Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1945~1954
e-ISSN: 2548-6861

Support Vector Machine Classification Algorithm for Detecting DDoS Attacks on Network Traffic

Yoki Irawan [1]*, Rina Pramitasari [2]*, Wahid Miftahul Ashari [3]*, Aiko Nur Hendry Yansyah [4]*
* Teknik Komputer, Faklultas Ilmu Komputer, Universitas Amikom Yogyakarta
yoki.irwn@students.amikom.ac.id [1], rina.pramitasari@amikom.ac.id [2], wahidashari@amikom.ac.id [3], aikohendry@students.amikom.ac.id [4]

Article Info

Article history:

Received 2025-06-30
Revised 2025-07-19
Accepted 2025-08-09

Keyword:

DDoS,
Support Vector Machine,
Attack Detection,
CICIDS2017,
Network Traffic.

ABSTRACT

Distributed Denial of Service (DDoS) attacks represent a significant danger in network security because they can lead to extensive service interruptions. With these attacks increasingly mirroring regular traffic, smart and effective detection systems are essential. This research seeks to assess the efficacy of the Support Vector Machine (SVM) classification algorithm in identifying DDoS attacks in network traffic. The data utilized is CICIDS2017, focusing on the subset Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv, which contains both legitimate traffic and DDoS attacks like DoS-Hulk, DoS-GoldenEye, and DDoS. The preprocessing stage included eliminating duplicates and null entries, label binary encoding, normalization through Min-Max Scaler, and feature selection applying the Chi-Square technique. The data was divided into 80% for training and 20% for testing purposes. The Radial Basis Function (RBF) kernel was utilized to train the SVM model, and hyperparameter optimization was performed with GridSearchCV. The evaluation of the model's performance was conducted through accuracy, precision, recall, F1-score, confusion matrix, and visual representations including ROC and Precision-Recall Curves. The findings indicate that prior to tuning, the model reached an accuracy of 97%, which increased to 99% post-tuning, accompanied by an F1-score of 0.99. This shows that the SVM algorithm, when paired with appropriate preprocessing and optimization, is very efficient in identifying DDoS attacks within network traffic.

I. PENDAHULUAN

Serangan Distributed Denial of Service (DDoS) adalah salah satu risiko besar dalam bidang keamanan jaringan. Serangan ini bekerja dengan cara membanjiri sistem target dengan lalu lintas jaringan secara masif, sehingga menyebabkan gangguan bahkan kegagalan total pada layanan. Kompleksitas lalu lintas DDoS modern yang menyerupai trafik normal menjadikannya semakin sulit untuk dideteksi menggunakan metode konvensional. Oleh karena itu, pendekatan berbasis pembelajaran mesin (Machine Learning) mulai dikembangkan dan diterapkan secara luas karena memiliki kemampuan dalam mengenali pola serangan secara otomatis. Terutama Support Vector Machine (SVM), yang adalah salah satu algoritma klasifikasi yang terbukti efektif di berbagai bidang, termasuk pada sistem deteksi intrusi jaringan seperti deteksi serangan *Distributed Denial of Service* (DDoS) dan *Denial of Service* (DoS). Keunggulan utama dari SVM terletak pada kemampuannya untuk mengklasifikasikan data dengan margin optimal, yang memungkinkan model menghasilkan generalisasi yang kuat, bahkan pada data kompleks dan berdimensi tinggi [1]. SVM menggunakan pendekatan matematis yang solid dalam menemukan hyperplane pemisah dengan margin maksimum, yang menjamin solusi global optimal dan menjadikan model lebih stabil serta akurat dibandingkan metode yang hanya mengandalkan solusi lokal.

Salah satu aspek penting dari SVM adalah penerapan kernel trick, seperti kernel *Radial Basis Function* (RBF), yang memungkinkan pemetaan data non-linear ke ruang fitur berdimensi lebih tinggi agar dapat dipisahkan secara linear [1]. Fitur ini menjadikan SVM sangat cocok dalam

menghadapi data jaringan yang bersifat non-linear dan dinamis. Selain itu, setelah proses pelatihan selesai, proses inferensi pada SVM relatif cepat karena hanya bergantung pada support vectors yang relevan, sehingga efisien untuk aplikasi real-time seperti deteksi intrusi, pengenalan wajah, maupun diagnosis medis[1].

Berbagai studi sebelumnya, menunjukkan bahwa SVM mampu mencapai nilai akurasi dan recall yang tinggi dalam mendeteksi serangan DDoS, bahkan ketika dihadapkan pada dataset yang tidak seimbang. Namun demikian, SVM juga memiliki beberapa keterbatasan, seperti kebutuhan komputasi yang tinggi saat diterapkan pada dataset berukuran besar, serta sensitivitas terhadap pemilihan parameter seperti C dan gamma. Oleh karena itu, diperlukan proses hyperparameter tuning yang cermat agar performa model dapat dioptimalkan secara maksimal. Secara keseluruhan, SVM menawarkan keseimbangan antara akurasi, efisiensi prediksi, dan fleksibilitas dalam menangani data non-linear, menjadikannya salah satu metode yang unggul dalam berbagai bidang, termasuk pengenalan pola, analisis citra, dan khususnya, sistem deteksi intrusi jaringan. tinggi[1].

Penelitian dalam bidang keamanan siber terus berkembang seiring dengan meningkatnya kompleksitas serangan, khususnya serangan *Distributed Denial of Service* (DDoS). Salah satu pendekatan yang telah terbukti efektif adalah penggunaan algoritma *Support Vector Machine* (SVM). Dalam studi yang berfokus pada pengembangan sistem deteksi otomatis berbasis SVM, dilakukan eksperimen menggunakan dataset CICIDS2017 yang disediakan oleh Canadian Institute for Cybersecurity [2]. Dataset ini terdiri dari lebih dari 10.000 tuple dan mencakup 78 atribut penting, seperti durasi aliran, port tujuan, dan jumlah paket maju-mundur. Setelah melalui tahapan prapemrosesan data seperti penghapusan nilai hilang dan normalisasi, model SVM menunjukkan peningkatan kinerja signifikan. Melalui penerapan teknik scaling, akurasi model meningkat dari 85% menjadi 97%, dengan metrik lain seperti precision, recall, dan F1-score juga menunjukkan peningkatan yang konsisten [3]. Temuan ini menegaskan potensi SVM dalam mendeteksi serangan DDoS secara real-time dan meningkatkan keandalan sistem keamanan jaringan.

Pendekatan lain yang juga terbukti efektif adalah kombinasi antara *Principal Component Analysis* (PCA) dan SVM [4]. PCA digunakan untuk mereduksi dimensi dari data trafik jaringan yang besar dan kompleks, dengan tujuan mengekstrak fitur-fitur utama yang relevan sebelum dilakukan klasifikasi oleh SVM. Pengurangan dimensi ini tidak hanya mempercepat proses klasifikasi, namun juga meningkatkan akurasi model dengan mengeliminasi fitur-fitur yang kurang relevan. Hasil pengujian menunjukkan bahwa metode ini mencapai akurasi pengujian sebesar 93,83% dan akurasi pelatihan sebesar 97,17%. Dengan demikian, kombinasi PCA dan SVM terbukti sebagai pendekatan yang efisien dan efektif dalam mendeteksi serangan DdoS.

Selain pendekatan berbasis fitur statistik jaringan, pendekatan berbasis konten juga menunjukkan hasil yang sangat menjanjikan. Salah satu studi mengusulkan teknik deteksi DDoS dengan memanfaatkan fitur *N-Gram* dari payload jaringan, khususnya HTTP, dikombinasikan dengan algoritma SVM. Ekstraksi fitur dilakukan menggunakan variasi ukuran *N-Gram* (dari 1-Gram hingga 6-Gram), disertai teknik hibrida yang menggabungkan analisis jarak seperti *Chi-Square* dan *Cosine Similarity* untuk meningkatkan akurasi klasifikasi. Hasil eksperimen menunjukkan bahwa penggunaan fitur 4-Gram memberikan performa terbaik, dengan akurasi mencapai 100% pada beberapa dataset, termasuk CIC-IDS2017 dan MIB-2016 [5]. Pendekatan ini menghasilkan peningkatan akurasi hingga 15% dibandingkan metode konvensional, memperkuat posisi SVM sebagai salah satu algoritma paling efektif dalam mendeteksi serangan DDoS berbasis konten. Namun, pendekatan ini masih memiliki keterbatasan dalam hal generalisasi terhadap berbagai jenis serangan dan dataset.

Penelitian yang mengeksplorasi efektivitas algoritma *Support Vector Machine* (SVM) dalam mendeteksi serangan *Distributed Denial of Service* (DDoS) pada sistem keamanan siber. Fokus utama kajian mencakup tahap pra-pemrosesan data, penanganan ketidakseimbangan kelas menggunakan teknik oversampling, serta penerapan *Principal Component Analysis* (PCA) sebagai metode pemilihan fitur untuk mengurangi dimensi data tanpa mengorbankan informasi penting. Proses klasifikasi dilakukan menggunakan dataset CICIDS2017, dengan tujuan untuk membedakan lalu lintas jaringan antara aktivitas benign dan malicious [6]. Hasil evaluasi menunjukkan bahwa model SVM berhasil mencapai akurasi 90%, dengan precision 0,87, recall 0,94, dan F1-score 0,94. Metrik ini mencerminkan kinerja yang cukup baik, terutama dalam konteks deteksi serangan DDoS, meskipun performanya masih berada di bawah beberapa algoritma lain seperti *Random Forest* dan *K-Nearest Neighbors* yang diuji pada dataset yang sama. Temuan ini mengindikasikan bahwa SVM tetap merupakan metode yang kompetitif dan efektif untuk deteksi serangan jaringan, terutama apabila didukung oleh strategi pra-pemrosesan dan seleksi fitur yang tepat.

Dalam upaya meningkatkan kinerja deteksi, teknik seleksi digunakan pada fitur berbasis optimisasi untuk meningkatkan deteksi DDoS dalam jaringan SDN menggunakan SVM dan algoritma lainnya [7]. Meskipun pendekatan ini menunjukkan peningkatan kinerja, masih diperlukan penelitian lebih lanjut untuk mengatasi keterbatasan dalam hal efisiensi dan akurasi.

Model hybrid yang menggabungkan *Convolutional Neural Networks* (CNN) dan algoritma genetika untuk seleksi fitur dalam mendeteksi serangan DDoS di lingkungan cloud. Model menunjukkan tingkat deteksi yang tinggi, namun kompleksitasnya dapat menjadi tantangan dalam implementasi nyata [8].

Memperkenalkan metode seleksi fitur berbasis *Generative Adversarial Networks* (GANFS) untuk mitigasi DdoS [9]. Metode ini menunjukkan peningkatan akurasi dan efisiensi komputasi dengan mengurangi dimensi fitur secara

signifikan, namun implementasinya memerlukan sumber daya komputasi yang besar.

Menggabungkan metode *feature importance* dengan SVM untuk mendeteksi serangan DdoS [10]. Studi menunjukkan bahwa pemilihan fitur yang tepat dapat meningkatkan akurasi deteksi, namun evaluasi lebih lanjut masih dibutuhkan dalam situasi dunia nyata.

Mengevaluasi efektivitas berbagai algoritma pembelajaran mesin dalam mendeteksi serangan DDoS di lingkungan SDN [11]. Mereka menemukan bahwa Random Forest memiliki akurasi tertinggi, namun SVM juga menunjukkan kinerja yang kompetitif, terutama dalam skenario dengan data berdimensi tinggi.

Mengembangkan model deteksi DDoS real-time menggunakan berbagai algoritma pembelajaran mesin, termasuk SVM. Studi ini menyoroti pentingnya pemilihan fitur dan preprocessing data dalam meningkatkan kinerja deteksi serangan DdoS [12].

Menekankan pentingnya seleksi fitur dan interpretabilitas dalam deteksi serangan DoS berbasis pembelajaran mesin. Mereka menunjukkan bahwa analisis statistik dan rekayasa fitur dapat meningkatkan pemahaman tentang perilaku serangan dan meningkatkan akurasi deteksi [13].

Dalam penelitian ini, pemilihan dataset CICIDS2017 didasarkan pada sejumlah pertimbangan strategis. CICIDS2017 merupakan dataset benchmark yang dikembangkan oleh Canadian Institute for Cybersecurity, dan secara luas digunakan dalampenelitian deteksi serangan siber [2]. Dataset ini merepresentasikan lingkungan jaringan nyata dengan menyertakan trafik normal dan berbagai jenis serangan seperti DDoS, DoS, Botnet, Brute Force, dan lainnya. Salah satu keunggulan dari CICIDS2017 adalah struktur datanya yang lengkap, dengan lebih dari 80 fitur flow-based yang menggambarkan perilaku jaringan dari berbagai protokol seperti HTTP, FTP, SSH, DNS, dan lain-lain. Selain itu, dataset ini bersifat open-access, terdokumentasi dengan baik, dan telah divalidasi oleh komunitas peneliti keamanan informasi sebagai acuan standar untuk pengujian algoritma deteksi.

Penelitian ini bertujuan untuk menilai efektivitas algoritma Support Vector Machine (SVM) dalam mengenali serangan DDoS pada lalu lintas jaringanmenggunakan dataset CICIDS2017. Berbeda dengan penelitian sebelumnya, penelitian ini menggunakan Kombinasi teknik preprocessing menyeluruh, seleksi fitur dengan *SelectKBest*, serta *tuning hyperparameter* SVM dengan *GridSearch*, memberikan performa tinggi yang diukur secara objektif melalui ROC dan *Precision-Recall Curve*. Ketiga, visualisasi kinerja model jarang dibahas dalam literatur baik lokal maupun internasional, sehingga hal ini menjadi keunggulan dalam penelitian ini. Dengan demikian, penelitian dapat dapat menyumbangkan pendekatan terstruktur yang lebih efektif dan tepat dalam mengidentifikasi serangan DDoS. Hasilnya diharapkan dapat digunakan sebagai dasar pengembangan sistem IDS yang praktis dan berkinerja tinggi, baik dalam lingkungan enterprise maupun IoT.

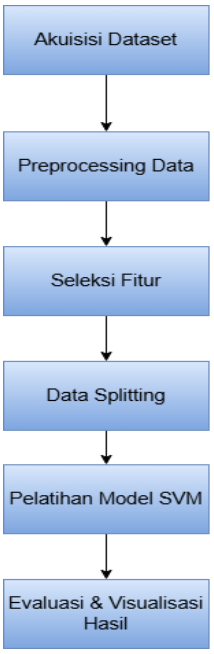
II. METODE

Penelitian ini menerapkan metode kuantitatif eksperimental melalui langkah-langkah terstruktur mulai dari pengumpulan data, preprocessing, pemilihan fitur, pelatihan model, penyesuaian parameter, hingga penilaian kinerja. Metodologi ini dirancang untuk mengevaluasi efektivitas algoritma *Support Vector Machine* (SVM) dalam mengidentifikasi serangan *Distributed Denial of Service* (DDoS) dengan memanfaatkan dataset CICIDS2017.

A. Metode Alur

Makalah Penelitian ini dilakukan dengan mengikuti alur metodologi yang sistematis, bertujuan untuk mengevaluasi performa algoritma *Support Vector Machine* (SVM) dalam mendeteksi serangan *Distributed Denial of Service* (DDoS) pada lalu lintas jaringan menggunakan dataset CICIDS2017.

Gambar 1 menyajikan diagram alur penelitian yang mencerminkan tahapan-tahapan utama dalam studi ini. Proses penelitian mencakup akuisisi data, pra-pemrosesan, seleksi fitur, pemisahan data, pelatihan model *Support Vector Machine* (SVM), dan diakhiri dengan evaluasi serta visualisasi hasil.



Gambar 1.Diagram Alur

B. Akuisisi Dataset

Dataset yang dipakai dalam penelitian adalah Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv, bagian dari CICIDS2017, yang dikembangkan oleh *Canadian Institute for Cybersecurity*(CIC) dan banyak digunakan sebagai benchmark dalam penelitian keamanan jaringan [2]. Dataset ini dirancang untuk merepresentasikan aktivitas jaringan nyata termasuk trafik normal dan serangan siber, dan sering dijadikan *benchmark* dalam penelitian deteksi intrusi dan klasifikasi serangan siber.

Berdasarkan dokumentasi resmi CIC, file mencakup beberapa label trafik, seperti yang tertera pada Tabel 1.

TABEL I
LABEL TRAFFIC

Label Serangan	Deskripsi Singkat
DoS Hulk	Mengirim trafik HTTP besar secara terus-menerus
DoS GoldenEye	Serangan HTTP yang mensimulasikan overload server
DDoS	Serangan terdistribusi dari banyak sumber terhadap satu target
BENIGN	Trafik normal tanpa aktivitas berbahaya

Dataset terdiri dari ± 45.000–50.000 baris data. Mayoritas data diklasifikasikan sebagai serangan (DDoS), sedangkan kelas minoritas merupakan trafik BENIGN. Ini menciptakan tantangan tersendiri dalam hal imbalance class, yang diatasi melalui teknik evaluasi seperti precision, recall, dan ROC Curve.

Adapun jumlah fitur (kolom) lebih dari 75, yang mencakup:

- Fitur berbasis flow: seperti Flow Duration, Total Fwd/Bwd Packets
- Fitur berbasis waktu**:** seperti Flow Bytes/s, Flow Packets/s
- Fitur arah trafik**:** seperti Fwd Header Length, Bwd Packets
- Fitur statistik**:** seperti Min, Max, Std Dev, Mean, dll.

C. Preprocessing Data

Tahap preprocessing dilakukan untuk membersihkan dan mempersiapkan data sebelum digunakan dalam proses pelatihan model. Langkah awal meliputi penghapusan nilai kosong (null values) dengan menggunakan fungsi dropna() guna menjaga konsistensi data. Selanjutnya, data duplikat dihapus untuk menghindari bias dalam proses pelatihan. Beberapa atribut yang dianggap tidak relevan, seperti Flow ID, Timestamp, dan alamat IP, dihapus agar tidak memengaruhi proses klasifikasi. Proses label encoding diterapkan dengan mengubah label 'BENIGN' menjadi 0, dan label serangan seperti 'DDoS' serta 'DoS-Hulk' menjadi 1, sesuai dengan kebutuhan klasifikasi biner. Tahap akhir preprocessing adalah normalisasi data menggunakan *MinMaxScaler*, yang bertujuan untuk menyamakan skala nilai fitur dalam rentang 0 hingga 1, sehingga dapat meningkatkan stabilitas dan akurasi model pembelajaran mesin yang digunakan.

D. Seleksi Fitur

Seleksi Seleksi fitur dilakukan menggunakan metode SelectKBest dengan fungsi statistik Chi-Square (χ^2) untuk mengukur ketergantungan antara setiap fitur dengan variabel target (label). Meskipun metode Chi-Square secara umum lebih sesuai untuk data kategorikal, dalam implementasinya, library Scikit-Learn mengakomodasi fitur numerik dengan mengubahnya menjadi distribusi frekuensi (histogram), sehingga perhitungan tetap valid secara statistik. Dari seluruh fitur yang tersedia, dipilih 10 fitur terbaik yang paling relevan terhadap label. Pemilihan ini bertujuan untuk meningkatkan efisiensi model, mengurangi risiko overfitting, serta mempercepat waktu pelatihan. Selain tingkat akurasinya yang dapat diandalkan, metode Chi-Square juga dikenal cepat dan ringan secara komputasi, menjadikannya sangat sesuai untuk menangani dataset berukuran besar[5], [7].

E. Data Spliting

Setelah proses seleksi fitur rampung, data dibagi menjadi dua kategori, yakni data latih (training set) dan data uji (testing set) dengan rasio 80% untuk pelatihan dan 20% untuk pengujian [14]. Pembagian ini dilakukan dengan menggunakan fungsi train_test_split dari pustaka Scikit-learn dengan parameter stratify=y untuk memastikan distribusi kelas antara data pelatihan dan data pengujian tetap seimbang. [15] Teknik stratifikasi ini penting dalam konteks deteksi serangan DDoS karena menjaga keseimbangan antara kelas serangan dan kelas normal (BENIGN), sehingga menghindari bias model terhadap kelas mayoritas. Pembagian data secara acak juga dilakukan dengan menetapkan random_state=42 untuk menjamin reprodusibilitas eksperimen.

F. Pelatihan Model SVM

Setelah melalui tahap preprocessing dan seleksi fitur, data dilanjutkan ke proses pelatihan model menggunakan algoritma *Support Vector Machine* (SVM). Algoritma SVM dipilih karena memiliki keunggulan dalam melakukan klasifikasi terhadap data berdimensi tinggi dan mampu membentuk hyperplane optimal yang memisahkan kelas secara maksimal [1]. Dalam konteks penelitian, SVM digunakan untuk membedakan antara trafik jaringan normal (BENIGN) dan trafik serangan (DDoS). Model dilatih menggunakan kernel *Radial Basis Function* (RBF) karena kernel ini efektif dalam menangani masalah klasifikasi non-linear, di mana data tidak dapat dipisahkan secara linear di ruang aslinya. Kernel RBF mengubah ruang fitur asli menjadi ruang berdimensi lebih tinggi menggunakan fungsi Gaussian, sehingga data dari kelas berbeda dapat dipisahkan secara optimal oleh hyperplane.

G. GridSearchCV

Untuk mengoptimalkan performa model, dilakukan proses tuning hyperparameter menggunakan metode *GridSearchCV* dari pustaka *Scikit-learn*. Proses ini mengevaluasi kombinasi parameter secara menyeluruh dengan menggunakan teknik *cross-validation* untuk mendapatkan parameter yang menghasilkan akurasi dan generalisasi terbaik. Adapun parameter utama yang diuji adalah sebagai berikut:

- ***C:*** Parameter regulasi yang mengatur trade-off antara maksimum margin dan sanksi untuk kesalahan klasifikasi. Nilai C yang besar akan berusaha mengklasifikasikan seluruh data pelatihan dengan tepat, tetapi berisiko terjadinya overfitting.

- ***gamma:*** Parameter kernel RBF yang mengontrol jangkauan pengaruh satu titik pelatihan terhadap lainnya. Nilai gamma yang rendah akan menghasilkan model yang lebih umum, sementara nilai yang tinggi dapat menangkap pola lokal tetapi cenderung mengalami overfitting.
- ***kernel:*** Fungsi pemetaan dari ruang fitur ke ruang dimensi tinggi. Dalam penelitian ini digunakan kernel 'rbf' karena mampu menangani data dengan distribusi kompleks.

GridSearchCV melakukan pencarian parameter terbaik dengan cara menguji beberapa kombinasi dari nilai ***C*** dan ***gamma***, serta mengevaluasi kinerjanya menggunakan *stratified 3-fold cross-validation*. Hasil dari GridSearch kemudian digunakan untuk melatih ulang model SVM dengan parameter optimal, sehingga model akhir yang diperoleh memiliki performa yang lebih baik dari segi akurasi dan *False Positive Rate* (FPR) yang rendah.

H. Evaluasi dan Visualisasi Hasil

Evaluasi Evaluasi kinerja dilakukan untuk mengukur efektivitas model dalam mendeteksi serangan DDoS. Berbagai metrik evaluasi yang dipakai meliputi akurasi, presisi, recall, F1-score, dan skor ROC-AUC. Metrik dipilih karena dapat memberikan gambaran komprehensif tentang kinerja model, khususnya dalam konteks klasifikasi biner dengan distribusi kelas yang tidak seimbang [16]. Akurasi menilai seberapa banyak prediksi yang tepat dibandingkan dengan total data pengujian. Namun, pada data tidak seimbang, metrik ini kurang representatif. Oleh karena itu, presisi dan recall digunakan untuk memberikan evaluasi yang lebih spesifik. Presisi mengukur jumlah prediksi positif yang akurat, sedangkan recall menilai jumlah serangan yang berhasil terdeteksi oleh model. F1-score kemudian digunakan sebagai harmonisasi antara presisi dan recall. Rumus masing-masing metrik dapat dituliskan sebagai berikut.

1) Matrik Evaluasi Kinerja

- Accuracy

  Pada Formula (1), Accuracy Score Mengukur seberapa besar proporsi dari seluruh prediksi yang dilakukan model yang benar.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

  Akurasi cocok untuk melihat performa keseluruhan model, namun bisa menyesatkan jika distribusi kelas tidak seimbang (misal jumlah BENIGN jauh lebih banyak daripada DDoS).

- Precision

  Pada Formula (2), Precision (Presisi) Mengukur ketepatan prediksi terhadap kelas positif (DDoS). Nilai precision yang tinggi menunjukkan bahwa model jarang salah mengklasifikasikan trafik normal sebagai serangan (false positive rendah).

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

  Presisi sangat penting dalam konteks sistem keamanan untuk menghindari alarm palsu (false alert).

- Recall

  Pada Formula (3). Recall (Sensitivity) Mengukur kemampuan model untuk mendeteksi semua kasus serangan yang sebenarnya ada.

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

  Recall yang tinggi berarti semua serangan berhasil dideteksi, yang sangat krusial dalam sistem pertahanan jaringan.

- F1-Score

  Pada Formula (4), F1-Score adalah rata-rata harmonis antara precision dan recall. F1-score bermanfaat saat diperlukan keseimbangan di antara keduanya, terutama ketika data tidak seimbang.

$$F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (4)$$

- ROC-AUC

  Metrik penilaian yang menilai kemampuan model dalam membedakan antara kelas positif dan kelas negatif. ROC (*Receiver Operating Characteristic*) yang dapat menggambarkan hubungan antara True Positive Rate (tingkat keberhasilan mendeteksi serangan) dan False Positive Rate (tingkat kesalahan mengklasifikasikan trafik normal sebagai serangan) [17]. AUC (*Area Under the Curve)* menunjukkan luas area di bawah kurva ROC, dengan nilai mendekati 1 berarti model sangat baik dalam membedakan kedua kelas tersebut. Nilai AUC yang tinggi mencerminkan model yang andal, bahkan ketika threshold klasifikasi diubah-ubah, dan sangat penting dalam sistem keamanan untuk mencapai keseimbangan antara deteksi serangan yang akurat dan minimnya alarm palsu.

2) Confusion Matrix

Untuk mendapatkan gambaran nyata dari prediksi model terhadap data uji, digunakan confusion matrix seperti Tabel 2 TP (*True Positive*) adalah jumlah kasus serangan DDoS yang berhasil dideteksi dengan benar oleh model, sedangkan TN (*True Negative*) menunjukkan jumlah trafik normal (BENIGN) yang juga dikenali secara tepat sebagai bukan serangan. Sebaliknya, FP (*False Positive*) adalah jumlah kasus BENIGN yang salah diklasifikasikan sebagai serangan DDoS, yang dapat menyebabkan alarm palsu, dan FN (*False Negative*) merupakan kasus serangan DDoS yang tidak terdeteksi oleh sistem, sehingga berisiko lolos tanpa penanganan.

TABEL II
CONFUSION MATRIX

	Prediksi BENIGN	**Prediksi DDoS**
BENIGN	TN	FP
DDoS	FN	TP

3) Visualisasi Evaluasi Model

Untuk memperkuat pemahaman terhadap hasil evaluasi model, digunakan beberapa grafik seperti *Confusion Matrix Heatmap* yang menunjukkan jumlah klasifikasi benar dan salah antar kelas, ROC Curve yang menggambarkan kinerja model pada berbagai threshold, serta *Precision-Recall Curve*

yang bermanfaat untuk data tidak seimbang karena menunjukkan trade-off antara presisi dan sensitivitas. Visualisasi ini dibuat menggunakan pustaka Matplotlib dan Seaborn di platform Google Colab.

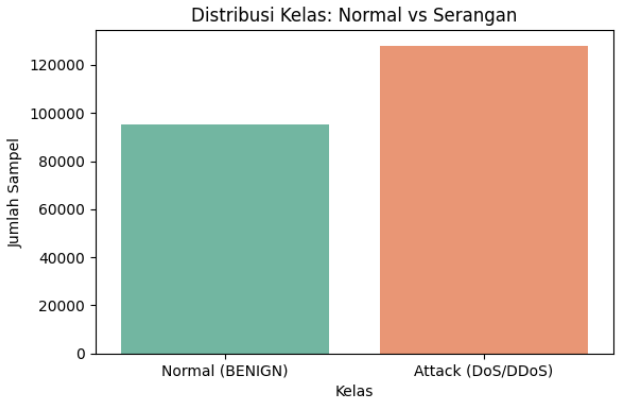
III. HASIL DAN PEMBAHASAN

Seluruh penelitian menerapakan algoritma Support Vector Machine (SVM) dengan pendekatan tuning hyperparameter dan evaluasi model melalui berbagai metrik performa. Dataset dimuat dari file Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv, dengan fokus pada empat label utama: BENIGN, DoS-Hulk, DoS GoldenEye, dan DDoS. Data yang dimuat dibersihkan dari nilai duplikat dan kosong (null value), kemudian dikonversi ke dalam format biner—label BENIGN diberi nilai 0 dan label serangan diberi nilai 1. Proses ini memastikan bahwa data siap digunakan untuk pelatihan dan pengujian model klasifikasi.

Fitur numerik dari dataset dinormalisasi menggunakan metode Min-Max Scaling. Untuk mengurangi dimensi dan meningkatkan efisiensi model, seleksi fitur dilakukan dengan metode Chi-Square melalui SelectKBest, yang memilih 10 fitur paling relevan, yang bertujuan mengurangi dimensi data, mempercepat pelatihan model, dan menghindari overfitting.

A. *Distribusi Kelas dalam Dataset*

Distribusi kelas pada dataset menunjukkan adanya ketidakseimbangan yang signifikan antara jumlah data serangan dan data normal. Berdasarkan analisis, trafik berlabel serangan (label 1) mencakup sebanyak 128.014 entri atau sekitar 57,38% dari keseluruhan data, sedangkan trafik normal (label 0) hanya berjumlah 95.068 entri atau sebesar 42,62%. Visualisasi distribusi ditampilkan pada Gambar 2. Distribusi Kelas Trafik Normal dan Serangan.

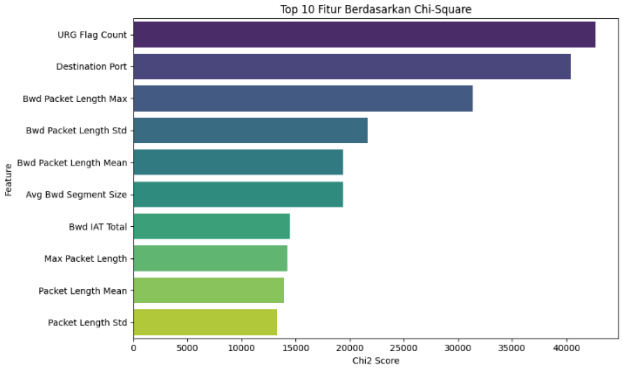


Gambar 2. Distribusi Kelas Trafik Normal dan Serangan.

Ketidakseimbangan distribusi pada Gambar 2 memiliki implikasi penting dalam proses evaluasi model. Dalam kondisi seperti ini, penggunaan metrik evaluasi yang lebih sensitif terhadap ketidakseimbangan data, seperti F1-score dan ROC AUC, menjadi sangat relevan dan lebih informatif dibandingkan hanya mengandalkan metrik akurasi, yang cenderung bias terhadap kelas mayoritas. Oleh karena itu, pemilihan metrik yang tepat menjadi aspek krusial dalam mengukur performa model secara objektif pada dataset yang tidak seimbang.

B. *Hasil Seleksi Fitur*

Proses seleksi fitur dilakukan menggunakan teknik SelectKBest dengan fungsi statistik Chi-Square (χ^2), yang digunakan untuk mengukur tingkat ketergantungan antara setiap fitur input dengan variabel target. Hasil seleksi ini menunjukkan bahwa terdapat 10 fitur teratas yang memiliki kontribusi paling signifikan terhadap proses klasifikasi. Sebagaimana ditunjukkan pada Gambar 3, dua fitur dengan skor tertinggi adalah *URG Flag Count* dan *Destination Port*, yang masing-masing menunjukkan nilai *Chi-Square* sebesar 42.675 dan 40.419.



Gambar 3. Top 10 Fitur Berdasarkan Nilai Chi-Square

Fitur-fitur ini memiliki hubungan statistik yang sangat kuat terhadap label target, sehingga sangat relevan dalam proses deteksi anomali atau serangan. Disusul di bawahnya adalah fitur-fitur seperti *Bwd Packet Length Max, Bwd Packet Length Std*, dan *Bwd Packet Length Mean*, yang juga menunjukkan tingkat signifikansi tinggi. Visualisasi pada Gambar 3 membantu memperlihatkan distribusi skor Chi-Square secara lebih intuitif, di mana panjang bar menunjukkan besarnya kontribusi masing-masing fitur. Sementara itu, pada Tabel 3 merangkum lima fitur dengan skor tertinggi yang menjadi kandidat kuat dalam pemodelan klasifikasi.

TABEL III
LIMA FITUR PALING SIGNIFIKAN BERDASARKAN NILAI CHI-SQUARE

No	Nama Fitur	Chi² Score
1	URG Flag Count	42,675.66
2	Destination Port	40,419.19
3	Bwd Packet Length Max	31,370.21
4	Bwd Packet Length Std	21,674.83
5	Bwd Packet Length Mean	19,935.83

C. *Data Spliting/Pembagian Data*

Setelah penyelesaian proses preprocessing dan pemilihan fitur, data dibagi menjadi dua bagian melalui train_test_split dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Teknik stratifikasi diterapkan untuk menjamin bahwa pembagian kelas tetap seimbang di kedua subset. Tujuan dari proses pembagian data ini adalah untuk memungkinkan

model SVM dilatih dengan maksimal dan diuji secara adil menggunakan data yang belum pernah dilihat sebelumnya, agar dapat mengevaluasi generalisasi model dengan tepat.

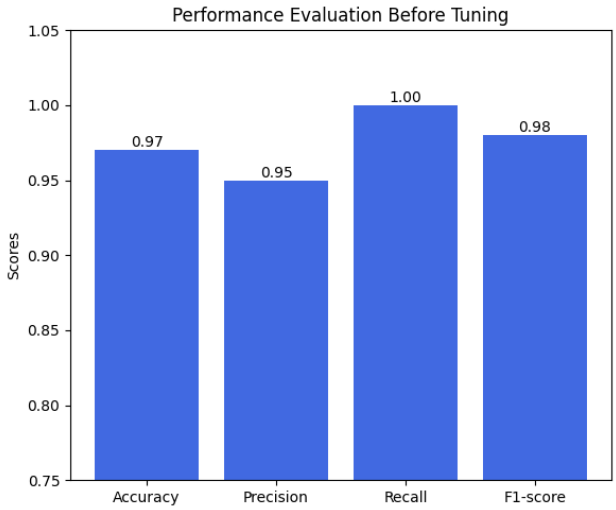
D. Hasil Evaluasi Model Sebelum Tuning GridSearchCV

Model SVM pertama kali dilatih menggunakan parameter default dengan kernel RBF (Radial Basis Function). Hasil evaluasi awal menunjukkan bahwa model ini sudah memiliki performa yang baik, dengan akurasi sebesar 97%, precision sebesar 0.95, recall 1.00, dan f1-score 0.98. Hal ini dapat dilihat pada Tabel 4.

TABEL IV
EVALUASI BEFORE TUNING GRIDSEARCHCV

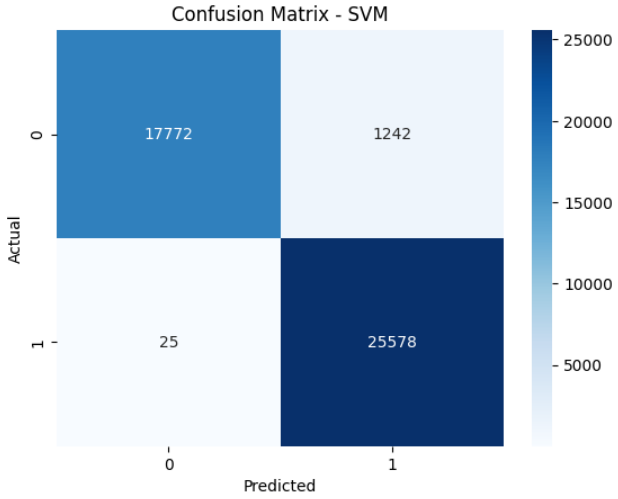
Accuracy	Precision	Recall	F1-score
0.97	0.95	1.00	0.98

Gambar 4 menunjukkan bahwa sebelum proses tuning, model SVM sudah cukup baik dalam membedakan antara trafik normal dan serangan, meskipun masih ada kesempatan untuk perbaikan, terutama terkait dengan presisi.



Gambar 4. Grafik Before Tuning

Sebagaimana ditunjukkan pada Gambar 5 terdapat 1.242 false positive (prediksi serangan padahal normal) dan 25 false negative (prediksi normal padahal serangan). Meskipun recall sudah mencapai angka maksimal, jumlah false positive yang tinggi menunjukkan bahwa model masih perlu dioptimalkan agar dapat lebih tepat dalam mengidentifikasi trafik normal.



Gambar 5. Confusion Matrix Before Tuning

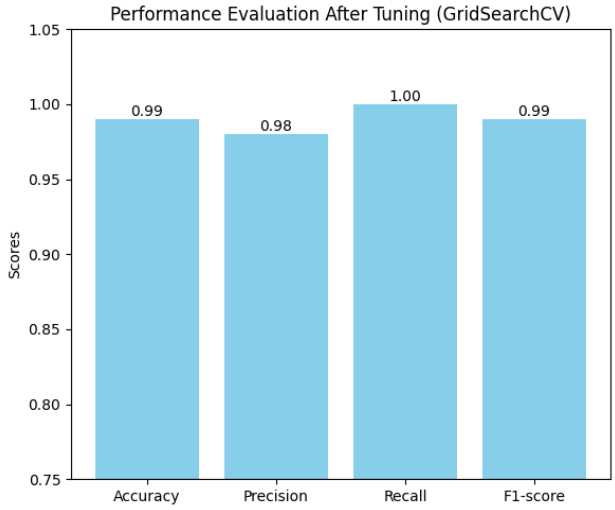
E. Hasil Evaluasi Model Setelah Tuning GridSearchCV

Untuk meningkatkan performa model, dilakukan tuning hyperparameter menggunakan GridSearchCV. Hasil dapat dilihat secara jelas pada Gambar 6 Parameter yang diuji meliputi nilai C, gamma, dan kernel.

TABEL V
EVALUASI AFTER TUNING GRIDSEARCHCV

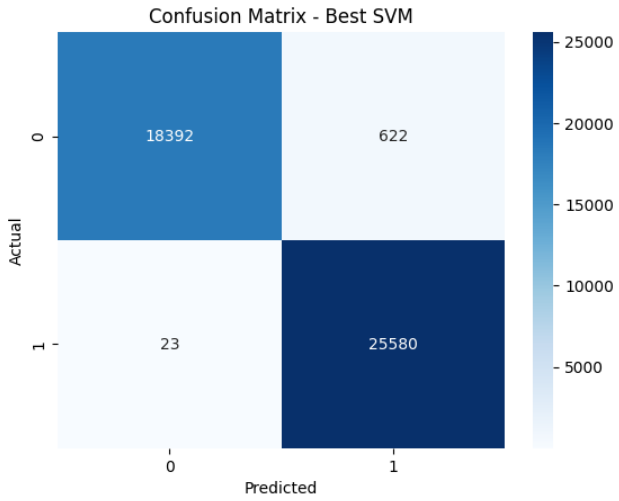
Accuracy	Precision	Recall	F1-score
0.99	0.97	1.00	0.99

Setelah proses tuning, diperoleh konfigurasi terbaik yang mampu meningkatkan metrik performa terdapat pada Gambar 6. Untuk meningkatkan performa model, dilakukan tuning dengan menguji berbagai kombinasi nilai C, gamma, dan kernel, seperti yang ditunjukkan pada Gambar 6. Konfigurasi terbaik yang berhasil meningkatkan performa model, ditunjukkan dengan peningkatan akurasi menjadi 0.99, precision menjadi 0.98, F1-score menjadi 0.99, serta recall yang tetap tinggi di angka 1.00.



Gambar 6. Grafik After Tuning

Model SVM hasil tuning menunjukkan peningkatan nyata dalam kemampuan mendeteksi serangan tanpa mengorbankan presisi terhadap trafik normal. Sebagaimana ditunjukkan pada Gambar 7 jumlah false positive berkurang dari 1.242 menjadi 622, dan false negative menurun dari 25 menjadi 23, yang menunjukkan bahwa model tidak hanya mampu mendeteksi hampir seluruh serangan, tetapi juga menjadi lebih presisi dalam membedakan antara serangan dan trafik normal.



Gambar 7. Confusion Matrix After Tuning

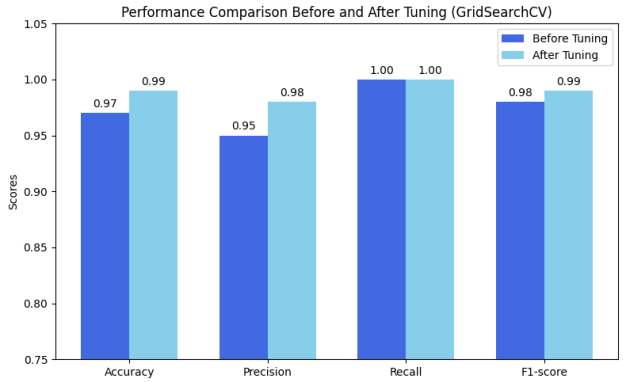
F. *Perbandingan Kinerja Sebelum dan Sesudah Tuning*

Perbandingan Untuk mempermudah analisis visual, perbandingan performa model sebelum dan sesudah tuning terdapat pada Tabel 6. Hasil evaluasi menunjukkan peningkatan performa yang konsisten di seluruh metrik setelah dilakukan tuning hyperparameter menggunakan GridSearchCV. Akurasi meningkat dari 0,97 menjadi 0,99, precision dari 0,95 menjadi 0,98, dan F1-score dari 0,98 menjadi 0,99. Sementara itu, nilai recall tetap sempurna di angka 1,00, menunjukkan bahwa model tetap mampu mendeteksi seluruh kasus serangan tanpa kehilangan sensitivitas.

TABEL VI
PERBANDINGAN BEFORE VS AFTER TUNING

	Before Tuning	**After Tuning GridSearchCV**
Accuracy	0.97	0.99
Precision	0.95	0.98
Recall	1.00	1.00
F1-score	0.98	0.99

Peningkatan nilai F1-score menandakan bahwa model menjadi semakin seimbang dalam menangani precision dan recall, yang sangat penting pada data yang tidak seimbang. Terdapat hasil visualisasi secara keseluruhan terdapat pada Gambar 8, hasil memperkuat bahwa proses tuning tidak hanya meningkatkan ketepatan klasifikasi, tetapi juga mengoptimalkan keseimbangan dan ketahanan model terhadap kesalahan prediksi, menjadikannya lebih andal untuk diterapkan pada sistem deteksi serangan siber.



Gambar 8. Grafik Perbandingan Before vs After Tuning

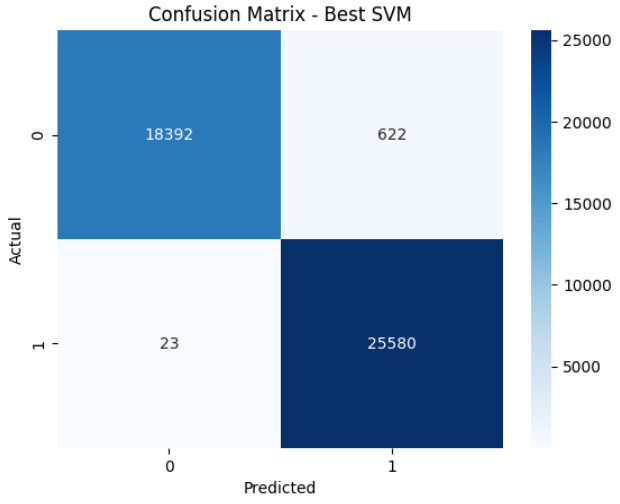
G. *Classification Report dan Confusion Matrix*

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	19014
1	0.98	1.00	0.99	25603
accuracy			0.99	44617
macro avg	0.99	0.98	0.99	44617
weighted avg	0.99	0.99	0.99	44617

Gambar 9. Clasification Report

Hasil klasifikasi divisualisasikan dalam confusion matrix dan classification report ditunjukan Pada Gambar 9 memperlihatkan bahwa model memiliki presisi sempurna (1.00) pada label BENIGN, dan recall sempurna (1.00) pada label serangan. Nilai-nilai tersebut menunjukkan bahwa model sangat minim kesalahan dalam mengidentifikasi serangan (false negative rendah) dan sangat akurat dalam mengklasifikasi trafik normal (false positive rendah).



Gambar 10. Confusion Matrix

Gambar 10 menunjukkan bahwa dari model SVM terbaik setelah proses tuning. Model berhasil mengklasifikasikan 18.392 data kelas negatif dan 25.580 data kelas positif secara benar, dengan jumlah kesalahan klasifikasi yang relatif kecil, yaitu 622 false positive dan 23 false negative. Peneilitian menunjukkan bahwa model menunjukkan kinerja klasifikasi yang sangat baik, terutama dalam mengidentifikasi kelas positif.

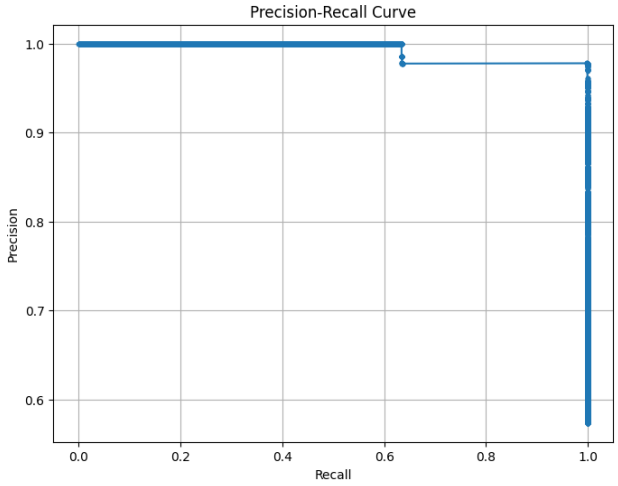
H. Evaluasi Overfitting

Proses evaluasi model dilakukan menggunakan teknik cross-validation sebanyak 5-fold untuk menguji kemampuan generalisasi model Support Vector Machine (SVM). Hasilnya menunjukkan akurasi rata-rata sebesar 97,13%, yang mengindikasikan performa model yang konsisten dan stabil pada data uji yang berbeda. Tidak ditemukan indikasi overfitting yang signifikan, karena selisih performa antara data latih dan data validasi tetap dalam batas wajar. Dari sisi efisiensi waktu, pelatihan model SVM memerlukan waktu sebesar 1.207 detik, sementara proses hyperparameter tuning menggunakan GridSearchCV membutuhkan waktu yang jauh lebih lama, yakni 23.562 detik.

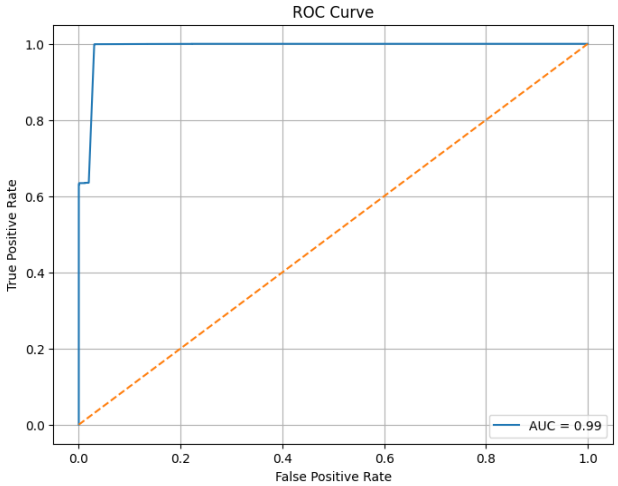
Meskipun proses tuning relatif mahal secara komputasi, hasil yang diperoleh menunjukkan peningkatan performa model secara keseluruhan. Hal ini menegaskan bahwa SVM, meskipun memiliki waktu pelatihan yang intensif, mampu menghasilkan model yang akurat dan cepat dalam inferensi, sehingga tetap layak untuk diterapkan pada skenario deteksi serangan yang memerlukan presisi tinggi. Meskipun SVM membutuhkan waktu pelatihan yang relatif lebih lama, terutama dengan kernel RBF dan dataset besar, waktu inference (prediksi) cenderung cepat dan efisien. Oleh karena itu, model ini cocok untuk sistem IDS yang telah dilatih sebelumnya dan hanya digunakan untuk inferensi. Penelitian lanjutan dapat mempertimbangkan kernel approximation atau linear SVM untuk mempercepat pelatihan.

I. Visualisasi ROC Curve

Setelah model terbaik ditemukan, dilakukan evaluasi kinerja model menggunakan metrik seperti akurasi, confusion matrix, dan classification report. Di samping itu, penggunaan visualisasi *ROC Curve* dan *Precision-Recall Curve* bertujuan untuk menganalisis kinerja model dalam mendeteksi serangan. Pada Gambar 11, ditampilkan kurva Precision-Recall yang menunjukkan bahwa model mampu mempertahankan tingkat presisi yang tinggi hampir di seluruh rentang recall. Hal ini menandakan bahwa model sangat efektif dalam mengidentifikasi kelas positif dengan tingkat kesalahan yang minimal (false positive rendah), yang sangat penting dalam konteks dataset dengan distribusi kelas yang tidak seimbang. Sementara itu, pada Gambar 12, ditampilkan Receiver Operating Characteristic (ROC) Curve yang memperlihatkan performa klasifikasi model dengan nilai Area Under Curve (AUC) sebesar 0.99. Nilai ini mengindikasikan bahwa model memiliki kemampuan sangat baik dalam membedakan antara kelas positif dan negatif, serta dalam meminimalkan kemungkinan kesalahan klasifikasi (false positive maupun false negative).



Gambar 11. Kurva Precision-Recall



Gambar 12. Kurva ROC

IV. KESIMPULAN

Penelitian ini berhasil mengevaluasi performa algoritma klasifikasi Support Vector Machine (SVM) dalam mendeteksi serangan Distributed Denial of Service (DDoS) pada lalu lintas jaringan dengan menggunakan dataset CICIDS2017. Dataset dipilih karena memuat berbagai jenis serangan DDoS dengan dominasi yang tinggi, seperti DoS Hulk, DoS GoldenEye, dan DDoS. Penelitian ini terbatas pada klasifikasi biner (0 = normal, 1 = serangan), sehingga belum mencakup pendekatan multiclass. Proses yang dilakukan mencakup tahapan praproses data, normalisasi fitur numerik, serta seleksi fitur menggunakan metode Chi-Square (SelectKBest). Hasil seleksi menunjukkan bahwa fitur seperti URG Flag Count, Destination Port, dan Bwd Packet Length Max memiliki kontribusi signifikan terhadap klasifikasi. Mengingat distribusi kelas yang tidak seimbang (serangan

57,38% dan trafik normal 42,62%), metrik evaluasi seperti F1-score dan ROC-AUC digunakan untuk melengkapi analisis akurasi. Evaluasi dilakukan sebelum dan sesudah tuning hyperparameter dengan GridSearchCV, menunjukkan peningkatan performa dari akurasi 97% menjadi 99% dan F1-score dari 0,98 menjadi 0,99. Validasi silang 5-fold menghasilkan akurasi rata-rata 97,13%, memperkuat konsistensi model. Namun, proses pelatihan membutuhkan waktu cukup lama (1207 detik untuk SVM dan 23562 detik untuk tuning), menunjukkan tingginya beban komputasi.

Model SVM yang dikembangkan menunjukkan potensi tinggi untuk diimplementasikan dalam sistem IDS berbasis machine learning, baik melalui REST API (misalnya dengan Flask/FastAPI) maupun integrasi ke IDS populer seperti Snort dan Suricata. Rekomendasi untuk penelitian selanjutnya meliputi perluasan ke klasifikasi multiclass, eksplorasi metode seleksi fitur lain (seperti RFE, mutual information, dan embedded method), serta pengujian langsung dalam lingkungan IDS real-time. Selain itu, pengaplikasian teknik seperti kernel approximation, downsampling, serta metode balancing data seperti SMOTE atau ADASYN juga disarankan untuk meningkatkan efisiensi dan sensitivitas model terhadap kelas minoritas.

DAFTAR PUSTAKA

[1] D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 81–90, Apr. 2021, doi: 10.48161/qaj.v1n2a50.

[2] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th InternationalConference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications,2018, pp. 108–116. doi: 10.5220/0006639801080116.

[3] V. Kathiresan, V. Yendapalli, J. Bhuvana, and E. Daniel, "Machine Learning-Based DDoS Attack Detection Using Support Vector Machine," in *Artificial Intelligence and Cyber Security in Industry 4.0*, V. Sarveshwaran, J. I.-Z. Chen, and D. Pelusi, Eds., in Advanced Technologies and Societal Change. , Singapore: Springer Nature Singapore, 2023, pp. 329–341. doi: 10.1007/978-981-99-2115-7_15.

[4] B. Goparaju and D. B. S. Rao, "A DDoS Attack Detection usingPCA Dimensionality Reduction and Support Vector Machine".

[5] A. Maslan, K. M. B. Mohamad, A. Hamid, H. Pangaribuan, and S. Sitohang, "Feature Selection to Enhance DDoS Detection Using Hybrid N-Gram Heuristic Techniques," *JOIV Int. J. Inform. Vis.*, vol. 7, no. 3, pp. 815–822, Sep. 2023, doi: 10.30630/joiv.7.3.1533.

[6] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDOS) attack detection using supervised machine learning algorithms," *Sci. Rep.*, vol. 15, no. 1, p. 13098, Apr. 2025, doi: 10.1038/s41598-024-84879-y.

[7] M. S. Raza, M. N. A. Sheikh, I.-S. Hwang, and M. S. Ab-Rahman, "Feature-Selection-Based DDoS Attack Detection Using AI Algorithms," *Telecom*, vol. 5, no. 2, pp. 333–346, Apr. 2024, doi: 10.3390/telecom5020017.

[8] S. Mohammed Fayadh, "Hybrid Machine Learning Model for Feature Selection in DDoS Attack Detection in Cloud Environments Using Convolutional Neural Networks and Genetic Algorithms," *Wasit J. Pure Sci.*, vol. 4, no. 1, pp. 94–103, Mar. 2025, doi: 10.31185/wjps.616.

[9] H. Patel, "Feature Selection via GANs (GANFS): EnhancingMachine Learning Models for DDoS Mitigation," Apr. 21, 2025, *arXiv*: arXiv:2504.18566. doi: 10.48550/arXiv.2504.18566.

[10] A. Sanmorino, R. Gustriansyah, and J. Alie, "DDoS AttacksDetection Method Using Feature Importance and Support Vector Machine," *JUITA J. Inform.*, vol. 10, no. 2, p. 167, Nov. 2022, doi: 10.30595/juita.v10i2.14939.

[11] A. Hamarshe, H. I. Ashqar, and M. Hamarsheh, "Detec%onofDDoS A,acks in So1ware Defined Networking Using Machine Learning Models".

[12] D. K. Suvra, "An Efficient Real Time DDoS Detection Model Using Machine Learning Algorithms," Jan. 24, 2025, *arXiv*: arXiv:2501.14311. doi: 10.48550/arXiv.2501.14311.

[13] P. B. Yakubu, E. Owusu, L. Santana, M. Rahouti, A. Chehri, and K. Xiong, "Exploring Feature Importance and Explainability Towards Enhanced ML-Based DoS Detection in AI Systems," Nov. 04,2024, *arXiv*: arXiv:2411.03355. doi: 10.48550/arXiv.2411.03355.

[14] M. B. Anggara, "Perbandingan Naïve Bayes Dan Svm Dalam Analisis Sentimen Ulasan Aplikasi Rsud Al Ihsan Mobile," vol. 20, 2025.

[15] Q. H. Nguyen *et al.*, "Influence of Data Splitting on Performanceof Machine Learning Models in Prediction of Shear Strength of Soil," *Math. Probl. Eng.*, vol. 2021, pp. 1–15, Feb. 2021, doi: 10.1155/2021/4832864.

[16] B. J. Erickson and F. Kitamura, "Magician's Corner: 9. Performance Metrics for Machine Learning Models," *Radiol. Artif. Intell.*, vol.3, no. 3, p. e200126, May 2021, doi: 10.1148/ryai.2021200126.

[17] A. M. Carrington *et al.*, "Deep ROC Analysis and AUC as Balanced Average Accuracy to Improve Model Selection, Understanding and Interpretation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.45,no. 1, pp. 329–341, Jan. 2023, doi: 10.1109/TPAMI.2022.3145392.