

[9] Tim Redaksi, "6 Juta Data NPWP Diduga Bocor, Ada Punya Jokowi dan Gibran," CNN Indonesia. Accessed: Jan. 10, 2026. [Online]. Available: https://www.cnnindonesia.com/teknologi/20240918154543-192-1145679/6-juta-data-npwp-diduga-bocor-ada-punya-jokowi-dan-gibran

[10] A. Theodora, "Data Pajak Bocor, DJP dan Kemenkominfo Tak Boleh Lepas Tangan," kompas.id. Accessed: Jan. 10, 2026. [Online]. Available: https://www.kompas.id/artikel/data-pajak-bocor-djp-dan-kemenkominfo-tidak-boleh-lepas-tangan

[11] Kementerian Pertahanan Republik Indonesia, "Data Ancaman Siber Perbulan TA 2024," Layanan Data Terbuka Kementerian Pertahanan. Accessed: Jan. 10, 2026. [Online]. Available: https://opendata.kemhan.go.id/lihat-detail?detail=436&keyfiledetil=89ed9b5fe483d21c0b25c748fde2c63e

[12] A. Kumar and J. A. Gutierrez, "Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure," *Information*, vol. 16, no. 7, p. 515, 2025, doi: 10.3390/info16070515.

[13] D. P. Amanda and E. D. Absharina, "Implementasi AI-Powered Intrusion Detection System Untuk Mendeteksi Ancaman Keamanan Pada Big Data," *J. Sist. Inf. DAN Tek. Komput.*, vol. 10, no. 1, pp. 29–33, 2025, doi: 10.51876/simtek.v10i1.1381.

[14] A. Pinto, L. C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure," *Sensors*, vol. 23, no. 5, pp. 1–18, 2023, doi: 10.3390/s23052415.

[15] A. Nanda, H. Wahyu, R. Rahmaddeni, S. Sutisna, and R. Rinaldi, "Perbandingan Efektivitas Random Forest, SVM, dan Logistic Regression dalam Deteksi Intrusi Jaringan," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 12, no. 2, pp. 129–139, 2025, doi: 10.35957/jatisi.v12i2.10908.

[16] K. Inayah and K. Ramli, "Analisis Kinerja Intrusion Detection System Berbasis Algoritma Random Forest Menggunakan Dataset Unbalanced Honeynet BSSN," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 867–876, 2024, doi: 10.25126/jtiik.1148911.

[17] L. Mhamdi and M. M. Isa, "Securing SDN: Hybrid autoencoder-random forest for intrusion detection and attack mitigation," *J. Netw. Comput. Appl.*, vol. 225, p. 103868, 2024, doi: 10.1016/j.jnca.2024.103868.

[18] C. Wang, Y. Sun, W. Wang, H. Liu, and B. Wang, "Hybrid Intrusion Detection System Based on Combination of Random Forest and Autoencoder," *Symmetry (Basel).*, vol. 15, no. 3, pp. 1–16, 2023, doi: 10.3390/sym15030568.

[19] V. Hassija *et al.*, "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence," *Cognit. Comput.*, vol. 16, no. 1, pp. 45–74, 2024, doi: 10.1007/s12559-023-10179-8.

[20] A. I. Udofot, O. M. Oluseyi, and E. Bassey, "Explainable AI for cyber security. Improving transparency and trust in intrusion detection systems," *Int. J. Adv. Eng. Manag.*, vol. 6, no. 12, pp. 229–240, 2024, doi: 10.35629/5252-0612229240.

[21] F. H. Saputra *et al.*, "Enhancing Intrusion Detection Using Random Forest and SMOTE on the NSL-KDD Dataset," *J. Syst. Comput. Eng.*, vol. 6, no. 3, pp. 240–247, 2025, doi: 10.61628/jsce.v6i3.2056.

[22] Y. Song and S. Hyun, "Analysis of Autoencoders for Network Intrusion Detection," *Sensors*, vol. 21, no. 13, p. 4294, 2021, doi: 10.3390/s21134294.

[23] A. Fadhil and H. Alharan, "Enhancing Intrusion Detection with Autoencoder Based Classifier and Statistical Feature Selection," *Wasit J. Pure Sci.*, vol. 2, no. 4, pp. 97–105, 2023, doi: 10.31185/wjps.257.

[24] S. H. Abbas, W. A. K. Naser, and A. A. Kadhim, "Subject review: Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)," *Glob. J. Eng. Technol. Adv.*, vol. 14, no. 2, pp. 155–158, 2023, doi: 10.30574/gjeta.2023.14.2.0031.

[25] Z. Umari and J. Supardi, "Deteksi Anomali Sinyal Vibrasi pada Mesin Industri Menggunakan Autoencoder di PT. Pusri Palembang," *J. Pendidik. dan Teknol. Indones.*, vol. 4, no. 12, pp. 737–746, 2024, doi: 10.52436/1.jpti.553.

[26] C. Molnar, *Interpretable Machine Learning A Guide for Making Black Box Models Explainable*, 2nd ed. Munich: Leanpub, 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[27] R. Genuer and J.-M. Poggi, *Use R ! Random Forests with R*. Cham: Springer, 2020. doi: 10.1007/978-3-030-56485-8.