e-ISSN : 2722-4368

Analisis Log Server dengan Data Mining untuk Deteksi Aktifitas Malicious

Muhammad Anis Al Hilmi[1*], Kurnia Adi Cahyanto[2], Azhar Al Afghani[1], Badrudin Hadibrata[1]**
[1]Program Studi Informatika, Universitas Swadaya Gunung Jati, Cirebon, Jawa Barat, Indonesia
[2] Program Doktor Sistem Informasi, Universitas Diponegoro, Semarang, Jawa Tengah, Indonesia
Email: muhammadanisalhilmi.dosen@ugj.ac.id*, kelixo@gmail.com**

Abstrak. *Keamanan server web menjadi perhatian utama seiring meningkatnya ancaman siber. Setiap interaksi pengguna terhadap aplikasi web terekam dalam log server yang menyimpan informasi berharga mengenai alamat IP, metode request, status respons, dan ukuran data. Penelitian ini memanfaatkan data log server periode Januari–Juli 2019 dari sebuah institusi pendidikan untuk mendeteksi aktivitas malicious menggunakan pendekatan data mining. Setelah melalui tahap praproses dan pelabelan berbasis aturan (rule-based labeling) dengan tiga kelas Aman, Dicurigai, dan Bahaya dataset direduksi dimensinya menggunakan Linear Discriminant Analysis (LDA) sebelum diklasifikasikan dengan lima algoritma: SVM-RBF, SVM-Linear, SVM-Polynomial, K-NN via GridSearch, dan Decision Tree. Hasil pengujian menunjukkan bahwa SVM-RBF memberikan performa paling stabil dengan akurasi training 88% dan testing 86%. Namun, ketidakseimbangan distribusi kelas mempengaruhi nilai recall pada kelas tertentu. Penelitian ini menegaskan efektivitas kombinasi LDA dan SVM-RBF sebagai fondasi sistem deteksi intrusi berbasis log server, sekaligus membuka peluang pengembangan lebih lanjut melalui teknik penyeimbangan data dan eksplorasi fitur tambahan.*

Kata Kunci :log server, data mining, SVM, deteksi intrusi, aktivitas malicious

Abstract. *Web server security is a primary concern amid the rising wave of cyber threats. Every user interaction with a web application is recorded in server logs, which contain valuable information including IP addresses, request methods, response status codes, and data sizes. This study leverages server log data from January to July 2019 collected from an educational institution to detect malicious activities using a data mining approach. After preprocessing and rule-based labeling into three classes Safe, Suspicious, and Dangerous dimensionality reduction was applied via Linear Discriminant Analysis (LDA) before classification using five algorithms: SVM-RBF, SVM-Linear, SVM-Polynomial, K-NN via GridSearch, and Decision Tree. Results show that SVM-RBF delivers the most stable performance, achieving a training accuracy of 88% and testing accuracy of 86%. However, class imbalance affects recall scores for certain categories. This study confirms the effectiveness of combining LDA and SVM-RBF as a basis for log-based intrusion detection systems, while also highlighting the need for further development through data balancing techniques and additional feature engineering.*

Keyword : log server, data mining, SVM,intrusion detection, malicious activity

PENDAHULUAN

Ancaman keamanan siber terhadap infrastruktur web terus berkembang seiring dengan meningkatnya volume dan kompleksitas lalu lintas data di internet. Berbagai laporan keamanan global menunjukkan tren peningkatan serangan berbasis web yang signifikan, mulai dari injeksi kode berbahaya, eksploitasi celah unggah berkas, hingga serangan brute force terhadap mekanisme autentikasi sistem [1][2]. Dalam konteks ini, log server menjadi artefak digital yang sangat bernilai karena merekam jejak seluruh aktivitas pengguna secara kronologis dan terstruktur, sehingga dapat dimanfaatkan sebagai sumber intelijen keamanan yang andal [3]. Salah satu vektor serangan yang paling konsisten dimanfaatkan oleh aktor ancaman adalah mekanisme unggah berkas pada aplikasi web. Berkas berbahaya seperti webshell yang berhasil diunggah ke server memberikan akses persisten kepada penyerang untuk mengeksekusi perintah, mengekstrak data, maupun menjadikan server sebagai titik pijak serangan lanjutan [4][5]. Karena seluruh aktivitas ini terekam dalam log server, analisis terhadap data log merupakan pendekatan deteksi yang sangat relevan secara praktis.

Penggunaan teknik data mining dalam analisis log server terbukti mampu mengotomasi proses identifikasi pola intrusi yang secara manual tidak efisien dilakukan pada volume data besar [6][7].

Berbagai algoritma klasifikasi telah dieksplorasi untuk keperluan deteksi intrusi berbasis log. Di antara pendekatan yang telah diuji, Support Vector Machine (SVM) secara konsisten dilaporkan memberikan akurasi kompetitif untuk klasifikasi data teks dan log server [5][8]. Namun, perbandingan sistematis antara berbagai keluarga algoritma mulai dari SVM dengan berbagai kernel, pendekatan berbasis jarak seperti K-NN, hingga model berbasis partisi seperti Decision Tree pada dataset log server produksi nyata masih sangat jarang dilakukan [9]. Sebagian besar studi komparatif yang ada hanya berfokus pada dataset benchmark seperti KDD Cup 99 atau NSL-KDD yang tidak mencerminkan karakteristik data log web server aktual, termasuk pola serangan berbasis webshell yang spesifik pada konteks institusi tertentu [10]. Celah inilah yang menjadi titik berangkat penelitian ini. Studi Cahyanto et al. [5] sebelumnya telah meletakkan fondasi penting dengan menunjukkan efektivitas kombinasi LDA dan SVM-RBF pada dataset log server nyata dari sebuah institusi pendidikan, namun evaluasinya terbatas pada dua konfigurasi model saja. Pertanyaan yang belum terjawab adalah: apakah keunggulan SVM-RBF tersebut bersifat konsisten ketika dibandingkan dengan algoritma-algoritma lain dari keluarga yang berbeda secara metodologis? Dan apakah karakteristik inheren data log server khususnya ketidakseimbangan kelas dan non-linearitas distribusi fitur berdampak secara berbeda terhadap masing-masing keluarga algoritma tersebut [11].

Penelitian ini menjawab pertanyaan tersebut dengan melakukan studi komparatif sistematis yang mengevaluasi lima algoritma dari tiga keluarga berbeda berbasis kernel (SVM-RBF, SVM-Linear, SVM-Polynomial), berbasis jarak (K-NN via GridSearch), dan berbasis partisi (Decision Tree) secara bersamaan pada dataset log server produksi nyata yang sama. Dengan demikian, kontribusi penelitian ini bukan sekadar penambahan varian algoritma, melainkan pemetaan komprehensif tentang karakteristik kesesuaian (fitness) masing-masing keluarga algoritma terhadap sifat inheren data log server. Hasil penelitian ini diharapkan menjadi panduan praktis bagi peneliti maupun praktisi keamanan siber dalam merancang sistem deteksi intrusi berbasis log yang lebih efektif dan tepat sasaran [12].

TINJAUAN PUSTAKA

Sejumlah penelitian telah mengeksplorasi penggunaan teknik data mining dan machine learning untuk analisis log server dan deteksi aktivitas malicious. Tinjauan berikut mengklasifikasikan pendekatan-pendekatan tersebut ke dalam empat kelompok: metode klasifikasi tradisional, metode clustering dan anomali, metode berbasis deep learning, serta platform dan sistem terpadu. Khaerani dan Handoko [4] mendemonstrasikan penerapan algoritma C4.5 untuk mengklasifikasikan jenis serangan pada Intrusion Detection System (IDS). Hasilnya membuktikan bahwa pohon keputusan berbasis aturan mampu mengidentifikasi pola anomali dari data log secara efisien dan interpretable, meskipun rentan terhadap overfitting pada data dengan distribusi kelas yang tidak seimbang. Cahyanto et al. [5] menerapkan SVM berbasis LDA pada dataset log server nyata dari sebuah institusi pendidikan. Penelitian tersebut menunjukkan bahwa kombinasi LDA sebagai pereduksi dimensi dan SVM-RBF sebagai pengklasifikasi mencapai akurasi testing 89,9% dengan waktu komputasi yang jauh lebih singkat dibandingkan SVM tanpa reduksi fitur. Studi ini menjadi fondasi langsung bagi penelitian yang sedang dilaporkan. Pamuji [13] mengusulkan algoritma Naive Bayes untuk memprediksi otorisasi pengguna sistem berkas, menyoroti keunggulan pendekatan probabilistik dalam skenario di mana distribusi prior kelas dapat diestimasi. Senada dengan itu, Injadat et al. [8] mengembangkan kerangka multi-tahap berbasis machine learning untuk deteksi intrusi jaringan, di mana SVM dikombinasikan dengan teknik seleksi fitur menghasilkan F1-score yang kompetitif pada dataset NSL-KDD. Namun, penulis mengakui bahwa performa pada dataset benchmark belum tentu dapat digeneralisasi ke data log produksi nyata.

Paramitha et al. [14] menggunakan algoritma Fuzzy C-Means (FCM) untuk mengelompokkan alert pada log IDS Snort ke dalam kategori tingkat risiko dari rendah hingga kritis. Pendekatan unsupervised ini berguna untuk eksplorasi awal data tanpa label, meskipun memerlukan interpretasi pakar untuk validasi kluster yang dihasilkan. Saputro et al. [6] mengimplementasikan Isolation Forest untuk mendeteksi anomali pada server NGINX. Keunggulan utama metode ini adalah kemampuannya bekerja tanpa data berlabel dan efisiensinya pada data berdimensi tinggi. Xu et al. [15] memperluas pendekatan anomali detection dengan mengusulkan variational autoencoder berbasis log untuk mendeteksi intrusi secara tidak terawasi pada lingkungan IoT, yang menunjukkan potensi besar metode generatif untuk mendeteksi serangan yang belum pernah dilihat sebelumnya (zero-day). Promodya et al. [7] mengembangkan sistem deteksi intrusi real-time berbasis deep neural network yang mampu mengklasifikasikan serangan jaringan dengan akurasi tinggi. Sementara itu, Chourasiya et al. [1] mengusulkan kombinasi LSTM dan Transformer untuk analisis log server secara forensik, memanfaatkan kemampuan pemodelan sekuensial LSTM dan mekanisme perhatian Transformer untuk menangkap dependensi jangka panjang dalam urutan log. Pendekatan deep learning ini umumnya membutuhkan volume data berlabel yang jauh lebih besar dibandingkan metode klasik, yang menjadi kendala dalam konteks dataset log server dengan kelas Bahaya yang langka. Landauer et al. [16] secara khusus mengkaji pentingnya kualitas dan representativitas dataset log untuk pelatihan model IDS, menyimpulkan bahwa dataset yang dibuat secara sintetis atau dari lingkungan terkontrol sering kali tidak mencerminkan distribusi serangan di lingkungan produksi nyata. Temuan ini memperkuat relevansi penggunaan data log produksi aktual seperti yang dilakukan dalam penelitian ini. Nova et al. [17] mengkaji platform Wazuh sebagai sistem manajemen log event sekaligus detektor keamanan terhadap serangan DoS, menggambarkan integrasi antara pengelolaan log real-time dan analisis keamanan berbasis aturan. Miani et al. [18] melakukan survei komprehensif terhadap sistem deteksi intrusi berbasis data stream, menyimpulkan bahwa tantangan terbesar dalam implementasi IDS nyata adalah latensi deteksi dan adaptasi terhadap konsep drift, yaitu perubahan pola serangan dari waktu ke waktu yang menyebabkan model yang dilatih pada data lama menjadi kurang efektif.

Berdasarkan tinjauan di atas, tabel berikut merangkum posisi penelitian ini relatif terhadap studi-studi terdahulu. Secara keseluruhan, dapat diidentifikasi tiga celah utama: (1) sebagian besar studi komparatif menggunakan dataset benchmark, bukan data log produksi nyata; (2) evaluasi terhadap lebih dari tiga algoritma dari keluarga berbeda pada dataset yang sama masih jarang; dan (3) dampak spesifik ketidakseimbangan kelas pada masing-masing keluarga algoritma belum dikaji secara eksplisit dalam konteks log server web. Penelitian ini secara langsung merespons ketiga celah tersebut.

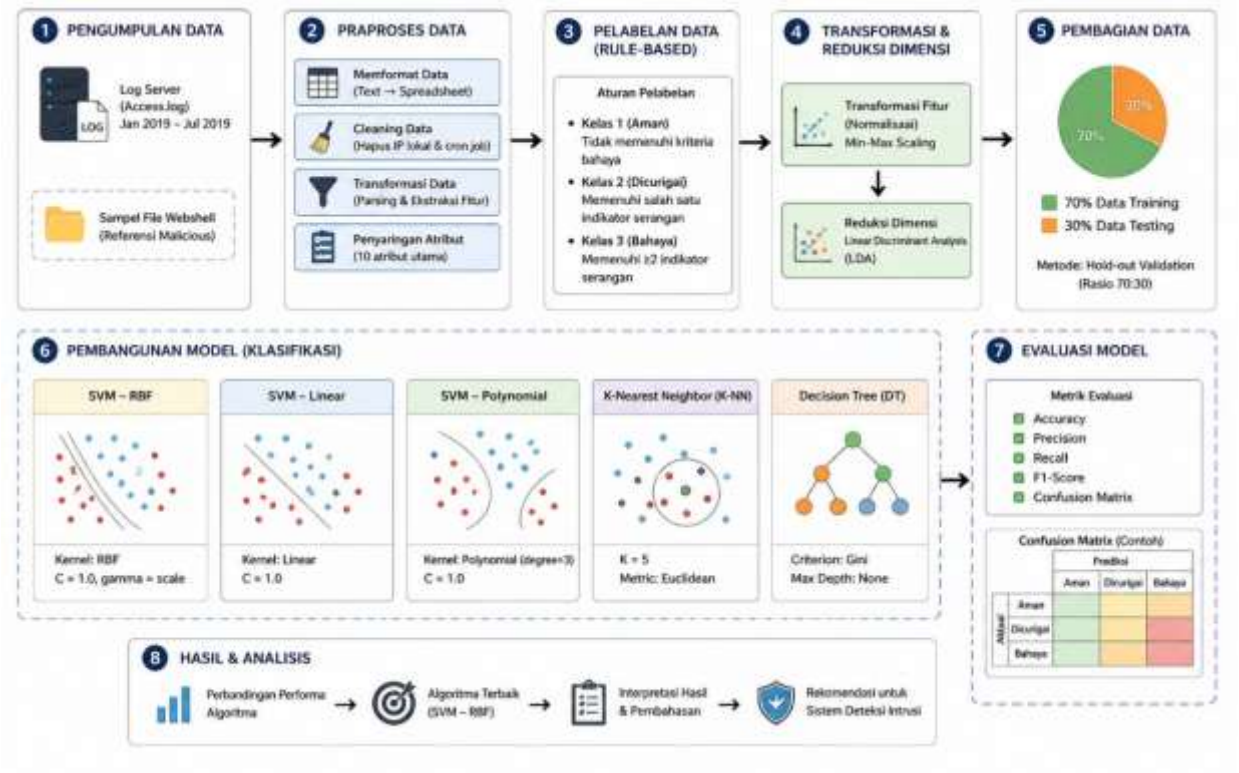
METODOLOGI PENELITIAN

Penelitian ini menggunakan bahasa pemrograman Python dengan library Scikit-learn untuk implementasi algoritma machine learning, serta Pandas dan NumPy untuk pengolahan data. Validasi model dilakukan menggunakan metode hold-out dengan rasio pembagian 70% data latih dan 30% data uji, serta evaluasi performa menggunakan metrik accuracy, precision, recall, dan F1-score. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1. Pengumpulan Data. Dataset yang digunakan merupakan file access.log dari sebuah institusi pendidikan (Universitas XYZ) periode Januari–Juli 2019, diperoleh melalui observasi dan wawancara dengan staf administrator sistem. Data awal terdiri dari 289.899 baris dengan 52 kolom. Selain log akses, juga dikumpulkan sampel berkas webshell yang pernah ditemukan pada server tersebut sebagai acuan dalam proses pelabelan.

Praproses Data. Praproses mengikuti prosedur yang ditetapkan pada penelitian sebelumnya [3]. Secara ringkas, proses ini mencakup tiga tahap utama: (1) pembersihan data, di mana rekaman dari IP lokal dan file cron job (ferguso.php) dihapus sehingga tersisa 37.693 baris data yang relevan;

e-ISSN : 2722-4368

(2) pemberian nama atribut pada kolom-kolom log menjadi: IP Address, User ID, Timestamp, Request, Status, Size, Referer, dan User Agent; serta (3) transformasi data, meliputi pemisahan field HTTP Request menjadi req_method, req_content, dan req_version, konversi IP address ke integer, encoding request ke nilai numerik, dan penghapusan nilai kosong (NaN).



Gambar 1. Diagram Metodologi Penelitian

Pelabelan Berbasis Aturan (Rule-Based Labeling). Pelabelan kelas dilakukan secara manual menggunakan sistem rule-based dengan mempertimbangkan geolokasi IP dan riwayat akses ke berkas webshell. Tiga kategori kelas ditetapkan sebagai berikut:

1. Aman: akses berasal dari Indonesia dan IP tidak memiliki jejak ke webshell.
2. Dicurigai: akses berasal dari luar wilayah Indonesia.
3. Bahaya: IP address terdokumentasi pernah mengakses berkas webshell.

Distribusi hasil pelabelan menghasilkan ketidakseimbangan kelas (class imbalance) yang cukup signifikan, di mana kelas Aman mendominasi dibandingkan kelas Dicurigai dan Bahaya. Kondisi ini menjadi salah satu faktor yang mempengaruhi performa model, khususnya pada metrik recall. Reduksi Dimensi dengan LDA. Linear Discriminant Analysis (LDA) diterapkan untuk mereduksi dimensi dataset menjadi dua komponen (n_components = 2). LDA merupakan metode supervised dimensionality reduction yang mengoptimalkan pemisahan antar kelas dengan memaksimalkan rasio variansi antar-kelas terhadap variansi dalam-kelas [3]. Setelah reduksi, distribusi data dapat divisualisasikan dalam ruang dua dimensi, di mana kelas Bahaya tampak sebagai outlier yang cukup terpisah dari dua kelas lainnya. Dataset kemudian dibagi menjadi 26.065 data latih (70%) dan 11.171 data uji (30%), dengan variabel bebas (X): ip, req_method, req_content, req_version, status, size; dan variabel target (y): detected.

Algoritma Klasifikasi. Lima algoritma klasifikasi diimplementasikan dan dibandingkan dalam penelitian ini:

1. SVM-RBF: SVM dengan kernel Radial Basis Function (C = 10, γ = auto). Kernel RBF

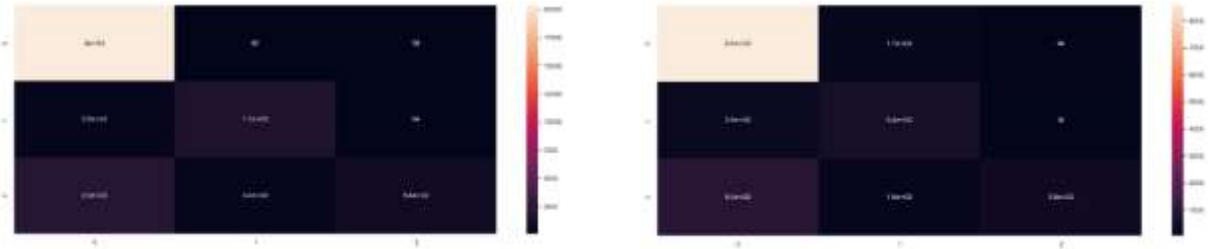
memungkinkan pemisahan non-linear antar kelas melalui transformasi ke ruang fitur berdimensi lebih tinggi.
2. SVM-Linear: SVM dengan kernel linear. Digunakan sebagai baseline untuk menguji apakah data memiliki separabilitas linear.
3. SVM-Polynomial: SVM dengan kernel polinomial, sebagai alternatif non-linear antara kernel RBF dan linear.
4. K-NN via GridSearch (K-NNGS): K-Nearest Neighbor dengan hyperparameter terbaik yang ditentukan melalui grid search (k=9, metrik jarak: Manhattan).
5. Decision Tree (DT): Algoritma pohon keputusan yang membangun model interpretable berbasis pemartisian fitur secara rekursif.

HASIL DAN PEMBAHASAN

Tabel 1 menyajikan performa model SVM-RBF. Model ini mencapai akurasi training sebesar 88% dan testing 85%, dengan validasi silang 87%.

Tabel 1. Hasil Pemodelan SVM-RBF

	Akurasi (%)	**Precision (%)**			**Recall (%)**			**Validasi (%)**
		0	**1**	**2**	**0**	**1**	**2**	
Training	88	89	80	88	99	80	27	87
Testing	85	88	62	76	97	63	26	



Gambar 2. Confusion Matrix training SVM-RBF dan Confusion Matrix testing SVM-RBF

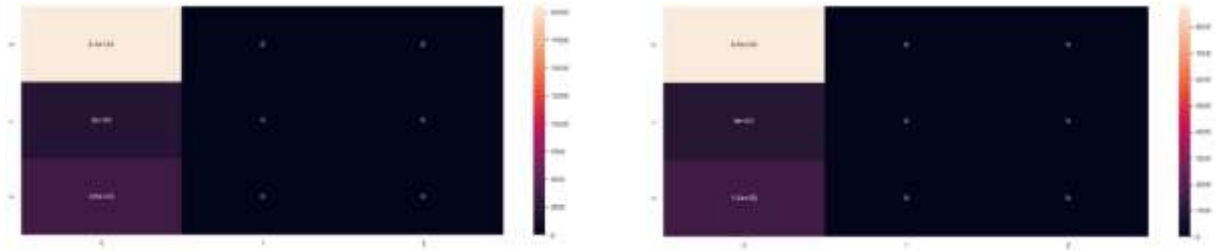
Dari seluruh model yang diuji, SVM dengan kernel Radial Basis Function (RBF) secara konsisten memberikan performa paling stabil. Dengan akurasi training 88% dan testing 86%, kesenjangan antara keduanya relatif kecil (2%), mengindikasikan bahwa model berhasil melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Nilai validasi silang sebesar 87% juga mendukung konsistensi ini. Keunggulan SVM-RBF terletak pada kemampuan kernel RBF dalam melakukan transformasi data ke ruang fitur berdimensi lebih tinggi (feature space), sehingga pola yang tidak dapat dipisahkan secara linear di ruang asal menjadi lebih mudah dipisahkan. Pada data log server yang mengandung campuran fitur numerik (IP, status code, size) dan fitur kategorikal yang di-encode (req_method, req_content), pola intrusi umumnya tidak bersifat linear, sehingga kernel RBF menjadi pilihan yang lebih tepat dibandingkan kernel linear. Perlu dicatat bahwa meskipun akurasi keseluruhan model ini cukup baik, recall pada kelas Bahaya (kelas 2) hanya mencapai 27% untuk training dan 26% untuk testing. Nilai recall yang rendah pada kelas ini mengindikasikan bahwa model masih cukup sering salah mengklasifikasikan aktivitas berbahaya sebagai aman atau dicurigai (false negative). Dalam konteks keamanan siber, false negative merupakan jenis kesalahan yang paling berbahaya karena serangan nyata dapat lolos dari deteksi. Oleh karena itu, meskipun SVM-RBF adalah model terbaik di antara yang diuji, peningkatan recall pada kelas Bahaya tetap menjadi prioritas utama pengembangan. Jika dibandingkan dengan hasil penelitian Cahyanto et al. [3] yang menggunakan dataset dan prosedur praproses yang sama, model

LDA+SVM-RBF pada penelitian tersebut menghasilkan akurasi testing 89,95%. Perbedaan hasil ini sebagian besar dapat dijelaskan oleh perbedaan nilai konstanta pelatihan C: penelitian [3] menggunakan C=1, sedangkan penelitian ini menggunakan C=10. Nilai C yang lebih besar memperkuat penalti terhadap kesalahan klasifikasi pada data training, yang cenderung menghasilkan margin pemisah yang lebih sempit namun bisa lebih sensitif terhadap noise. Di sisi lain, nilai C=10 berpotensi meningkatkan kemampuan model dalam mengenali pola kelas minoritas yang jumlahnya jauh lebih sedikit dalam dataset.

Tabel 2 menunjukkan performa SVM-Linear. Meskipun akurasi training dan testing masing-masing mencapai 78,8% dan 78,6%, model ini gagal mendeteksi kelas Dicurigai dan Bahaya sama sekali, dengan recall dan precision keduanya bernilai 0%.

Tabel 2. Hasil Pemodelan SVM-Linear

	Akurasi (%)	**Precision (%)**			**Recall (%)**			**Validasi (%)**
		0	**1**	**2**	**0**	**1**	**2**	
Training	78,8	79	0	0	100	0	0	78,8
Testing	78,6	79	0	0	100	0	0	

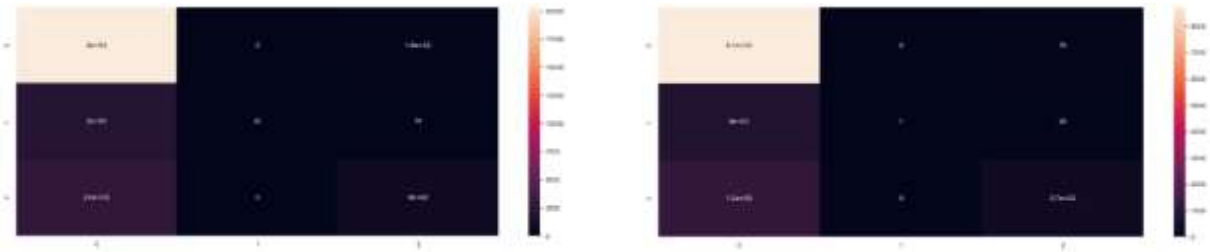


Gambar 3. Confusion Matrix training SVM-Lin dan Confusion Matrix testing SVM-Lin

SVM-Linear menghasilkan performa yang secara statistik terlihat memadai pada level akurasi keseluruhan (78,8% training dan 78,6% testing), namun analisis per kelas mengungkapkan kegagalan yang sangat fundamental: model ini sama sekali tidak mampu mendeteksi kelas Dicurigai (kelas 1) maupun kelas Bahaya (kelas 2), dengan precision dan recall keduanya bernilai 0%. Ini berarti model hanya memprediksi kelas mayoritas (Aman) untuk hampir seluruh sampel, sehingga akurasi tinggi yang tampak hanyalah artefak dari dominasi kelas mayoritas dalam dataset, bukan cerminan kemampuan deteksi yang sesungguhnya. Kegagalan SVM-Linear ini secara eksplisit mengkonfirmasi hipotesis bahwa data log server tidak bersifat linearly separable. Distribusi kelas dalam ruang fitur dua dimensi hasil LDA juga menunjukkan adanya overlap yang signifikan antarkelas, yang tidak dapat diatasi dengan hyperplane linear sederhana. Tabel 3 memperlihatkan hasil SVM-Polynomial. Performa model ini serupa dengan SVM-Linear dalam hal ketidakmampuan membedakan kelas minoritas, dengan nilai recall kelas Dicurigai mendekati nol.

Tabel 3. Hasil Pemodelan SVM-Polynomial

	Akurasi (%)	**Precision (%)**			**Recall (%)**			**Validasi (%)**
		0	**1**	**2**	**0**	**1**	**2**	
Training	81	82	100	76	99	0,02	0,24	81
Testing	81	82	100	75	99	0,01	24	

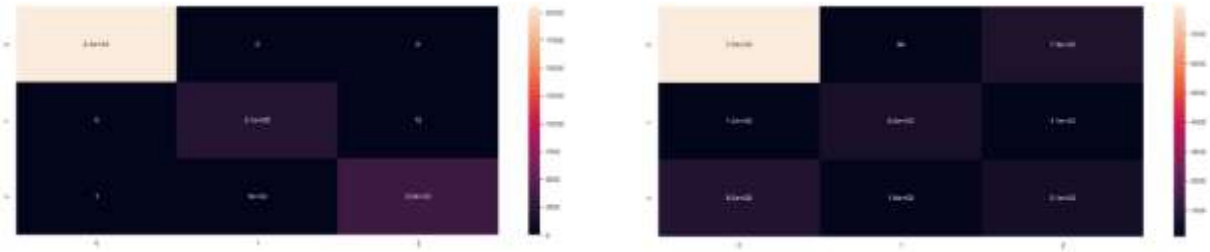


Gambar 4. Confusion Matrix training SVM-Poly dan Confusion Matrix testing SVM-Poly

SVM-Polynomial menunjukkan pola kegagalan yang serupa, meskipun sedikit lebih baik dalam mendeteksi kelas Bahaya (kelas 2) dengan recall 0,24% pada training dan 24% pada testing. Nilai precision kelas 1 yang mencapai 100% namun recall-nya mendekati nol mengindikasikan bahwa model hanya berani memprediksi kelas 1 pada kasus yang sangat sedikit dan hampir selalu benar ketika melakukannya, namun melewatkan sebagian besar anggota kelas tersebut. Perilaku ini menunjukkan bahwa kernel polinomial juga tidak cukup fleksibel untuk menangkap batas keputusan (decision boundary) yang diperlukan pada data ini. K-NN via GridSearch (K-NNGS). Hyperparameter terbaik hasil grid search adalah k=9 dengan metrik jarak Manhattan. Tabel 4 menunjukkan bahwa K-NNGS memiliki akurasi training sangat tinggi (99,6%) namun turun signifikan pada testing (80,4%), mengindikasikan overfitting.

Tabel 4. Hasil Pemodelan K-NNGS

	Akurasi (%)	**Precision (%)**			**Recall (%)**			**Validasi (%)**
		0	**1**	**2**	**0**	**1**	**2**	
Training	99,6	100	95	100	100	100	97	95
Testing	80,4	89	66	31	90	81	25	



Gambar 5. Confusion Matrix training K-NNGS dan Confusion Matrix testing K-NNGS

Baik K-NNGS maupun Decision Tree menunjukkan tanda-tanda overfitting yang sangat jelas. K-NNGS mencapai akurasi training 99,6% dengan penurunan tajam menjadi 80,4% pada testing, sementara Decision Tree menunjukkan pola yang hampir identik: 98% pada training turun menjadi 78% pada testing. Kesenjangan sebesar hampir 20 poin persentase pada kedua model ini jauh melebihi ambang toleransi yang lazim dalam praktik machine learning. Penyebab utama overfitting pada K-NNGS adalah sensitivitas algoritma KNN terhadap jumlah tetangga (k) dan distribusi data. Meskipun grid search telah mengoptimalkan hyperparameter (k=9, jarak Manhattan), algoritma ini masih rentan menghafal pola lokal pada data training yang tidak dapat digeneralisasi. Semakin kecil nilai k, semakin sensitif model terhadap noise dan outlier pada data training.

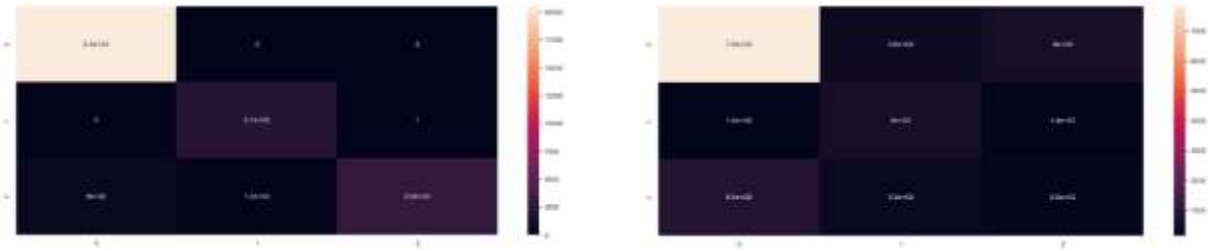
Decision Tree (DT). Tabel 5 menunjukkan pola serupa dengan K-NNGS, di mana Decision Tree mencapai akurasi training 98% namun hanya 78% pada testing, memperkuat indikasi overfitting pada kedua algoritma berbasis instance dan partisi ini.

e-ISSN : 2722-4368

Tabel 5. Hasil Pemodelan Decision Tree

	Akurasi (%)	Precision (%)			Recall (%)			Validasi (%)
		0	1	2	0	1	2	
Training	98	98	94	100	100	100	82	93
Testing	78	88	46	25	89	68	17	

Gambar 6. Pohon Keputusan yang Terbentuk



Gambar 7. Confusion Matrix training DT dan Confusion Matrix testing DT

Pada Decision Tree, overfitting terjadi karena pohon yang terbentuk cenderung tumbuh terlalu dalam (deep) hingga memisahkan setiap titik data training secara sempurna, termasuk noise dan outlier. Tanpa mekanisme pruning yang agresif atau pembatasan kedalaman pohon (max_depth), Decision Tree hampir pasti akan overfit pada dataset dengan distribusi kelas yang tidak seimbang. Tingginya akurasi validasi silang (93%) dibandingkan testing (78%) juga mengindikasikan bahwa pembagian data secara acak dalam k-fold memberikan distribusi kelas yang lebih seimbang per fold dibandingkan pembagian hold-out 70:30. Kondisi ini diperparah secara sistemik oleh ketidakseimbangan kelas (class imbalance). Ketika kelas Bahaya hanya merepresentasikan sebagian kecil dari total data, algoritma berbasis instance seperti KNN akan didominasi oleh tetangga dari kelas mayoritas dalam radius pencarian, sementara Decision Tree akan cenderung membuat keputusan berdasarkan kelas mayoritas di setiap node karena memberikan penurunan Gini impurity atau information gain yang lebih besar. Tabel 6 merangkum F1-score seluruh model. Rata-rata akurasi training dan testing seluruh model masing-masing adalah 88,9% dan 80,6%.

Tabel 6. Ringkasan F1-Score Seluruh Model

Model	Training F1 (%)			Testing F1 (%)			Ket.
	0	1	2	0	1	2	
SVM-RBF	94	80	50	92	63	40	Terbaik
SVM-Lin	88	0	0	88	0	0	Gagal deteksi
SVM-Poly	90	0,04	38	90	0,02	36	Tidak stabil
K-NNGS	100	97	98	89	72	28	Overfit
DT	99	97	90	88	55	20	Overfit

Linear Discriminant Analysis berperan penting dalam penelitian ini sebagai tahap pra-pemodelan. Dengan mereduksi dimensi dari delapan fitur asli menjadi dua komponen LDA, proses pelatihan menjadi jauh lebih efisien. Berbeda dengan penelitian [3] yang melaporkan pengurangan waktu komputasi dari 324 detik (SVM-RBF tanpa LDA) menjadi hanya 1,3 detik (LDA+SVM-RBF), penelitian ini juga merasakan manfaat serupa dalam hal efisiensi komputasi. Visualisasi sebaran data pada ruang dua dimensi hasil LDA mengungkapkan karakteristik distribusi kelas yang penting. Kelas Bahaya tampak sebagai kelompok outlier yang terpisah dari dua kelas lainnya, yang secara teoritis seharusnya memudahkan deteksi. Namun, jumlah sampel yang sangat sedikit pada kelas ini menyebabkan batas keputusan yang dipelajari oleh model menjadi kurang presisi. Sementara itu, kelas Aman dan Dicurigai menunjukkan overlap yang cukup signifikan dalam ruang LDA, mengindikasikan bahwa kedua kelas ini memiliki karakteristik fitur yang secara statistik sangat mirip. Terdapat satu keterbatasan fundamental dari LDA yang perlu diperhatikan: LDA mengasumsikan bahwa setiap kelas memiliki distribusi Gaussian dengan matriks kovarians yang sama (homoscedasticity). Jika asumsi ini tidak terpenuhi yang sangat mungkin terjadi pada data log server yang bersifat heterogen komponen LDA yang dihasilkan mungkin tidak merepresentasikan arah pemisahan antar kelas yang paling optimal. Hal ini dapat menjadi salah satu faktor yang membatasi performa model meskipun LDA telah diterapkan.

Ketidakseimbangan distribusi kelas merupakan tantangan struktural yang paling menentukan performa seluruh model dalam penelitian ini. Berdasarkan distribusi data yang tercermin dari confusion matrix, kelas Aman mendominasi dataset secara signifikan, sementara kelas Bahaya merupakan kelas minoritas dengan jumlah sampel yang jauh lebih sedikit. Kondisi ini menciptakan bias sistemik dalam proses pelatihan di mana model cenderung mengoptimalkan performa pada kelas mayoritas dengan mengorbankan sensitivitas terhadap kelas minoritas. Dalam konteks keamanan siber, ketidakseimbangan ini bersifat inherent dan mencerminkan kondisi nyata: pada sebuah server yang berjalan normal, sebagian besar akses memang bersifat aman, sementara aktivitas berbahaya merupakan kejadian langka (rare event). Ironisnya, justru kejadian langka inilah yang paling kritis untuk dideteksi. Oleh karena itu, metrik evaluasi yang hanya berfokus pada akurasi keseluruhan menjadi menyesatkan dan tidak tepat untuk skenario ini. Untuk penelitian selanjutnya, beberapa strategi penanganan class imbalance dapat dipertimbangkan. Pada level data, teknik oversampling seperti SMOTE (Synthetic Minority Over-sampling Technique) atau ADASYN dapat digunakan untuk menghasilkan sampel sintetis pada kelas minoritas, sehingga distribusi kelas menjadi lebih seimbang tanpa kehilangan informasi. Pada level algoritma, parameter class_weight='balanced' pada SVM dapat digunakan untuk memberikan bobot penalti yang lebih besar pada kesalahan klasifikasi kelas minoritas, sehingga mendorong model untuk lebih berhati-hati dalam mendeteksi kelas Bahaya dan Dicurigai. Untuk memahami lebih dalam kegagalan model, analisis kualitatif terhadap pola

kesalahan klasifikasi (misclassification) dilakukan berdasarkan informasi yang tersedia dari confusion matrix dan karakteristik fitur dataset. Analisis ini bertujuan mengidentifikasi kondisi-kondisi spesifik yang secara sistematis menyulitkan seluruh model, terlepas dari algoritma yang digunakan. Dari confusion matrix SVM-RBF yang merupakan model terbaik, terlihat bahwa kesalahan terbesar terjadi pada prediksi kelas Bahaya (kelas 2) yang sering diklasifikasikan sebagai kelas Aman (kelas 0). Pola ini mengindikasikan adanya ambiguitas fitur antara kedua kelas tersebut. Secara kontekstual, hal ini masuk akal: IP address yang termasuk kelas Bahaya karena pernah mengakses webshell kemungkinan besar juga melakukan banyak akses normal (GET halaman biasa, memuat aset statis) sebelum atau sesudah melakukan akses berbahaya. Akibatnya, sebagian besar rekaman log dari IP berbahaya tersebut memiliki profil fitur yang identik dengan akses aman, dan hanya sebagian kecil yang memiliki ciri khas serangan pada field req_content.

Kondisi serupa terjadi pada kebingungan antara kelas Dicurigai (kelas 1) dan kelas Aman (kelas 0). Kedua kelas ini dibedakan semata-mata berdasarkan geolokasi IP (luar negeri vs. dalam negeri), sementara fitur-fitur teknis lainnya seperti req_method, status code, dan ukuran respons seringkali tidak berbeda secara statistik. Dengan kata lain, rule-based labeling yang digunakan menciptakan label kelas yang tidak sepenuhnya berkorelasi dengan sinyal yang dapat ditangkap dari fitur teknis log itu sendiri. Hal ini menjelaskan mengapa overlap antara kelas Aman dan Dicurigai sangat signifikan bahkan setelah reduksi LDA. Temuan ini memiliki implikasi penting: kualitas dan relevansi fitur yang digunakan untuk pelabelan harus selaras dengan fitur yang tersedia untuk pelatihan model. Jika pelabelan didasarkan pada informasi geolokasi, maka informasi geolokasi tersebut atau fitur turunannya seperti flag "is_foreign" sebaiknya juga dimasukkan secara eksplisit sebagai fitur model. Dalam penelitian ini, IP address dikonversi ke integer tanpa informasi geolokasi yang eksplisit, sehingga model tidak memiliki sinyal yang memadai untuk membedakan kelas Dicurigai dari kelas Aman secara konsisten [15][16]. Implikasi praktis dari analisis ini adalah rekomendasi penambahan fitur geolokasi (misalnya kode negara atau flag is_domestic) sebagai fitur eksplisit dalam model, serta pertimbangan untuk menggabungkan kelas Dicurigai dan Aman menjadi satu kelas non-bahaya dalam penelitian selanjutnya, sehingga masalah menjadi binary classification yang lebih tajam antara "berbahaya" dan "tidak berbahaya". Penyederhanaan ini berpotensi meningkatkan recall kelas Bahaya secara signifikan tanpa memerlukan data tambahan.

Penelitian ini memberikan kontribusi nyata dalam bentuk evaluasi komparatif multi-algoritma yang lebih komprehensif dibandingkan studi sebelumnya [3]. Dengan menguji lima algoritma secara bersamaan pada dataset yang sama, penelitian ini memberikan gambaran yang lebih lengkap tentang kekuatan dan kelemahan relatif masing-masing pendekatan, yang dapat dijadikan panduan bagi praktisi keamanan siber dalam memilih metode yang tepat untuk sistem deteksi intrusi berbasis log. Namun, terdapat beberapa keterbatasan yang perlu diakui. Pertama, dataset yang digunakan hanya berasal dari satu institusi dalam periode tujuh bulan, sehingga kemampuan generalisasi model ke konteks server yang berbeda belum dapat dipastikan. Kedua, proses pelabelan rule-based yang dilakukan secara manual berpotensi mengandung bias subjektif dan mungkin tidak menangkap seluruh variasi pola serangan yang ada. Ketiga, penelitian ini belum mempertimbangkan fitur-fitur kontekstual yang lebih kaya seperti analisis payload request, pola temporal (frekuensi akses per menit/jam), atau korelasi antar-sesi dari IP yang sama, yang berpotensi meningkatkan kemampuan diskriminasi model secara signifikan. Arah pengembangan yang direkomendasikan mencakup: (1) penerapan teknik penyeimbangan data seperti SMOTE sebelum pelatihan model; (2) eksplorasi fitur tambahan berbasis analisis temporal dan payload; (3) pengujian model pada dataset log server dari institusi berbeda untuk menguji generalisabilitas; dan (4) pertimbangan penggunaan metode ensemble seperti Random Forest atau Gradient Boosting yang secara inheren lebih robust terhadap ketidakseimbangan kelas dibandingkan model tunggal.

KESIMPULAN

Penelitian ini mengimplementasikan dan membandingkan lima algoritma klasifikasi berbasis data mining SVM-RBF, SVM-Linear, SVM-Polynomial, K-NN via GridSearch, dan Decision Tree untuk mendeteksi aktivitas malicious pada log server nyata menggunakan kombinasi rule-based labeling dan reduksi dimensi LDA. Berdasarkan hasil evaluasi, tiga kesimpulan utama dapat ditarik. Pertama, SVM-RBF memberikan performa paling stabil dengan akurasi testing 86% dan kesenjangan training-testing yang relatif kecil, menjadikannya pilihan terbaik untuk skenario deteksi intrusi berbasis log. Kedua, K-NNGS dan Decision Tree mengalami overfitting signifikan akibat ketidakseimbangan kelas, dengan akurasi training jauh melampaui performa pada data uji. Ketiga, SVM-Linear dan SVM-Polynomial gagal mendeteksi kelas minoritas secara efektif, membuktikan bahwa data log server bersifat non-linearly separable. Penelitian ini berkontribusi dalam memperluas studi komparatif pada dataset log server nyata, melengkapi temuan sebelumnya [3]. Pengembangan selanjutnya dapat difokuskan pada penanganan ketidakseimbangan kelas melalui teknik oversampling, peningkatan rekayasa fitur, serta evaluasi pada dataset log yang lebih beragam untuk meningkatkan generalisasi model.

DAFTAR PUSTAKA

[1] M. Khan, “Advanced System Log Analyzer for Anomaly Detection and Cyber Forensic Investigations Using LSTM and Transformer Networks,” *Journal of Cloud Computing*, vol. 14, no. 1, p. 60, 2025, doi: 10.1186/s13677-025-00789-y.

[2] A. R. Nisa, A. D. Wijayanto, A. P. J. Priana, and A. Setiawan, “Analisis Log Server untuk Mendeteksi Serangan DDoS pada Keamanan Jaringan di Website,” *Journal of Internet and Software Engineering*, vol. 1, no. 3, p. 17, 2024, doi: 10.47134/pjise.v1i3.2612.

[3] A. H. Shah, D. Pasha, E. Habib Zadeh, and S. Konur, “Automated Log Analysis and Anomaly Detection Using Machine Learning,” in *Fuzzy Systems and Data Mining VIII*, vol. 358, in Frontiers in Artificial Intelligence and Applications, vol. 358. , IOS Press, 2022, pp. 137–147. doi: 10.3233/FAIA220378.

[4] I. Khaerani and L. Budi Handoko, “IMPLEMENTASI DAN ANALISA HASIL DATA MINING UNTUK KLASIFIKASI SERANGAN PADA INTRUSION DETECTION SYSTEM (IDS) DENGAN ALGORITMA C4.5,” 2015.

[5] K. A. Cahyanto, M. Anis, A. Hilmi, and M. Mustamiin, “PENGUJIAN RULE-BASED PADA DATASET LOG SERVER MENGGUNAKAN SUPPORT VECTOR MACHINE BERBASIS LINEAR DISCRIMINAT ANALYSIS UNTUK DETEKSI MALICIOUS ACTIVITY,” vol. 9, no. 2, pp. 245–254, 2022, doi: 10.25126/jtiik.202294107.

[6] A. R. Saputro, Nurchim, and V. Atina, “Identifikasi Anomali Keamanan Server Nginx Menggunakan Algoritma Isolation Forest,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, 2025, doi: 10.36040/jati.v9i2.13110.

[7] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, “Deep Neural Network Based Real-Time Intrusion Detection System,” *SN Comput. Sci.*, vol. 3, no. 2, p. 145, 2022, doi: 10.1007/s42979-022-01031-1.

[8] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1803–1816, 2021, doi: 10.1109/TNSM.2020.3014929.

[9] H. Hindy *et al.*, “A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems,” *IEEE Access*, vol. 8, pp. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.

[10] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, and A. Rauber, "A Framework for Cyber Threat Intelligence Extraction from Raw Log Data," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, p. 3. doi: 10.1109/BigData47090.2019.9006328.

[11] H. Kaur, H. S. Pannu, and A. K. Malhi, "A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, 2019, doi: 10.1145/3343440.

[12] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of Network-based Intrusion Detection Data Sets," *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.

[13] A. Pamuji, "Prediksi Otorisasi Pengguna Sistem Berkas pada Algoritma Klasifikasi dengan Teknik Naïve Bayes," *Infomatek: Jurnal Informatika, Manajemen dan Teknologi*, vol. 24, no. 1, 2022, doi: 10.23969/infomatek.v24i1.4604.

[14] I. A. S. Dewi Paramitha, G. M. A. Sasmita, and I. M. S. Raharja, "Analisis Data Log IDS Snort dengan Algoritma Clustering Fuzzy C-Means," *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 1, pp. 95–100, 2020, doi: 10.24843/MITE.2020.v19i01.P14.

[15] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward Effective Intrusion Detection Using Log-Cosh Conditional Variational Autoencoder," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6187–6196, 2021, doi: 10.1109/JIOT.2020.3034621.

[16] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni, and P. Filzmoser, "Dynamic Log File Analysis: An Unsupervised Cluster Evolution Approach for Anomaly Detection," *Comput. Secur.*, vol. 79, pp. 94–116, 2018, doi: 10.1016/j.cose.2018.08.009.

[17] F. Nova, M. D. Pratama, and D. Prayama, "Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan DoS," *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 3, no. 1, pp. 1–7, 2022, doi: 10.62527/jitsi.3.1.59.

[18] R. S. Miani, G. D. G. Bernardo, G. W. Cassales, H. Senger, and E. R. de Faria, "A Survey of Data Stream-Based Intrusion Detection Systems," *IEEE Access*, vol. 13, pp. 72953–72983, 2025, doi: 10.1109/ACCESS.2025.3561105.