
ANALISIS KOMPARASI ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE UNTUK DETEKSI INTRUSI JARINGAN

**Hasanal Fachri Satia Simbolon^{1*}, Ade Linhar P²,
Rafi Septiawan Putra³, Fahmi Izhari⁴**

^{1,2,3,4}UIN Syekh Ali Hasan Ahmad Addary, Padangsidempuan

¹hasanalfachri@uinsyahada.ac.id

Abstrak

Meningkatnya kompleksitas serangan siber menuntut adanya sistem keamanan jaringan yang adaptif dan efisien. Intrusion Detection System (IDS) tradisional seringkali memiliki keterbatasan dalam mengenali pola serangan baru. Penelitian ini bertujuan untuk mengevaluasi kinerja dua algoritma Machine Learning, yaitu Random Forest (RF) dan Support Vector Machine (SVM), dalam mengklasifikasikan trafik jaringan normal dan serangan. Eksperimen dilakukan menggunakan dataset NSL-KDD dengan melibatkan seluruh 41 fitur melalui tahapan preprocessing, normalisasi, dan validasi data dengan rasio 80:20. Hasil pengujian menunjukkan bahwa algoritma Random Forest mengungguli SVM dengan tingkat akurasi mencapai 99.78%, presisi 1.00, dan recall 1.00. Sebaliknya, SVM mencatatkan akurasi sebesar 99.03%. Selain unggul dalam akurasi, Random Forest terbukti lebih efisien dengan waktu pelatihan (training time) rata-rata 3.72 detik, hampir dua kali lebih cepat dibandingkan SVM yang membutuhkan 6.61 detik. Berdasarkan hasil tersebut, Random Forest direkomendasikan sebagai algoritma yang lebih efektif untuk implementasi IDS pada lingkungan yang membutuhkan respons waktu nyata (real-time).

Kata Kunci : Keamanan Jaringan; Intrusion Detection System; Machine Learning; Random Forest; SVM; NSL-KDD.

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

1. Pendahuluan

Keamanan jaringan (Network Security) telah menjadi isu krusial seiring dengan pertumbuhan eksponensial dalam volume lalu lintas data global. Metode pertahanan tradisional seperti firewall dan antivirus berbasis tanda tangan (signature-based) semakin tidak memadai untuk menangani serangan siber yang kompleks dan dinamis, seperti zero-day attacks. Oleh karena itu, pendekatan berbasis Anomaly Detection menggunakan teknik Machine Learning menjadi solusi yang mendesak karena kemampuannya mendeteksi pola serangan yang belum dikenal sebelumnya (Buczak & Guven, 2016; Garcia-Teodoro et al., 2009).

Meskipun menjanjikan, penerapan Machine Learning dalam deteksi intrusi menghadapi tantangan besar terkait perbedaan karakteristik antara lingkungan pelatihan dan dunia nyata (Sommer & Paxson, 2010). Tantangan utama lainnya adalah ketersediaan dataset yang valid. Dataset klasik seperti KDD Cup 99 diketahui memiliki banyak kelemahan, termasuk redundansi data yang masif yang dapat menyebabkan bias pada hasil evaluasi algoritma (Stolfo et al., 2000). Sebagai respons terhadap masalah ini, Tavallaee et al. (2009) memperkenalkan dataset NSL-KDD, yang telah dibersihkan dan kini menjadi standar evaluasi yang lebih adil dalam penelitian keamanan siber (Revathi & Malathi, 2013).

Penelitian ini bertujuan untuk mengevaluasi efektivitas dua algoritma populer dalam mendeteksi intrusi pada dataset NSL-KDD, yaitu Random Forest dan Support Vector Machine (SVM). Random Forest, yang diperkenalkan oleh Breiman (2001), bekerja dengan prinsip ensemble learning yang menggabungkan banyak pohon keputusan untuk menjaga stabilitas prediksi. Sementara itu, SVM yang dikembangkan oleh Cortes dan Vapnik (1995) menggunakan pendekatan pencarian hyperplane optimal untuk memisahkan kelas data.

Meskipun kedua algoritma ini telah terbukti andal secara teoritis, implementasi praktis pada lingkungan jaringan membutuhkan

keseimbangan antara akurasi deteksi dan efisiensi waktu komputasi (computational cost). Penelitian ini akan membandingkan kedua aspek tersebut secara komprehensif.

Buczak dan Guven (2016) dalam survei komprehensifnya menyatakan bahwa metode Data Mining dan Machine Learning adalah kunci utama untuk membangun sistem deteksi intrusi yang adaptif. Namun, mereka juga mencatat bahwa kompleksitas algoritma seringkali menjadi penghalang untuk implementasi waktu nyata (real-time).

Studi spesifik pada dataset NSL-KDD dilakukan oleh Dhanabal dan Shantharajah (2015), yang menganalisis kinerja algoritma J48 dan SVM. Temuan mereka menunjukkan bahwa meskipun SVM memiliki tingkat akurasi yang tinggi, algoritma ini membutuhkan waktu pelatihan (training time) yang jauh lebih lama dibandingkan algoritma berbasis pohon keputusan. Hal ini menjadi indikasi awal bahwa SVM mungkin kurang efisien untuk dataset berskala besar.

Penelitian lain oleh Belavagi dan Muniyal (2016) membandingkan empat algoritma: Logistic Regression, Gaussian Naive Bayes, SVM, dan Random Forest. Hasil eksperimen mereka menempatkan Random Forest sebagai algoritma dengan akurasi tertinggi dalam mengklasifikasikan serangan. Meski demikian, penelitian tersebut berfokus utama pada metrik akurasi semata dan kurang mendalami aspek efisiensi waktu pemrosesan secara mendetail.

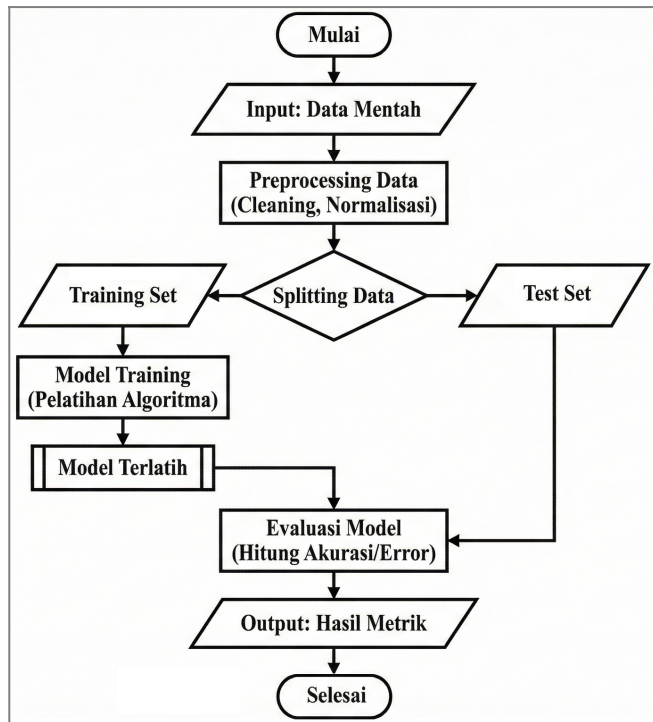
Berbeda dengan penelitian-penelitian di atas (Dhanabal & Shantharajah, 2015; Belavagi & Muniyal, 2016) penelitian ini tidak hanya berfokus pada pencapaian akurasi tertinggi. Penelitian ini secara spesifik melakukan analisis komparatif head-to-head antara Random Forest dan SVM dengan melibatkan pengukuran waktu komputasi secara presisi. Selain itu, penelitian ini menggunakan seluruh 41 fitur dataset NSL-KDD tanpa reduksi dimensi untuk menguji ketahanan (robustness) algoritma terhadap data yang kompleks. Tujuannya adalah merekomendasikan algoritma yang paling layak untuk sistem keamanan yang responsif.

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

2. Metode Penelitian

Alur Penelitian

Untuk mencapai tujuan penelitian, penulis menyusun tahapan sistematis yang digambarkan dalam alur kerja (framework). Tahapan ini meliputi pengumpulan data, pra-pemrosesan data, pembagian data, pelatihan model, dan evaluasi kinerja.



Gambar 1. Diagram Alir Penelitian

Dataset Penelitian

Data yang digunakan dalam penelitian ini bersumber dari dataset NSL-KDD, yang merupakan dataset standar de facto untuk evaluasi sistem deteksi intrusi jaringan. Secara spesifik, penelitian ini menggunakan varian KDDTrain+_20Percent.

Tabel 1. Representasi dataset dengan menampilkan sebagian atribut

Protocol_Type	Service	Src_Bytes	Dst_Bytes	Flag	Label
tcp	ftp_data	491	0	SF	normal
udp	other	146	0	SF	normal
tcp	private	0	0	S0	neptune
tcp	http	232	8153	SF	normal
tcp	http	199	420	SF	normal
tcp	private	0	0	REJ	neptune
tcp	private	0	0	S0	neptune
tcp	private	0	0	S0	neptune
tcp	remote_job	0	0	S0	neptune
tcp	private	0	0	S0	neptune

Karakteristik dataset yang digunakan adalah sebagai berikut:

- 1) Jumlah Data: Total data berjumlah 25.192 baris rekaman lalu lintas jaringan.
- 2) Fitur (Atribut): Terdiri dari 41 fitur yang mencakup fitur dasar (basic features), fitur konten (content features), dan fitur lalu lintas (traffic features), atribut yang ada di tabel 1 adalah atribut sampel yang diambil dari dataset.
- 3) Label (Target): Kolom label berisi kategori trafik yang dikelompokkan menjadi dua kelas utama (Binary Classification), yaitu:
 - a) Normal: Trafik jaringan yang aman.
 - b) Attack: Trafik yang terindikasi sebagai serangan (termasuk DoS, Probe, R2L, dan U2R). Neptune yang ada di dalam tabel 1 adalah jenis label serangan yang termasuk di dalam jenis serangan DoS.

Data Preprocessing

Sebelum data dimasukkan ke dalam algoritma, dilakukan tahapan pra-pemrosesan agar data dapat dibaca oleh mesin dan

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

menghasilkan model yang optimal. Tahapan ini meliputi Transformasi Data (Encoding) dan Normalisasi Data (Feature Scaling).

Transformasi Data (Encoding)

Dataset NSL-KDD memiliki beberapa fitur bertipe kategorikal (teks), seperti `protocol_type` (tcp, udp, icmp), `service` (http, ftp, dll), dan `flag` (SF, S0). Karena algoritma Machine Learning hanya dapat memproses input berupa angka, dilakukan proses transformasi menggunakan teknik Label Encoding.

- Contoh: tcp diubah menjadi 0, udp menjadi 1, dst.

Normalisasi Data (Feature Scaling)

Fitur-fitur dalam dataset memiliki rentang nilai yang sangat bervariasi. Sebagai contoh, fitur `duration` memiliki rentang 0-50.000, sedangkan `num_failed_logins` hanya 0-5. Perbedaan skala ini dapat menyebabkan bias, terutama pada algoritma SVM yang sensitif terhadap jarak antar data. Oleh karena itu, diterapkan teknik Standard Scaler (Z-score normalization) untuk mengubah distribusi nilai setiap fitur agar memiliki rata-rata 0 dan variansi 1. Rumus yang digunakan adalah:

$$z = \frac{x - \mu}{\sigma} \dots\dots\dots (1)$$

Dimana x adalah nilai asli, μ adalah rata-rata (mean), dan σ adalah standar deviasi.

Skenario Eksperimen

Pembagian Data (Data Splitting)

Untuk mengevaluasi model secara objektif, dataset dibagi menjadi dua bagian menggunakan metode Hold-out Validation dengan rasio 80:20:

- Data Latih (Training Set): Sebanyak 80% data (20.153 data) digunakan untuk melatih algoritma agar mengenali pola serangan.

• Data Uji (Testing Set): Sebanyak 20% data (5.039 data) digunakan untuk menguji keandalan model dalam memprediksi data baru yang belum pernah dilihat sebelumnya.

Konfigurasi Algoritma

Dua algoritma klasifikasi dibangun menggunakan pustaka Scikit-Learn dengan konfigurasi sebagai berikut:

Random Forest (RF)

Menggunakan parameter $n_estimators=100$, yang berarti algoritma akan membangun 100 pohon keputusan (decision trees) untuk melakukan prediksi kolektif. Random Forest, yang dikembangkan oleh Breiman (2001), adalah metode ensemble learning yang beroperasi dengan membangun banyak pohon keputusan (decision trees) pada saat pelatihan. Prinsip dasar dari algoritma ini adalah Bagging (Bootstrap Aggregating). Cara kerja Random Forest dalam menentukan klasifikasi adalah sebagai berikut:

- (1) Bootstrapping: Algoritma mengambil sampel acak dari dataset dengan pengembalian (replacement) untuk membentuk beberapa subset data yang berbeda.
- (2) Konstruksi Pohon: Untuk setiap subset, sebuah pohon keputusan dibangun. Pada setiap pemecahan node (node splitting), algoritma memilih fitur terbaik dari sekumpulan fitur acak, bukan dari seluruh fitur yang ada. Hal ini bertujuan untuk mengurangi korelasi antar pohon.
- (3) Voting (Agregasi): Untuk klasifikasi, prediksi akhir ditentukan berdasarkan majority voting (suara terbanyak). Jika terdapat N pohon, dan k_i adalah prediksi dari pohon ke- i , maka prediksi akhir Y adalah modus dari kumpulan prediksi tersebut:

$$Y = \text{mode}\{k_1, k_2, \dots, k_N\} \dots\dots\dots (2)$$

Keunggulan utama mekanisme ini adalah kemampuannya mereduksi varians tanpa meningkatkan bias secara signifikan, sehingga model menjadi lebih tahan terhadap overfitting dibandingkan pohon keputusan tunggal.

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

Support Vector Machine (SVM)

Menggunakan kernel RBF (Radial Basis Function) yang efektif untuk menangani data dengan batas keputusan non-linear. Support Vector Machine (SVM) adalah algoritma supervised learning yang bekerja dengan prinsip pencarian hyperplane (bidang pemisah) optimal yang memisahkan dua kelas data dengan margin terbesar (Cortes & Vapnik, 1995). Proses penentuan klasifikasi pada SVM melibatkan langkah-langkah berikut:

Pencarian Hyperplane

SVM mencari garis (pada 2D) atau bidang (pada dimensi tinggi) $w \cdot x + b = 0$ yang memisahkan kelas positif (serangan) dan negatif (normal).

Maksimasi Margin

Algoritma mengoptimalkan jarak antara hyperplane dengan titik data terdekat dari masing-masing kelas, yang disebut sebagai Support Vectors. Secara matematis, tujuannya adalah meminimalkan $\|w\|^2$ dengan batasan tertentu.

Kernel Trick

Karena data lalu lintas jaringan seringkali tidak dapat dipisahkan secara linear (non-linearly separable), penelitian ini menggunakan fungsi Kernel RBF (Radial Basis Function). Kernel ini memetakan data asli ke ruang dimensi yang lebih tinggi di mana pemisahan linear menjadi mungkin. Fungsi Kernel RBF didefinisikan sebagai:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \dots\dots\dots (3)$$

Dimana γ adalah parameter yang menentukan pengaruh dari setiap sampel pelatihan.

Metrik Evaluasi

Kinerja model diukur menggunakan parameter Confusion Matrix. Metrik utama yang menjadi fokus perbandingan adalah:

1. Akurasi (Accuracy): Rasio prediksi benar (positif dan negatif) terhadap keseluruhan data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (4)$$

2. Presisi (Precision): Tingkat ketepatan prediksi positif.

3. Recall (Sensitivity): Kemampuan model mendeteksi seluruh serangan yang ada.

4. Waktu Komputasi: Waktu yang dibutuhkan model untuk menyelesaikan proses pelatihan (training time) dalam satuan detik.

3. Hasil dan Pembahasan

Hasil Eksperimen

Pada bagian ini, dipaparkan hasil pengujian kinerja algoritma Random Forest (RF) dan Support Vector Machine (SVM) dalam mengklasifikasikan trafik jaringan pada dataset NSL-KDD. Eksperimen dilakukan menggunakan skenario pembagian data 80% data latih dan 20% data uji.

Kinerja Klasifikasi

Berdasarkan eksperimen yang telah dilakukan, ringkasan performa kedua algoritma disajikan dalam Tabel 2 berikut. Evaluasi dilakukan berdasarkan empat metrik utama: Akurasi, Presisi, Recall, dan F1-Score.

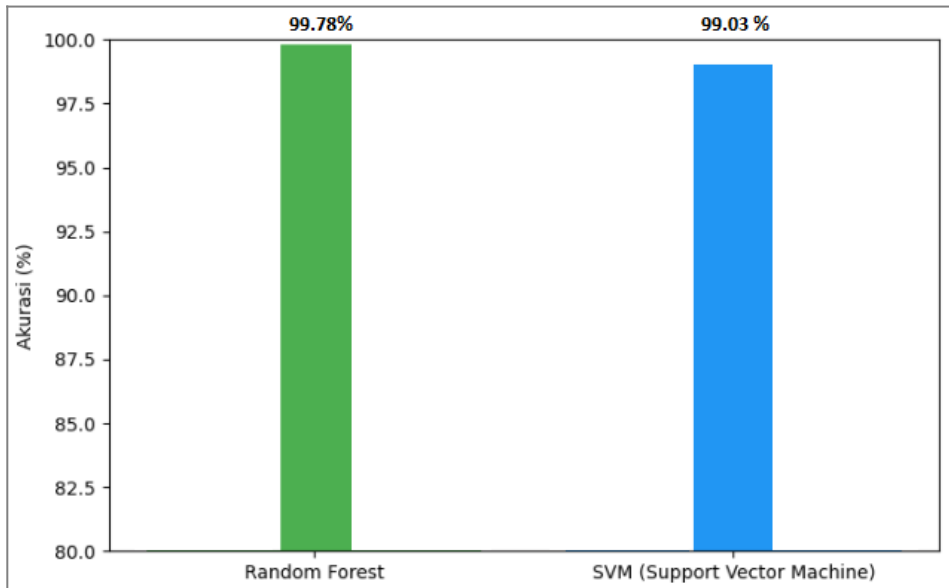
Tabel 2. Perbandingan Kinerja Algoritma

Algoritma	Accuracy	Precision	Recall	F1-Score
Random Forest	99.78%	1.00	1.00	1.00
Support Vector Machine (SVM)	99.03%	0.99	0.99	0.99

Data pada Tabel 2 menunjukkan bahwa Random Forest mencatatkan kinerja yang lebih unggul dibandingkan SVM di seluruh metrik pengujian. Random Forest mencapai akurasi hampir sempurna

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

sebesar 99.78%, sedangkan SVM berada sedikit di bawahnya dengan akurasi 99.03%. Meskipun selisih akurasi terlihat kecil (0.75%), dalam konteks keamanan jaringan dengan jutaan paket data per detik, selisih ini sangat signifikan dalam meminimalkan celah keamanan.



Gambar 2. Grafik Perbandingan Akurasi Random Forest vs SVM

Efisiensi Waktu Komputasi

Selain akurasi, faktor krusial dalam implementasi IDS adalah kecepatan pemrosesan (Computational Cost). Tabel 3 memperlihatkan perbandingan waktu pelatihan (training time) yang dibutuhkan kedua algoritma.

Tabel 3. Perbandingan Waktu Pelatihan (Training Time)

Algoritma	Waktu Pelatihan (Detik)
Random Forest	3.72 s
Support Vector Machine (SVM)	6.61 s

Hasil pengukuran waktu menunjukkan bahwa Random Forest bekerja hampir 2x lebih cepat dibandingkan SVM. Random Forest hanya membutuhkan waktu 3.72 detik untuk mempelajari pola dari 20.153 data latih, sedangkan SVM membutuhkan waktu 6.61 detik.

Analisis Keunggulan Random Forest

Hasil eksperimen membuktikan bahwa Random Forest lebih efektif dibandingkan SVM untuk dataset NSL-KDD. Keunggulan ini dapat dianalisis dari karakteristik algoritma itu sendiri:

(1) Kemampuan Ensemble Learning

Random Forest bekerja dengan membangun banyak pohon keputusan (Decision Trees) secara acak dan menggabungkan hasilnya (voting). Metode ini membuat Random Forest sangat tangguh (robust) terhadap noise dan variasi data yang ada pada 41 fitur dataset NSL-KDD.

(2) Penanganan Dimensi Tinggi

Dataset NSL-KDD memiliki fitur yang kompleks (kategorikal dan numerik). Random Forest mampu menyeleksi fitur-fitur penting secara internal saat pembentukan pohon, sehingga fitur yang kurang relevan tidak terlalu mengganggu hasil prediksi.

(3) Stabilitas

Nilai Presisi dan Recall Random Forest yang mencapai angka 1.00 (sempurna) mengindikasikan bahwa algoritma ini sangat minim menghasilkan False Positive (alarm palsu) maupun False Negative (serangan yang lolos). Hal ini sangat vital bagi administrator jaringan agar tidak terganggu oleh peringatan palsu.

Analisis Keterbatasan SVM

Meskipun SVM masih menunjukkan performa yang sangat baik (99.03%), algoritma ini tertinggal dari segi efisiensi waktu. Hal ini disebabkan oleh kompleksitas komputasi SVM dalam mencari hyperplane optimal, terutama ketika menggunakan kernel RBF (Radial Basis Function).

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

Pada dataset dengan jumlah fitur banyak (41 fitur) dan jumlah baris data puluhan ribu, proses pemetaan data ke dimensi tinggi oleh kernel SVM membutuhkan sumber daya komputasi yang lebih besar dibandingkan proses pembentukan pohon pada Random Forest. Oleh karena itu, SVM membutuhkan waktu pelatihan yang lebih lama (6.61 detik).

Implikasi Penelitian

Berdasarkan hasil komparasi di atas, dapat disimpulkan bahwa untuk kebutuhan sistem deteksi intrusi (IDS) yang menuntut akurasi tinggi sekaligus respons cepat (real-time), algoritma Random Forest adalah pilihan yang lebih direkomendasikan dibandingkan SVM. Efisiensi waktu yang dimiliki Random Forest memungkinkan sistem untuk melakukan pembaruan model (retraining) secara berkala dengan lebih cepat tanpa membebani kinerja server.

4. Kesimpulan dan Saran

Kesimpulan

Berdasarkan hasil eksperimen dan analisis perbandingan kinerja algoritma Random Forest dan Support Vector Machine (SVM) dalam mendeteksi intrusi jaringan menggunakan dataset NSL-KDD, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Kinerja Klasifikasi

Kedua algoritma Machine Learning yang diuji terbukti sangat andal dalam membedakan trafik normal dan serangan, dengan tingkat akurasi di atas 99%. Hal ini menunjukkan bahwa metode pembelajaran mesin sangat efektif untuk diterapkan sebagai mesin inti pada Intrusion Detection System (IDS).

2. Algoritma Terbaik

Algoritma Random Forest menunjukkan kinerja yang lebih unggul dibandingkan SVM. Random Forest mencatatkan akurasi sebesar

99.78%, nilai Presisi 1.00, dan Recall 1.00. Sementara itu, SVM memiliki akurasi sedikit lebih rendah yaitu 99.03%.

3. Efisiensi Waktu

Dari segi biaya komputasi, Random Forest terbukti jauh lebih efisien dengan waktu pelatihan (training time) rata-rata 3.72 detik, sedangkan SVM membutuhkan waktu 6.61 detik. Kecepatan ini menjadikan Random Forest kandidat yang lebih baik untuk implementasi pada sistem jaringan yang membutuhkan respons cepat (real-time).

Secara keseluruhan, penelitian ini menyimpulkan bahwa algoritma Random Forest adalah pilihan yang lebih direkomendasikan dibandingkan SVM untuk kasus deteksi intrusi pada dataset NSL-KDD karena unggul baik dari segi ketepatan prediksi maupun kecepatan pemrosesan.

Saran

Penelitian ini masih memiliki ruang untuk pengembangan lebih lanjut. Disarankan untuk penelitian selanjutnya menguji algoritma pada dataset yang lebih modern seperti CIC-IDS2017 dan mengeksplorasi metode Deep Learning untuk melihat potensi peningkatan akurasi lebih lanjut.

Daftar Pustaka

- Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, 89, 117–123.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.

Analisis Komparasi Algoritma Random Forest dan Support Vector Machine untuk Deteksi Intrusi Jaringan

- Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6), 446–452.
- Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2), 18–28.
- Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, 2(12), 1848–1853.
- Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316.
- Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. *DARPA Information Survivability Conference and Exposition*, 2, 130–144.
- Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 1–6.