Jurnal Informatika Universitas Pamulang **ISSN: 2541-1004**
Penerbit: Fakultas Ilmu Komputer Universitas Pamulang **e-ISSN: 2622-4615**
Vol.10, No. 2, Juni 2025 (75-85) **https://doi.org/10.32493/informatika.v10i2.52163**

Implementasi Sistem Deteksi Anomali Berbasis Jaringan Menggunakan *CNN* dan *SVM* untuk Klasifikasikan Data Secara *Real-time*

Arief Luqman Hadiyani[1*], Bana Handaga[2]

[1]Magister Informatika, Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Jawa Tengah 57162
e-mail: [1]l208230009@student.ums.ac.id, [2]bana@ums.ac.id
*Corresponding author

Submitted Date: March 30th, 2025 Reviewed Date: April 15th, 2025
Revised Date: May 5th, 2025 Accepted Date: June 20th, 2025

Abstract

The growing volume and complexity of network traffic have created new challenges in maintaining information security. Conventional signature-based intrusion detection systems are inadequate against modern threats, especially zero-day attacks that remain undocumented. Anomaly-based approaches using classical machine learning methods such as Support Vector Machine (SVM) show promise but still rely on manual feature engineering, which is time-consuming and requires expertise. This study proposes an anomaly detection system combining the automatic feature extraction capability of Convolutional Neural Network (CNN) with the strong classification performance of SVM. The NSL-KDD dataset is used for training, while real-time testing data are captured using Scapy. The system updates its analysis every five minutes, and detection results are presented as graphical reports and log tables sent to administrators via a Telegram Bot. Experimental results show that the hybrid CNN–SVM model achieves high accuracy and stable performance in real-time scenarios, contributing to more adaptive and intelligent intrusion detection.

Keywords: Network Anomaly Detection; Computer Network; CNN–SVM; NSL-KDD

Abstrak

Meningkatnya volume sekaligus kompleksitas lalu lintas jaringan telah menghadirkan tantangan baru dalam menjaga keamanan informasi. Sistem deteksi intrusi konvensional yang berbasis tanda tangan terbukti tidak memadai dalam menghadapi ancaman modern, khususnya serangan *zero-day* yang belum terdokumentasi sebelumnya. Upaya berbasis anomali melalui *algoritma machine* learning klasik, seperti *Support Vector Machine (SVM)* memang menjanjikan, tetapi masih terkendala proses rekayasa fitur manual yang memerlukan waktu serta keahlian khusus. Berangkat dari masalah tersebut, penelitian ini merancang sistem deteksi anomali dengan memadukan kemampuan *Convolutional Neural Network (CNN)* dalam mengekstraksi fitur otomatis dan kekuatan klasifikasi *SVM*. Dataset *NSL-KDD* digunakan untuk tahap pelatihan, sementara data uji diperoleh secara *real-time* melalui perekaman paket dengan *Scapy*. Sistem ini dirancang untuk memperbarui analisis setiap lima menit, dan hasil deteksi ditampilkan dalam bentuk laporan grafik serta tabel *log* yang dikirimkan ke *administrator* melalui *Telegram Bot*. Pengujian menunjukkan bahwa pendekatan hibrida *CNN–SVM* mampu mengenali pola anomali dengan akurasi yang baik dan bekerja stabil pada skenario *real-time*. Dengan demikian, penelitian ini memberikan kontribusi pada pengembangan sistem deteksi instrusi yang lebih adaptif terhadap dinamika ancaman siber.

Kata Kunci: Deteksi Anomali Jaringan; Komputer Jaringan; CNN-SVM; NSL-KDD

ISSN: 2541-1004
e-ISSN: 2622-4615
https://doi.org/10.32493/informatika.v10i2.52163

1. Pendahuluan

Perkembangan teknologi digital dan internet telah secara dramatis meningkatkan *volume* dan kompleksitas lalu lintas jaringan, secara bersamaan membuka celah bagi berbagai ancaman siber (Irfan, Nusri, Rachmat, & Wulandari, 2024). Sistem Deteksi Intrusi berbasis *network* menjadi komponen penting dalam arsitektur keamanan jaringan komputer untuk mengidentifikasi aktivitas berbahaya (Lateef, Al-Janabi, & Al-Khateeb, 2019). IDS tradisional yang mengandalkan basis data tanda tangan *(signature-based)* efektif untuk serangan yang telah diketahui, namun memiliki kelemahan signifikan dalam mendeteksi ancaman baru atau zero-day attack (Khraisat, Gondal, Vamplew, Kamruzzaman, & Alazab, 2019). Kelemahan ini mendorong pengembangan IDS berbasis anomali *(anomaly-based),* yang bertujuan mengenali pola-pola yang menyimpang dari perilaku jaringan normal (Firdaus, Fahira, & Rianti, 2023).

Beberapa tahun terakhir, penerapan *machine learning (ML)* telah menunjukkan hasil yang menjanjikan untuk deteksi anomali. Berbagai model ML klasik telah dieksplorasi, salah satunya adalah *Support Vector Machine* (*SVM*) yang dikenal andal dalam masalah klasifikasi karena kemampuannya menemukan *hyperplane* pemisah yang optimal (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020). Berbagai penelitian di Indonesia juga telah membuktikan kapabilitas *Support Vector Machine* (*SVM*) sebagai metode yang efektif untuk melakukan klasifikasi serangan pada lalu lintas jaringan komputer. Penelitian lain juga telah menerapkan *SVM* untuk klasifikasi serangan spesifik seperti *DDoS* dengan hasil yang memuaskan. Meskipun demikian, model-model klasik ini seringkali bergantung pada proses rekayasa fitur (feature engineering) manual yang tidak hanya memakan waktu tetapi juga memerlukan keahlian domain yang mendalam untuk memilih fitur-fitur yang paling relevan (Ngurah, Dika, Agung, & Arya, 2025).

Untuk mengatasi tantangan rekayasa fitur, pendekatan *deep learning* mulai diadopsi secara luas sebagai paradigma baru dalam keamanan siber, *CNN* yang awalnya populer di bidang pengenalan gambar, telah berhasil diadaptasi untuk tugas-tugas keamanan jaringan (Ogah, Essien, Ogharandukun, & Abdullahi, 2024). Keunggulan utama *CNN* terletak pada kemampuannya untuk belajar secara hierarkis dan mengekstrak fitur secara otomatis langsung dari data mentah. Beberapa penelitian telah memvalidasi metode konversi data paket jaringan mentah (dalam format .pcap) menjadi representasi gambar untuk kemudian diproses oleh *CNN*. Pendekatan serupa menggunakan beberapa jenis *Deep Learning (DL)* lain seperti *Autoencoder* juga terbukti efektif dalam mempelajari representasi data normal untuk mengidentifikasi penyimpanga (Sajid et al., 2024).

Menggabungkan keunggulan antara kemampuan klasifikasi *SVM* yang kuat dengan kemampuan ekstraksi fitur *CNN* yang otomatis telah menjadi arah penelitian yang aktif. Model *hybrid* semacam ini bertujuan untuk memaksimalkan performa dengan membiarkan setiap komponen melakukan tugasnya yang paling unggul (Berhane, Melese, Walelign, & Mohammed, 2023). Konsep *feature fusion* merupakan fitur yang diekstrak oleh model *DL* lalu ke *classifier ML* klasik yang menunjukkan peningkatan akurasi yang signifikan dalam berbagai kasus (Altunay & Albayrak, 2023). Pendekatan model ini telah berhasil diterapkan pada data jaringan berbasis file *PCAP dan log Snort* untuk membedakan lalu lintas normal dan anomali. Hasil ini juga menunjukkan bahwa pendekatan *hybrid deep learning* dalam konteks big data mampu mencapai akurasi yang sangat tinggi dalam mendeteksi serangan siber, khususnya pada sistem kendali industri dan infrastruktur vital (Jagtap, Shankar Sriram, & Subramaniyaswamy, 2021).

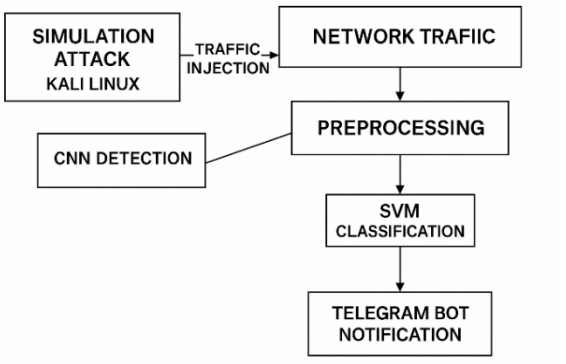
Penelitian relevan dalam bidang klasifikasi deteksi anomali pada lalu lintas jaringan komputer salah satunya berjudul *A novel model for anomaly detection in network traffic based on kernel Support Vector Machine.* Penelitian ini menganalisis penggunaan metode Kernel *SVM* untuk mendeteksi anomali pada lalu lintas jaringan. Penelitian tersebut menguji tiga dataset dan berhasil mencapai akurasi lebih dari 99% pada seluruh dataset menggunakan metode yang diusulkan (Ma, Sun, Cui, & Jin, 2021). Penelitian ini bertujuan melakukan deteksi anomali pada dataset yang diperoleh dari hasil rekaman lalu lintas jaringan menggunakan *snort* pada sebuah simulasi serangan menggunakan OS Kali linux dan ubuntu. Proses klasifikasi dilakukan dengan algoritma *hybrid CNN* dan *SVM*, sekaligus membandingkan hasilnya dengan dataset yang telah melalui tahap pra-pemrosesan berupa normalisasi data untuk mendeteksi paket anomali, serta transformasi data guna memperoleh performa model yang optimal (Al Ghamdi, 2023).

Intrusion Detection System (IDS) merupakan sistem yang dirancang untuk memantau dan mendeteksi aktivitas abnormal dalam jaringan atau sistem komputer guna mencegah akses tidak sah dan berbagai bentuk ancaman siber. Salah satu *IDS* yang populer, yaitu Snort yang memiliki kemampuan untuk menganalisis lalu lintas jaringan secara langsung *(real-time)* dan mengenali berbagai jenis ancaman, seperti *buffer overflow, port scanning*, hingga serangan *Denial of Service (DoS)* (Ozkan-Okay, Samet, Aslan, & Gupta, 2021). Penelitian ini mengimplementasikan pendekatan gabungan antara perangkat keras dan lunak dalam sebuah lingkungan jaringan komputer. Pada tahap awal pengujian, sistem diuji dengan mengintegrasikan fitur notifikasi melalui Telegram Bot guna memberi peringatan otomatis ketika terjadi serangan. Selanjutnya, dilakukan pengujian terhadap kemampuan Snort dalam mengenali berbagai serangan lain seperti *DoS*, percobaan login *FTP*, dan *port scanning*. Hasil evaluasi menunjukkan bahwa Snort mampu secara efektif mendeteksi beragam jenis serangan, termasuk *Port Scan, Ping of Death, FTP Brute Force, TCP SYN Flood*, hingga *Denial of Service*, bahkan ketika serangan berasal dari sistem operasi yang berbeda. Temuan ini mempertegas bahwa Snort merupakan solusi krusial dalam memperkuat pertahanan keamanan jaringan di tengah meningkatnya ancaman dunia digital saat ini (S, Wahyuddin, Kautsar, & Setyawan, 2025).

Penelitian ini menggunakan *scapy* sebagai sistem deteksi intrusi *(Intrusion Detection System/IDS)* untuk yang dikombinasikan dengan sistem notifikasi *real-time* melalui Telegram. scapy dikonfigurasi secara khusus dan diuji menggunakan simulasi serangan yang dilakukan oleh perangkat secara langsung *real-time* terhadap jaringan target. Integrasi deteksi peringatan secara *real-time* memberikan keunggulan dalam hal kecepatan respons terhadap ancaman keamanan siber, sehingga membantu mempercepat proses penanggulangan serangan. Secara keseluruhan, penelitian membuktikan bahwa kombinasi antara *Snort* dan sistem notifikasi otomatis merupakan solusi yang efektif dalam memperkuat sistem keamanan jaringan serta mempercepat mitigasi ancaman secara aktif dan efisien (Januantoro, Scanning, Scanning, & Force, 2025). Penelitian ini melakukan klasifikasi deteksi anomali pada dataset yang dihasilkan oleh *capture scapy* pada sebuah komputer secara *real-time* dengan *report* data setiap 5 menit dengan menggunakan model algoritma *CNN* dan *SVM* sehingga menghasilkan data deteksi anomali jaringan. Kebaruan dari penelitian sebelumnya adalah menghasilkan model *machine learning CNN-SVM* dalam mengidentifikasi perilaku anomali disuatu lalu lintas jaringan komputer dengan melakukan pelatihan model terlebih dahulu menggunakan dataset *NSL-KDD* sehingga menciptakan model deteksi pola yang cerdas dan mandiri.

2. Metode Penelitian

Penelitian ini dilakukan melalui eksperimen berbasis program komputer dan pengembangan model menggunakan Python. Peralatan yang digunakan meliputi laptop atau PC dengan spesifikasi yang memadai untuk menjalankan program, Python dimanfaatkan untuk membangun model *hybrid* berbasis *SVM* dan *CNN*, Bot Telgeram sebagai *alert* secara *real-time*. Beberapa parameter jaringan yang digunakan dalam penelitian ini antara lain *timestamp, ip address src, ip address dst, protocol, Flag.* Tahap pengumpulan data dilakukan dengan proses melatih dataset *NSL-KDD* menggunakan model *CNN-SVM* untuk mengetahui pola serangan sehingga dapat mendeteksi lalu lintas jaringan yang mencurigakan atau anomaly, menggunakan *tools scapy*. Sistem diimplementasikan dalam mode *real-time*, dengan batch lalu lintas jaringan terbaru diproses setiap lima menit. Setiap interval inferensi menghasilkan laporan visual dan log, termasuk grafik top 5 IP sumber anomali, layanan target, flag TCP mencurigakan, serta tabel aktivitas anomali. Pendekatan ini meniru strategi pemrosesan streaming dan respon temporal dari model machine learning yang diaplikasikan pada deteksi anomali trafik jaringan. Setiap 5 menit data klasifikasi anomali diperoleh kemudian dibuat file laporan yang dikirimkan ke smartphone melalui telegram.

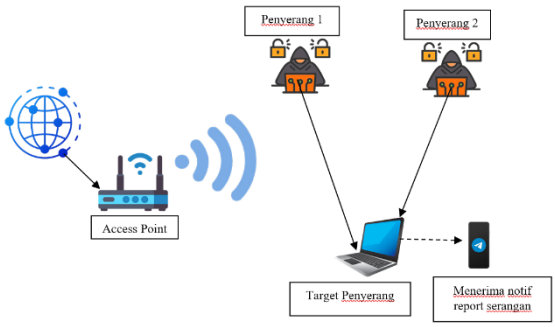


Gambar 1. Desain Model Deteksi Anomali

Alur pendekatan sistem deteksi anomali jaringan berbasis *deep learning & machine learning* dengan kombinasi model *CNN* dan *SVM*. Dalam Alur deteksi, terdapat perangkat utama: dua komputer penyerang dengan sistem operasi Kali Linux (PC 1 dan PC 2) dan satu komputer target dengan sistem operasi Ubuntu (PC 3). Kedua perangkat penyerang melakukan simulasi serangan terhadap PC target. Pada sisi target, sistem dijalankan menggunakan Python dan pustaka Scapy untuk melakukan perekaman dan ekstraksi lalu lintas jaringan secara *real-time*. Data hasil lalu lintas tersebut kemudian diproses oleh model deteksi yang menggabungkan *Convolutional Neural Network (CNN)* untuk ekstraksi fitur spasial dari data jaringan, dan *Support Vector Machine* (*SVM*) sebagai *classifier* untuk mengidentifikasi apakah lalu lintas tergolong normal atau anomali. Model *CNN-SVM* dilatih terlebih dahulu menggunakan dataset benchmark *NSL-KDD*, yang merupakan versi perbaikan dari dataset KDD'99, untuk meningkatkan akurasi dan menghindari duplikasi data pelatihan(Liu & Wang, 2023). Hasil deteksi anomali kemudian dikirimkan secara otomatis dalam bentuk notifikasi ke Telegram admin melalui API, sehingga administrator dapat memantau aktivitas mencurigakan secara cepat dan responsif menggunakan perangkat smartphone atau laptop.

2.1 Tahapan Simulasi

Tahapan penelitian meliputi langkah mendeteksi lalu lintas jaringan yang tidak wajar atau anomali, yaitu Perancangan Lingkungan Uji *(Testbed Environment)* Lingkungan pengujian terdiri dari tiga unit komputer, PC 1 dan PC 2 menggunakan sistem operasi Kali Linux sebagai mesin penyerang *(attacker).* PC 3 menggunakan sistem operasi Ubuntu sebagai mesin target serangan. PC attacker digunakan untuk menjalankan skenario serangan seperti port scanning, DDoS, dan serangan flooding menggunakan *tools* seperti hping3, nmap, dan slowloris. Sementara PC target menjalankan sistem pemantauan dan deteksi menggunakan Scapy yang terintegrasi dengan model *CNN-SVM*, tentunya scapy sudah dilatih menggunakan dataset *NSL-KDD* (Choi, Choi, & Seo, 2023).



Gambar 2. Tahapan Simulasi

Tahap pertama simulasi perancangan arsitektur sistem jaringan *DHCP* yang terdiri dari tiga buah komputer utama. Dua komputer berperan sebagai perangkat penyerang (PC 1 dan PC 2) yang menggunakan sistem operasi Kali Linux, dan satu komputer sebagai target serangan (PC 3) yang menggunakan sistem operasi Ubuntu. Perangkat penyerang mensimulasikan serangan terhadap PC target dengan berbagai jenis serangan seperti serangan *Denial of Service (DoS), port scanning,* dan serangan lainnya untuk menghasilkan lalu lintas jaringan anomali. Selanjutnya, pada tahap kedua dilakukan pengumpulan data lalu lintas jaringan yang terjadi pada komputer target. Proses ini menggunakan *tools* seperti Scapy untuk menangkap paket data secara *real-time*. Data yang diperoleh dikumpulkan dalam parameter *ip address src ip address dst, protocol, flag* dan diekstrak menjadi fitur-fitur numerik yang merepresentasikan karakteristik lalu lintas jaringan.

Tahap ketiga adalah ekstraksi fitur dan klasifikasi data. Data yang telah diekstrak kemudian diproses menggunakan metode *CNN* untuk mendapatkan representasi fitur yang lebih kompleks dan bermakna. *Output* dari *CNN* tidak langsung digunakan sebagai hasil akhir, namun diteruskan ke algoritma *SVM* untuk melakukan proses klasifikasi akhir. *SVM* digunakan karena kemampuannya dalam memisahkan kelas dengan margin maksimum dan bekerja baik pada data yang memiliki dimensi tinggi. Kombinasi *CNN-SVM* ini diharapkan mampu meningkatkan akurasi dalam mendeteksi lalu lintas jaringan yang bersifat anomali dibandingkan dengan metode konvensional. Pada tahap keempat, ketika sistem berhasil mendeteksi adanya anomali, maka akan dilakukan proses pengiriman notifikasi secara otomatis kepada admin melalui platform Telegram. Notifikasi ini berisi informasi waktu, jenis anomali,

dan sumber paket jaringan yang terdeteksi. Hal ini dimaksudkan agar proses monitoring dapat dilakukan secara *real-time* dan admin jaringan dapat segera melakukan tindakan mitigasi. Penerapan model algoritma *Hybrid CNN-SVM* ini yang memproses pelabelan jika terdapat kegiatan lalu lintas komputer yang tidak wajar dilakukan secara terus menerus di setiap port layanan jaringan komputer. Model ini mendeteksi dan melaporkan jumlah terjadinya serangan. Notifikasi serangan yang dibentuk dari hasil deteksi scapy kemudian menghasilkan laporan serangan dikirim melalui telegram dalam bentuk data yang sudah di label anomali beserta jenis layanan yang menjadi target serangan.

2.2 Jenis Serangan Target Layanan Jaringan

Target layanan yang menjadi titik serangan pada simulasi ini yaitu layanan *HTTP, FTP, SSH*, dalam penelitian ini, simulasi serangan terhadap layanan jaringan dilakukan menggunakan alat bantu bernama *Hydra (THC-Hydra).* Hydra merupakan salah satu tool populer dalam dunia keamanan siber yang digunakan untuk melakukan serangan brute force, yaitu serangan dengan cara mencoba berbagai kombinasi username dan password untuk mendapatkan akses tidak sah ke sistem. Beberapa layanan jaringan yang menjadi target dalam simulasi ini meliputi *SSH (Secure Shell), FTP (File Transfer Protocol), POP3, DNS, dan HTTP*. Pada layanan *SSH*, serangan *brute force* dilakukan untuk mensimulasikan upaya masuk ke *server* target melalui *port* 22 dengan mencoba kombinasi kredensial secara terus-menerus. Serangan ini menciptakan lalu lintas jaringan yang intensif dan mencurigakan, yang sangat relevan untuk pengujian sistem deteksi anomali.

Pada layanan *FTP*, serangan dilakukan untuk mengakses *file server* dengan memanfaatkan kombinasi daftar *username* dan *password* yang telah disiapkan. Sedangkan pada *HTTP*, Hydra digunakan untuk menyerang halaman login berbasis *web* dengan mencoba berbagai kemungkinan autentikasi, yang jika gagal akan menghasilkan respons "*Login failed*" dari *server*. Penggunaan Hydra dalam simulasi ini bertujuan untuk menghasilkan lalu lintas jaringan yang bersifat anomali secara realistis, sehingga dapat digunakan sebagai data pelatihan dan pengujian sistem deteksi anomali berbasis *CNN* dan *SVM*. Selain itu, simulasi ini juga menggambarkan skenario nyata yang umum terjadi dalam dunia keamanan jaringan, sehingga sistem yang dibangun dapat dievaluasi efektivitasnya dalam mengenali dan merespons serangan secara *real-time*.

Tabel 1. Jenis Target dan Port Layanan

Layanan	Port	Protokol
SSH	22	TCP
FTP	21	TCP
HTTP	80	TCP

2.3 Tahapan Ekstrasi, Preprocessing & Klasifikasi

Tahapan ini bertujuan untuk melatih model *CNN-SVM* menggunakan dataset *NSL-KDD* yang sudah dilabel untuk memberikan arahan agar nantinya proses deteksi melalui scapy punya referensi dalam menentukan klasifikasi data anomali sehingga model *CNN-SVM* dapat mandiri dan detail dalam mendeteksi jaringan komputer yang tidak wajar atau anomali dalam mengenali pola. Berikut beberapa tahapan ekstrasi, *preprocessing* dan klasifikasi (Alrayes, Zakariah, Amin, Khan, & Alqurni, 2024).

a. *Ekstrasi Sliding-window features*

Tahap ini memerlukan waktu 2 detik untuk mengekstrak fitur dari paket jaringan, sejalan dengan pendekatan khusus pada dataset *NSL-KDD* dan UNSW-NB15 dalam literatur IDS terkini. Rumus yang digunakan.

$$serror\_rate = \frac{Jumlah\ Koneksi\ dengan\ error\ SYN}{total\ koneksi\ ke\ host} \quad (1)$$

$$rerror_{rate} = \frac{Jumlah\ Koneksi\ dengan\ error\ RST}{total\ koneksi\ ke\ host} \quad (2)$$

$$srv_{error_{rate}} = \frac{Jumlah\ error\ SYN\ pada\ service}{jumlah\ koneksi\ ke\ service} \quad (3)$$

b. *Preprocessing Data*

Objek *preprocessor.transform()* melakukan normalisasi dan encoding fitur, seperti skala dan encoding kategori. Ini konsisten dengan metode *preprocessing* pada penelitian *CNN* untuk IDS, Jika menggunakan *StandardScaler*. Sedangkan untuk *one-hot encoding*, variabel kategorik seperti *protocol_type* dan service diubah ke format numerik biner.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4)$$

Ilustrasi kasus IDS, Fitur duration (lama koneksi) dalam dataset bisa bernilai 0 sampai 58329 detik. Jika tanpa normalisasi, *CNN* akan sulit belajar karena nilai terlalu besar.
Setelah normalisasi (0–1):

- *duration* = 0 → 0
- *duration* = 58329 → 1
- *duration* = 100 → $\frac{100-0}{58329-0}$ = 0.0017

One-Hot Encoding (untuk kategori) Untuk variabel kategorikal seperti protocol_type (tcp, udp, icmp), diubah ke representasi biner.
tcp=[1,0,0],udp=[0,1,0],icmp=[0,0,1]

Ilustrasi kasus IDS

- Fitur *protocol_type* memiliki nilai *tcp, udp, icmp.*
- *CNN/SVM* tidak bisa langsung membaca string → diubah ke vektor biner.
- Jika data tcp, maka input model = [1, 0, 0].

Log Transformation (untuk distribusi *skewed*) Jika ada fitur yang distribusinya tidak normal (misalnya jumlah *byte* sangat besar pada serangan *DDoS*), maka dilakukan *log-scaling*:

$$X' = log(1 + X)$$

Ilustrasi kasus IDS, Fitur *src_bytes* bernilai ekstrem (misalnya 0 sampai jutaan). Dengan log transformasi:

- src_bytes = 0 → 0
- src_bytes = 1000 → log(1001) ≈ 6.9
- src_bytes = 1.000.000 → log(1.000.001) = 13.8

c. Ekstraksi Fitur dengan *CNN*

CNN diposisikan sebagai feature extractor, bukan classifier akhir. Ia melakukan operasi konvolusi terhadap fitur bentuk 1D, dengan ϕ sebagai fungsi aktivasi (contoh: *ReLU*). Arsitektur ini umum digunakan untuk deteksi intrusi berbasis *CNN-LSTM/CNN-BiLSTM.*

$$fk(x) = \phi((x * wk) + bk)$$

d. Klasifikasi Anomali menggunakan *SVM*

Output CNN kemudian dilabeli sebagai normal atau anomali oleh *SVM*. Fungsi keputusan *SVM* dituliskan sebagai Kelas ditentukan (5),(6),(7).

f(x)=+1f(x) = +1f(x)=+1 jika anomali (5)

f(x)=−1f(x) = -1f(x)=−1 jika normal (6)

$$f(x) = sign(wTx + b) \quad (7)$$

3. Hasil dan Pembahasan

Penelitian ini menghasilkan kebaruan deteksi anomali jaringan komputer yang membedakan dengan hasil berbasis rules snort saja karena statis tidak bisa mengenali pola keberlanjutan jenis serangan di masa yang akan datang maka ditemukan lah sebuah model *CNN-SVM* yang dapat mendeteksi pola serangan dan tingkah laku tidak wajar atau anomali di suatu lalu lintas jaringan komputer menggunakan bantuan *tools scapy* yang tentunya sudah dilatih terlebih dahulu dengan dataset *NSL-KDD* sehingga dapat belajar dan bereksplorasi mengenali pola anomali di suatu jaringan komputer tanpa harus mengandalkan *rules statis*. proses ini meliputi tahapan dengan melakukan simulasi serangan di dalam jaringan komputer agar komputer target dapat mendeteksi terdapat lalu lintas yang anomali dengan menyiapkan 2 unit komputer penyerang dengan sistem operasi kali linux kemudain terkoneksi jaringan komputer target dengan sistem operasi ubuntu yang sudah di konfigurasi scapy dan dilatih untuk mengenali dan mengidentifikasi pola serangan, berikut data file python dan dataset training *NSL-KDD* melalui github https://github.com/arflqmanhdyni/Deteksi-Anomali-*SVM-CNN*/tree/main.

3.1. Simulasi Serangan

Jaringan komputer yang dikonfigurasi meliputi *Ip address* dengan mode *(DHCP) Dynamic Host Configuration Protocol,* kemudian konfigurasi *firewall inbound rules* dengan mengizinkan *ip address DHCP* yang diterima di komputer penyerang 1 & 2 serta komputer target agar proses serangan simulasi tidak terblokir *firewall*. Setelah itu dilakukan proses penyerangan. Berikut prompt untuk melakukan simulasi penyerangan ke komputer target melalui mode promiscuous di perangkat switch dan perangkat komputer penyerang. mensimulasikan serangan *SYN Flood* terhadap layanan *SSH (Secure Shell),* digunakan perintah Hydra Gambar 3 (Sinha, Bera, & Satpathy, 2025).

```
hping3 -S -p 22 --flood 192.168.1.124
```

Gambar 3. *Prompt* Serangan *SYN Flood SSH*

Jaringan komputer yang dikonfigurasi meliputi *Ip address* dengan mode *(DHCP) Dynamic Host Configuration Protocol,* kemudian konfigurasi *firewall inbound rules* dengan mengizinkan *ip address DHCP* yang diterima di komputer penyerang 1 & 2 serta komputer target agar proses serangan simulasi tidak terblokir *firewall*. Setelah itu dilakukan proses penyerangan. Berikut prompt untuk melakukan simulasi penyerangan ke komputer target melalui mode *promiscuous* di perangkat switch dan perangkat komputer penyerang. mensimulasikan serangan *SYN Flood*

terhadap layanan *SSH (Secure Shell),* digunakan perintah Hydra Gambar 4.

```
hping3 -S -p 22 --flood 192.168.1.124
```

Gambar 4. *Prompt* Serangan *SYN Flood SSH*

Simulasi serangan terhadap layanan SSH dilakukan menggunakan perintah hping3 -S -p 22 --flood 192.168.1.124, yang bertujuan menghasilkan lalu lintas anomali berbentuk serangan *SYN Flood* terhadap port 22, yaitu port default layanan Secure Shell (*SSH*). Dalam konteks keamanan jaringan, *SYN Flood* merupakan jenis serangan Denial of Service (DoS) yang mengeksploitasi kelemahan pada proses inisiasi koneksi *TCP*. Perintah ini menggunakan opsi -S untuk mengirimkan paket *TCP* dengan flag *SYN* aktif, menandakan permintaan awal untuk membentuk koneksi. Opsi -p 22 menunjukkan bahwa sasaran serangan adalah port layanan SSH, sementara --flood digunakan untuk mengirimkan paket secara terus-menerus dan cepat, tanpa menunggu respons dari *server*. Tujuan dari serangan ini adalah membanjiri *server* dengan koneksi palsu sehingga sumber daya *server* terkunci pada status koneksi setengah terbuka (half-open), yang pada akhirnya dapat menyebabkan layanan SSH menjadi tidak responsif atau gagal melayani koneksi sah dari pengguna yang valid.

Berikutnya melakukan serangan terhadap layanan *FTP* disimulasikan menggunakan perintah hping3 -S -p 21 --flood 192.168.1.124. Tujuan dari perintah ini adalah menghasilkan lalu lintas jaringan yang tidak normal, menyerupai serangan tipe *SYN Flood*. Hping3 dipilih karena kemampuannya dalam mengirimkan paket *TCP*/IP dengan parameter yang dapat disesuaikan, sehingga cocok digunakan untuk pengujian skenario serangan. Flag -S mengaktifkan pengiriman paket *SYN*, sedangkan opsi -p 21 menargetkan port standar layanan *FTP*. Perintah --flood digunakan untuk membanjiri target dengan paket secara terus-menerus tanpa jeda, yang dapat membebani sistem dan mengganggu layanan. Serangan *SYN Flood* umumnya mengeksploitasi kelemahan pada proses awal koneksi *TCP*, di mana penyerang mengirim banyak permintaan koneksi namun tidak pernah menyelesaikan prosesnya, sehingga koneksi tertahan dalam status setengah terbuka. Lalu lintas hasil simulasi ini digunakan sebagai data anomali untuk melatih dan menguji model deteksi berbasis *CNN* dan *SVM*. Melalui pendekatan ini, sistem diuji kemampuannya dalam mengenali aktivitas jaringan yang mencurigakan. Simulasi ini penting karena meniru pola serangan yang realistis dan sering terjadi dalam praktik keamanan jaringan. Serangan dilakukan melalui *prompt* Gambar 5.

```
hping3 -S -p 21 --flood 192.168.1.124
```

Gambar 5. *Prompt* Serangan *SYN Flood FTP*

Perintah di atas Dengan memasukkan pola lalu lintas yang menyerupai serangan nyata ke dalam dataset pelatihan, model pembelajaran mesin dapat mengenali karakteristik lalu lintas berbahaya secara lebih akurat. Dalam kasus serangan *SYN Flood* terhadap *port FTP*, sistem diharapkan mampu membedakan antara koneksi *TCP* yang sah dan koneksi yang bersifat destruktif atau mencurigakan. Ini menjadi dasar penting dalam pengembangan sistem deteksi yang tidak hanya mengandalkan metode signature-based, tetapi juga mampu mengidentifikasi pola perilaku abnormal secara *real-time*. Selain itu, pengujian dengan data yang dihasilkan dari simulasi nyata memberikan validitas yang lebih kuat terhadap performa sistem, sekaligus memungkinkan penyesuaian parameter model agar lebih sesuai dengan kondisi lingkungan jaringan sebenarnya.

Simulasi selanjutnya menyerang layanan *HTTP* dilakukan simulasi serangan jenis *SYN Flood* yang menargetkan layanan *HTTP* pada *server* beralamat IP *192.168.1.124*. Serangan ini dimaksudkan untuk merepresentasikan trafik anomali yang umum dijumpai dalam insiden Denial of Service (DoS). *SYN Flood* merupakan serangan DoS klasik yang bertujuan menghabiskan sumber daya sistem dengan mengirimkan sejumlah besar permintaan koneksi (*SYN*) ke *server* target tanpa pernah menyelesaikan proses *three-way handshake*. Akibatnya, *server* target akan menyimpan status koneksi sementara (*half-open connections*) dalam jumlah besar hingga kehabisan buffer atau memory, sehingga tidak dapat melayani permintaan yang sah (*legitimate*). Perintah yang digunakan untuk menghasilkan serangan adalah sebagai berikut Gambar 6.

```
hping3 -S -p 80 --flood 192.168.1.124
```

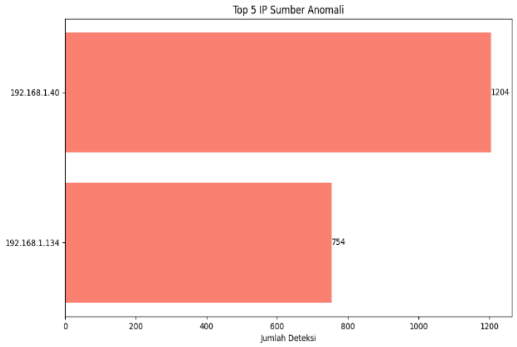
Gambar 6. *Prompt* Serangan *SYN Flood HTTP*

Simulasi dilakukan dalam lingkungan terkontrol untuk menghasilkan lalu lintas jaringan yang mengandung pola anomali spesifik, yang selanjutnya digunakan untuk mendeteksi anomali jaringan *(Intrusion Detection System/IDS)* berbasis deep learning *CNN* dan *SVM*. Ekstraksi fitur lalu lintas abnormal, seperti jumlah koneksi *SYN* tidak lengkap, rasio *TCP handshake* tidak valid, dan frekuensi koneksi dalam rentang waktu tertentu. Dampak Terukur pada Target, Kenaikan tajam pada koneksi *TCP* yang tidak lengkap. Penurunan kemampuan sistem dalam menerima koneksi sah. Potensi *crash* atau *overloading service HTTP* jika tidak dibatasi dengan *firewall* atau *rate-limiting.* Berikut hasil dari deteksi yang dihasilkan di setiap 5 menit secara periode mengumpulkan laporan deteksi anomali jaringian komputer.

3.2. Hasil Deteksi Anomali Jaringan Komputer

Hasil Laporan Aktivitas Anomali Jaringan Periodik yang mencakup analisis lalu lintas jaringan selama periode lima menit terakhir, tepatnya dimulai dari 12.35.00 sampai pada pukul 12:30:32. Menghasilkan ringkasan Aktivitas dalam bentuk matriks teks dan visualisasi grafik anomali. Pada bagian ringkasan aktivitas, disebutkan bahwa total paket jaringan yang dianalisis adalah 2323, dan dari jumlah tersebut, sebanyak 1958 paket teridentifikasi sebagai anomali berdasarkan sistem deteksi yang digunakan berdasarkan model *hybrid* seperti *CNN* dan *SVM*.

Bagian kedua dari laporan menampilkan visualisasi bar chart horizontal yang menggambarkan Top 5 IP sumber anomali, meskipun pada grafik ini hanya dua IP yang ditampilkan. IP 192.168.1.40 menjadi sumber anomali tertinggi dengan 1204 deteksi, diikuti oleh 192.168.1.134 dengan 754 deteksi. Warna merah muda yang digunakan dalam grafik menunjukkan intensitas aktivitas mencurigakan yang signifikan dari masing-masing IP, memberikan indikasi kuat adanya potensi serangan siber atau perilaku jaringan tidak normal yang berasal dari alamat IP tersebut. Visualisasi ini sangat berguna bagi administrator jaringan untuk cepat mengidentifikasi sumber gangguan dan mengambil langkah mitigasi yang tepat seperti yang dihasilkan pada Gambar 7.

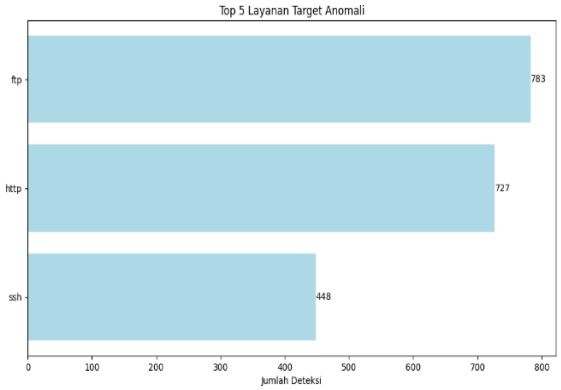


Gambar 7. Sumber IP Anomali

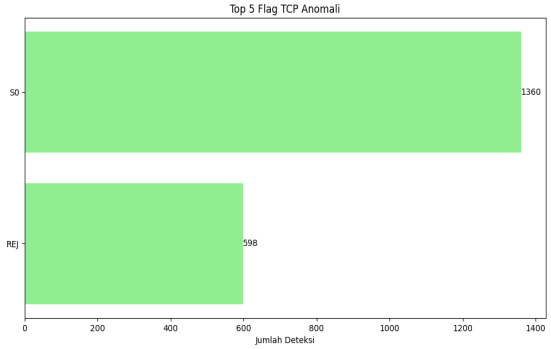
Selama periode pemantauan terakhir, layanan jaringan yang paling sering menjadi sasaran aktivitas anomali adalah *FTP, HTTP,* dan *SSH*. Layanan *FTP* menempati posisi teratas dengan total 783 deteksi, diikuti oleh *HTTP* sebanyak 727 deteksi dan *SSH* sebanyak 448 deteksi. Hal ini menunjukkan bahwa pelaku serangan cenderung menargetkan layanan-layanan penting yang umumnya terbuka pada sistem jaringan, seperti *FTP* untuk transfer file, *HTTP* untuk akses web, serta *SSH* yang biasa digunakan untuk remote login. Fenomena ini mengindikasikan adanya kemungkinan percobaan eksploitasi, brute-force login, atau scanning port yang secara terus-menerus membanjiri layanan-layanan tersebut, terutama dalam bentuk trafik tidak sah.

Selain itu, analisis terhadap flag *TCP* yang terdeteksi menunjukkan bahwa flag S0 mendominasi aktivitas anomali dengan total 1360 deteksi. Flag ini umumnya muncul saat koneksi *TCP* diinisiasi tetapi tidak mendapatkan respon dari host tujuan, yang menjadi indikasi kuat dari aktivitas scanning atau percobaan koneksi ke port yang tidak aktif. Flag *REJ* juga muncul cukup sering dengan 598 deteksi, menandakan adanya koneksi yang secara eksplisit ditolak oleh host target, kemungkinan besar akibat percobaan koneksi ke port yang tertutup atau layanan yang dibatasi. Pola ini memperkuat dugaan bahwa terjadi aktivitas pemindaian dan probing secara intensif terhadap sistem target. Dengan mengetahui karakteristik anomali melalui analisis layanan dan flag *TCP*, sistem deteksi intrusi dapat lebih responsif dalam mengidentifikasi ancaman serta memberikan dasar yang kuat untuk mitigasi secara proaktif, hasil visualisasi grafik layanan target anomali Gambar 8.

ISSN: 2541-1004
e-ISSN: 2622-4615
https://doi.org/10.32493/informatika.v10i2.52163



Gambar 8. Layanan Target Anomali



Gambar 9. *Flag TCP* Anomali

Tabel 2. Log Anomali Terdeteksi (1958) setiap periode 5 menit

Waktu	IP Sumber	IP Tujuan	Layanan	Flag
12:25:32	192.168.1.40	192.168.1.124	http	S0
12:25:32	192.168.1.40	192.168.1.124	ftp	S0
12:25:32	192.168.1.40	192.168.1.124	ssh	S0
12:25:32	192.168.1.40	192.168.1.124	http	S0
12:25:33	192.168.1.134	192.168.1.124	ftp	REJ
12:25:33	192.168.1.134	192.168.1.124	ssh	S0
12:25:33	192.168.1.134	192.168.1.124	ftp	S0
12:25:33	192.168.1.134	192.168.1.124	Ssh	REJ
.....	.....	.....	....	....
12:30:35	192.168.1.40	192.168.1.124	ftp	S0

Tabel 2 menunjukkan rekaman log anomali yang terdeteksi selama periode lima menit, dengan total 1958 entri anomali. Data tersebut mencakup waktu kejadian, alamat IP sumber, IP tujuan, layanan yang ditargetkan, dan jenis *TCP* flag yang terdeteksi. Dari log tersebut terlihat bahwa terdapat sejumlah besar paket dengan flag S0 dan REJ, yang merupakan indikator kuat terhadap aktivitas jaringan yang mencurigakan, seperti port scanning atau serangan *brute force,* pada waktu 12:25:32, terlihat bahwa host dengan alamat 192.168.1.40 mengirimkan koneksi ke IP 192.168.1.124 secara simultan ke beberapa layanan, yakni *HTTP, FTP,* dan *SSH*, semuanya dengan flag S0. Ini menunjukkan bahwa host tersebut melakukan SYN scan suatu teknik stealth scanning yang digunakan untuk mengidentifikasi port terbuka tanpa menyelesaikan three-way handshake TCP. Paket S0 (SYN sent, no reply) menandakan bahwa permintaan koneksi tidak mendapat respon dari target, yang dapat disebabkan oleh port tertutup, firewall, atau target yang memblokir koneksi.

Selanjutnya, pada waktu 12:25:33, alamat 192.168.1.134 terlihat mencoba mengakses FTP dan SSH dari target yang sama, dengan flag REJ dan S0. Flag REJ (reject) menunjukkan bahwa target secara aktif menolak koneksi yang masuk ke port tertentu, umumnya karena port tertutup atau permintaan tidak sah. Kombinasi penggunaan flag S0 dan REJ dalam log ini memperkuat dugaan bahwa telah terjadi aktivitas pemindaian atau serangan berbasis percobaan masuk ke berbagai layanan, yang kemungkinan berasal dari sistem yang dikompromi atau digunakan sebagai perantara serangan. Log ini juga menunjukkan bahwa aktivitas mencurigakan berlangsung terus-menerus selama rentang waktu yang cukup lama, bahkan hingga pukul 12:30:35, di mana koneksi FTP dengan flag S0 masih tercatat dari IP 192.168.1.40 ke target yang sama.

Berdasarkan pola ini, dapat disimpulkan bahwa sistem monitoring telah berhasil mendeteksi serangkaian anomali yang signifikan dalam lalu lintas jaringan, dan kemungkinan besar merupakan bagian dari aktivitas scanning atau eksploitasi layanan tertentu. Hal ini menunjukkan urgensi penerapan sistem deteksi anomali *real-time* untuk mitigasi potensi serangan siber lebih lanjut.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan dan membuktikan efektivitas sebuah model hibrida yang menggabungkan *Convolutional Neural*

Network (CNN) dan *Support Vector Machine (SVM)* untuk sistem deteksi anomali jaringan secara *real-time*. Inovasi utama dari penelitian ini terletak pada penciptaan model deteksi pola serangan yang cerdas dan mandiri, yang tidak lagi bergantung pada rules statis yang kaku. Dengan melatih model menggunakan dataset *NSL-KDD* terlebih dahulu, sistem ini mampu belajar dan secara otomatis mengenali perilaku lalu lintas jaringan yang mencurigakan.Dalam skenario pengujian yang melibatkan simulasi serangan dari dua komputer penyerang berbasis Kali Linux terhadap satu komputer target berbasis Ubuntu, sistem yang memanfaatkan *scapy* untuk menangkap lalu lintas jaringan menunjukkan kinerja yang sangat baik. Sistem ini terbukti mampu mendeteksi berbagai jenis serangan yang disimulasikan, seperti serangan SYN Flood yang menargetkan layanan-layanan vital seperti *SSH*, *FTP*, dan *HTTP*. Laporan deteksi yang dihasilkan secara periodik setiap lima menit menyajikan analisis mendalam, termasuk visualisasi sumber anomali berdasarkan alamat IP , layanan yang paling sering menjadi sasaran , dan analisis flag *TCP* (misalnya *S0* dan *REJ*) yang menjadi indikator kuat adanya aktivitas pemindaian atau serangan. Dengan demikian, penelitian ini menegaskan bahwa pendekatan hibrida *CNN-SVM* merupakan solusi yang kuat dan efektif untuk memperkuat keamanan jaringan, memungkinkan deteksi ancaman secara proaktif dan respons mitigasi yang lebih cepat.

Referensi

Al Ghamdi, M. A. (2023). A Fine-Grained System Driven of Attacks over Several New Representation Techniques Using Machine Learning. *IEEE Access*, *11*(July), 96615–96625. https://doi.org/10.1109/ACCESS.2023.3307018

Alrayes, F. S., Zakariah, M., Amin, S. U., Khan, Z. I., & Alqurni, J. S. (2024). CNN Channel Attention Intrusion Detection System Using *NSL-KDD* Dataset. *Computers, Materials and Continua*, *79*(3), 4319–4347. https://doi.org/10.32604/cmc.2024.050586

Altunay, H. C., & Albayrak, Z. (2023). A *hybrid* CNN + LSTMbased intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, *38*, 101322. https://doi.org/10.1016/j.jestch.2022.101322

Berhane, T., Melese, T., Walelign, A., & Mohammed, A. (2023). A *Hybrid Convolutional Neural Network* and *Support Vector Machine*-Based Credit Card Fraud Detection Model. *Mathematical Problems in Engineering*, *2023*(1). https://doi.org/10.1155/2023/8134627

Choi, Y., Choi, H., & Seo, S. C. (2023). AVX512Crypto: Parallel Implementations of Korean Block Ciphers Using AVX-512. *IEEE Access*, *11*(May), 55094–55106. https://doi.org/10.1109/ACCESS.2023.3278993

Firdaus, D., Fahira, F., & Rianti, R. (2023). Deteksi Anomali Dan Serangan Low Rate Ddos Dalam Lalu Lintas Jaringan Menggunakan Naive Bayes. *Naratif : Jurnal Nasional Riset, Aplikasi Dan Teknik Informatika*, *5*(2), 140–148. https://doi.org/10.53580/naratif.v5i2.208

Irfan, A., Nusri, A. Z., Rachmat, Z., & Wulandari, S. (2024). Analisis Keamanan Jaringan Nirkabel Menggunakan Wireless Intrusion Detection System (WIDS). *Jurnal Ilmiah Sistem Informasi Dan Teknik Informatika (JISTI)*, *7*(1), 110–119. https://doi.org/10.57093/jisti.v7i1.195

Jagtap, S. S., Shankar Sriram, V. S., & Subramaniyaswamy, V. (2021). A hypergraph based Kohonen map for detecting intrusions over cyber–physical systems traffic. *Future Generation Computer Systems*, *119*, 84–109. https://doi.org/10.1016/j.future.2021.02.001

Januantoro, A., Scanning, P., Scanning, P., & Force, B. (2025). Deteksi Serangan Jaringan Komputer Berbasis Snort Dengan, *8*(1), 100–105.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of *hybrid* intrusion detection system for detecting internet of things attacks. *Electronics (Switzerland)*, *8*(11). https://doi.org/10.3390/electronics8111210

Lateef, A. A. A., Al-Janabi, S. T. F., & Al-Khateeb, B. (2019). Survey on intrusion detection systems based on deep learning. *Periodicals of Engineering and Natural Sciences*, *7*(3), 1074–1095. https://doi.org/10.21533/pen.v7i3.635

Liu, H., & Wang, H. (2023). *Real-time* Anomaly Detection of Network Traffic Based on CNN. *Symmetry*, *15*(6). https://doi.org/10.3390/sym15061205

Ma, Q., Sun, C., Cui, B., & Jin, X. (2021). A novel model for anomaly detection in network traffic based on kernel *Support Vector Machine*. *Computers and Security*, *104*, 102215. https://doi.org/10.1016/j.cose.2021.102215

Ngurah, I. G., Dika, M., Agung, I. G., & Arya, G. (2025). Klasifikasi Serangan Distributed Denial of Service (DDoS) Mempergunakan *Support Vector Machine* dengan Correlation- Based Feature Selection, *13*(3), 543–558.

Ogah, M. D., Essien, J., Ogharandukun, M., & Abdullahi, M. (2024). Machine Learning Models for Heterogenous Network Security Anomaly Detection. *Journal of Computer and Communications*, *12*(06), 38–58. https://doi.org/10.4236/jcc.2024.126004

Ozkan-Okay, M., Samet, R., Aslan, O., & Gupta, D.

(2021). A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access*, *9*, 157727–157760. https://doi.org/10.1109/ACCESS.2021.3129336

S, D. S., Wahyuddin, W., Kautsar, A., & Setyawan, A. (2025). Intrusion Detection System Menggunakan Snort dan Telegram Sebagai Media Notifikasi. *SisInfo*, *7*(1), 40–49. https://doi.org/10.37278/sisinfo.v7i1.1068

Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: a *hybrid* machine and deep learning approach. *Journal of Cloud Computing*, *13*(1). https://doi.org/10.1186/s13677-024-00685-x

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. *IEEE Access*, *8*(November), 222310–222354. https://doi.org/10.1109/ACCESS.2020.3041951

Sinha, M., Bera, P., & Satpathy, M. (2025). SYN-Monitor: An Energy Efficient Defense System against TCP-SYN Flooding Attacks in SDN. *ICDCN 2025 - Proceedings of the 26th International Conference on Distributed Computing and Networking*, 346–351. https://doi.org/10.1145/3700838.3703695