

Analisis Kinerja dan Interpretabilitas *Multilayer Perceptron* untuk Deteksi Serangan DDoS pada *Software-Defined Networking*

David Suharjanto^{1*}, Muhammad Taufiq Nuruzzaman¹

¹Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta, Indonesia

*Email : davidsuharjanto7@gmail.com

Info Artikel

Kata Kunci :

DDoS, SDN, *Multilayer Perceptron*, Analisis Performa, Interpretabilitas Model

Keywords :

DDoS, SDN, *Multilayer Perceptron*, Performance Analysis, Model Interpretability

Tanggal Artikel

Dikirim : 4 April 2026

Direvisi : 12 Juni 2026

Diterima : 21 Oktober 2026

Abstrak

Penelitian ini menganalisis kinerja dan interpretabilitas model *Multilayer Perceptron* (MLP) untuk mendeteksi serangan *Distributed Denial of Service* (DDoS) pada arsitektur *Software-Defined Networking* (SDN). Analisis kinerja dilakukan dengan mengevaluasi berbagai kedalaman lapisan dan jumlah neuron pada arsitektur MLP menggunakan dataset publik DDoS attack SDN, sementara metode SHapley Additive exPlanations (SHAP) digunakan untuk mengidentifikasi fitur dominan yang memengaruhi keputusan model. Hasil eksperimen menunjukkan bahwa arsitektur MLP tiga lapisan dengan konfigurasi *funneling* (256–128–32) memberikan kinerja terbaik, mencapai akurasi 99,84% dan F1-score 99,80%, serta menunjukkan bahwa efektivitas optimasi jumlah layer dan neuron dalam meningkatkan performa deteksi dibandingkan penelitian terdahulu. Analisis SHAP mengungkapkan bahwa *pktpcount*, *pktrate*, dan *pktpflow* merupakan fitur paling dominan dalam mendeteksi serangan DDoS. Penelitian ini mengombinasikan analisis arsitektur dan interpretabilitas berbasis SHAP untuk meningkatkan transparansi keputusan model pada keamanan jaringan SDN.

Abstract

*This study investigates the performance and interpretability of Multilayer Perceptron (MLP) models for Distributed Denial of Service (DDoS) attack detection in Software-Defined Networking (SDN) environments. Performance analysis was conducted by comprehensively evaluating variations in layer depth and neuron count across MLP architectures using a public DDoS attack SDN dataset, while SHapley Additive exPlanations (SHAP) were applied to identify dominant features influencing the model's decision-making process. The results show that a three-layer MLP architecture with a funneling configuration (256–128–32) achieves the best performance, with an accuracy of 99.84% and an F1-score of 99.80%, demonstrating the effectiveness of optimizing the number of layers and neurons in improving detection performance compared to previous studies. SHAP analysis further reveals that *pktpcount*, *pktrate*, and *pktpflow* are the most dominant features in DDoS attack detection. This study combines architectural analysis and SHAP-based interpretability to enhance the transparency of model decision-making in SDN network security.*

1. PENDAHULUAN

Serangan *Distributed Denial of Service* (DDoS) merupakan salah satu ancaman siber terbesar pada infrastruktur jaringan modern, tak terkecuali pada arsitektur *Software-Defined Networking* (SDN) [1], [2]. Arsitektur SDN yang memisahkan *control plane* dan *data plane* memberikan fleksibilitas dalam pengelolaan jaringan, namun secara bersamaan meningkatkan risiko eksploitasi karena kontrol terpusat dapat menjadi titik lemah yang rentan diserang [3], [4]. Serangan DDoS pada arsitektur SDN dapat membanjiri *controller* maupun *switch* dengan lalu lintas yang sangat tinggi, sehingga mengakibatkan konsumsi sumber daya yang berlebihan, kemacetan aliran data, hingga potensi lumpuhnya fungsi layanan jaringan secara keseluruhan [5]. Oleh karena itu, dibutuhkan metode deteksi dini serangan DDoS pada arsitektur SDN guna memastikan stabilitas dan keberlangsungan layanan jaringan.

Seiring dengan berkembangnya teknologi di bidang kecerdasan buatan, berbagai pendekatan machine learning telah diterapkan untuk meningkatkan efektivitas deteksi serangan DDoS pada lingkungan SDN. Ye et al. [6] menerapkan algoritma Support Vector Machine (SVM) pada dataset simulasi SDN dan mencapai akurasi 95,24%. Deepa et al. [7] mengusulkan pendekatan hybrid yang menggabungkan SVM dan Self Organizing Map (SOM), menghasilkan akurasi sebesar 98,12% pada dataset simulasi berbasis Mininet. Myint et al. [8] mengembangkan Advanced SVM (ASVM) yang diuji menggunakan dataset lalu lintas sintesis SDN dan menghasilkan akurasi hingga 97%. Selain itu, Dong & Sarem [9] mengembangkan algoritma Improved K-Nearest Neighbors (KNN) pada dataset simulasi berbasis Mininet, yang berhasil mencapai tingkat deteksi (True Positive Rate) sebesar 99,4%.

Sementara itu, sejumlah penelitian telah mengevaluasi berbagai metode machine learning untuk deteksi serangan DDoS menggunakan dataset benchmark publik. Perbandingan beberapa penelitian terdahulu ditunjukkan pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terdahulu

Peneliti	Dataset	Algoritma	Akurasi
Sudar et al. [10]	KD99	SVM	85,00%
		Decision Tree	78,00%
Ahuja et al. [11]	DDoS Attack SDN	SVC-RF	98,80%
		ANN	98,20%
Rahman & Sutabri [12]	DDoS Attack SDN	MLP	99,05%

Di antara berbagai algoritma yang ada, MLP memiliki potensi yang signifikan untuk dikembangkan lebih lanjut dalam keamanan siber di lingkungan SDN [13]. MLP memiliki kemampuan untuk memodelkan hubungan non-linear yang kompleks antar fitur lalu lintas jaringan, yang seringkali sulit ditangkap oleh model linier sederhana [14]. Fleksibilitas MLP dalam mempelajari representasi fitur melalui *hidden layer* menjadikannya efektif dalam mempelajari representasi fitur dari karakteristik lalu lintas SDN yang berdimensi tinggi [15].

Meskipun studi terdahulu telah menunjukkan performa yang baik, terdapat keterbatasan yang menyisakan ruang untuk pengembangan lebih lanjut. Penelitian sebelumnya yang menggunakan MLP, seperti yang dilakukan oleh Rahman & Sutabri [12], cenderung menerapkan satu konfigurasi arsitektur standar tanpa mengevaluasi dampak variasi jumlah *hidden layer* secara mendalam. Padahal struktur jaringan saraf sangat berperan penting dalam kapasitas model untuk mempelajari pola data yang kompleks [16]. Selain itu, sebagian besar penelitian masih berfokus pada metrik performa seperti akurasi maupun F1-score tanpa memberikan analisis mengenai fitur apa yang paling berpengaruh dalam memicu deteksi serangan, padahal interpretabilitas model sangat krusial dalam keamanan jaringan agar administrator dapat memahami karakteristik serangan DDoS dan mengambil keputusan mitigasi secara tepat [17], [18].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengoptimalkan kinerja deteksi DDoS pada arsitektur SDN melalui evaluasi komprehensif terhadap variasi arsitektur MLP. Penelitian ini akan membandingkan berbagai konfigurasi *hidden layer* untuk menentukan arsitektur terbaik yang mampu menyeimbangkan akurasi tinggi dan presisi deteksi dalam dataset publik serangan DDoS pada SDN. Selain itu, penelitian ini juga bertujuan untuk menganalisis interpretabilitas model terbaik menggunakan pendekatan SHapley Additive exPlanations (SHAP), guna mengidentifikasi fitur-fitur jaringan yang paling berpengaruh dalam mendeteksi serangan DDoS.

2. METODE PENELITIAN

Penelitian ini mengikuti alur kerja sistematis sebagaimana diilustrasikan pada Gambar 1. Tahapan dimulai dengan pemuatan dataset dan pra-pemrosesan data untuk mempersiapkan input model. Langkah selanjutnya adalah pembagian dataset menjadi data latih dan data uji, yang dilanjutkan dengan pelatihan model melalui variasi arsitektur MLP. Kinerja setiap model kemudian diukur pada tahap evaluasi model menggunakan berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Model terbaik dilakukan analisis SHAP untuk mengevaluasi kontribusi setiap fitur serta meningkatkan interpretabilitas prediksi. Terakhir, hasil yang diperoleh dianalisis secara komprehensif sebagai dasar pembahasan eksperimen.



Gambar 1. Diagram Alir Penelitian

2.1 Dataset

Penelitian ini menggunakan dataset publik bernama DDoS attack SDN, yang dikembangkan oleh Ahuja et al. [19]. Dataset ini terdiri dari 104.345 baris dengan 22 fitur numerik serta label biner yang merepresentasikan kategori *benign* (0) dan *malicious* (1).

2.2 Pra-pemrosesan

Tahapan pra-pemrosesan pertama adalah menangani atribut kategorikal yang terdapat pada kolom *src*, *dst*, dan *protocol* melalui proses *label encoding*. Tahap ini mengubah nilai kategorikal menjadi representasi numerik sehingga dapat diproses oleh model MLP yang hanya menerima input berupa angka. Selanjutnya, dilakukan penanganan *missing values* pada kolom *rx_kbps* dan *tot_kbps*, yang masing-masing memiliki 506 entri kosong, menggunakan metode imputasi median. Strategi ini dipilih karena lebih robust terhadap distribusi data yang miring [20]. Setelah itu, seluruh fitur numerik dinormalisasi menggunakan *StandardScaler* untuk menyamakan skala variabel.

Tahap terakhir adalah pembagian dataset menjadi 80% train dan 20% test, dengan nilai random state sebesar 42. Proses split dataset menggunakan metode stratifikasi agar distribusi kelas tetap seimbang pada kedua subset. Detail pembagian dataset dapat diamati pada Tabel 2. Keseluruhan tahapan pra-pemrosesan ini memastikan bahwa data berada dalam kondisi optimal sebelum masuk ke tahap pelatihan model dan evaluasi performa.

Tabel 2. Detail Pembagian Dataset

Kategori	Train	Test
<i>benign</i>	50.489	12.712
<i>malicious</i>	32.627	8.157
Total	83.476	20.869

2.3 Arsitektur Model

Penelitian ini menggunakan MLP Classifier sebagai model utama untuk tugas klasifikasi serangan DDoS pada data tabular. MLP dipilih karena kemampuannya dalam mempelajari relasi non-linear antar fitur serta fleksibilitas dalam pengaturan jumlah layer dan neuron [21]. Untuk menganalisis pengaruh kompleksitas arsitektur terhadap kinerja deteksi, penelitian ini mengevaluasi 14 konfigurasi MLP yang terdiri atas satu hingga tiga *hidden layer*. Pembatasan hingga tiga *hidden layer* dilakukan karena arsitektur dengan kedalaman moderat umumnya telah memadai untuk memodelkan data tabular, sedangkan penambahan layer yang lebih dalam dapat meningkatkan kompleksitas model dan risiko overfitting [22]. Jumlah neuron dipilih secara bertingkat menggunakan konfigurasi 32, 64, 128, dan 256 untuk merepresentasikan peningkatan kapasitas model secara sistematis. Rentang tersebut juga sejalan dengan penelitian deteksi DDoS berbasis MLP yang umumnya menggunakan *hidden layer* hingga 256 neuron [23]. Daftar lengkap variasi arsitektur ditunjukkan pada Tabel 3.

Tabel 3. Variasi Arsitektur MLP

Kategori Arsitektur	Konfigurasi Neuron
Satu Lapisan	32, 64, 128, 256
Dua Lapisan	64–32, 128–32, 128–64, 256–32, 256–64, 256–128
Tiga Lapisan	128–64–32, 256–128–32, 256–128–64, 256–64–32

2.4 Konfigurasi Pelatihan

Seluruh konfigurasi pelatihan dilakukan secara konsisten untuk memastikan perbandingan yang adil antar arsitektur MLPC. Model dilatih menggunakan optimizer Adam dan fungsi aktivasi ReLU dengan 30 iterasi pelatihan. Selain itu, proses pelatihan dijalankan menggunakan random state sebesar 42 untuk memastikan reproduktabilitas.

2.5 Metrik Evaluasi

Kinerja setiap model dievaluasi menggunakan empat metrik utama meliputi akurasi, presisi, recall, dan F1-score. Seluruh metrik dihitung dengan pendekatan *macro-average*, di mana nilai masing-masing kelas dihitung nilai rata-ratanya tanpa memperhitungkan proporsi distribusi datanya. Untuk analisis kesalahan yang lebih mendalam, penelitian ini juga memanfaatkan confusion matrix.

Secara umum, confusion matrix terdiri dari empat komponen utama:

- *False Positive* (FP): data negatif yang keliru diprediksi sebagai positif.
- *False Negative* (FN): data positif yang keliru diprediksi sebagai negatif.
- *True Positive* (TP): data positif yang berhasil diprediksi sebagai positif.
- *True Negative* (TN): data negatif yang berhasil diprediksi sebagai negatif.

Perhitungan masing-masing metrik dilakukan berdasarkan persamaan berikut:

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

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

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

$$\text{F1 - score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

2.6 Lingkungan Pengembangan

Seluruh eksperimen dilakukan menggunakan Python 3.12 dengan memanfaatkan pustaka scikit-learn, khususnya MLPClassifier, sebagai model utama. Pelatihan dijalankan pada platform Google Colaboratory menggunakan CPU runtime. Lingkungan komputasi ini menyediakan sumber daya CPU standar dengan kapasitas memori hingga 12,7 GB RAM.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan analisis komprehensif terhadap kinerja dan interpretabilitas model MLP dalam mendeteksi serangan DDoS pada lingkungan SDN. Pembahasan diawali dengan evaluasi kinerja berbagai variasi arsitektur MLP untuk mengidentifikasi konfigurasi yang memberikan performa terbaik. Selanjutnya, analisis confusion matrix digunakan untuk mengevaluasi kemampuan model dalam mengklasifikasikan lalu lintas normal (*benign*) dan serangan DDoS (*malicious*) secara lebih detail. Untuk mendukung aspek Interpretabilitas, analisis SHAP dilakukan guna mengetahui fitur-fitur dominan yang memengaruhi keputusan model. Terakhir, hasil yang diperoleh dibandingkan dengan penelitian sebelumnya untuk mengevaluasi kinerja arsitektur MLP yang diusulkan secara komparatif.

3.1 Evaluasi Kinerja Variasi Arsitektur MLP

Hasil eksperimen yang merangkum kinerja dari 14 variasi arsitektur MLP disajikan secara lengkap pada Tabel 4. Secara umum, hasil pengujian menunjukkan bahwa semakin kompleks arsitektur model, baik dari jumlah neuron maupun lapisan, maka semakin baik pula performanya dalam mendeteksi serangan DDoS.

Tabel 4. Perbandingan Performa Berbagai Arsitektur MLP

Arsitektur	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
32	98,35	97,45	98,36	97,90
64	99,25	98,65	99,45	99,05
128	99,54	99,27	99,55	99,41
256	99,59	99,52	99,44	99,48
64–32	99,73	99,64	99,67	99,66
128–32	99,70	99,52	99,71	99,61
128–64	99,65	99,26	99,87	99,56
256–32	99,73	99,47	99,83	99,65
256–64	99,78	99,55	99,90	99,72
256–128	99,72	99,89	99,39	99,64
128–64–32	99,70	99,94	99,30	99,62
256–128–32	99,84	99,74	99,85	99,80
256–128–64	99,71	99,52	99,74	99,63
256–64–32	99,65	99,41	99,71	99,56

Pada konfigurasi satu lapisan tersembunyi, terlihat bahwa jumlah neuron memiliki dampak signifikan terhadap akurasi. Model dengan jumlah neuron terkecil (32 neuron) menghasilkan performa terendah di antara seluruh variasi percobaan, dengan akurasi 98,35% dan F1-score 97,90%. Seiring dengan peningkatan jumlah neuron menjadi 64, 128, hingga 256, terjadi peningkatan akurasi yang konsisten hingga mencapai 99,59% pada variasi 256 neuron. Hal ini menunjukkan bahwa pada arsitektur yang dangkal, penambahan jumlah neuron memainkan peran penting dalam memperkaya representasi fitur.

Meskipun model satu lapisan dengan neuron tinggi sudah memberikan hasil yang baik, penerapan arsitektur *multilayer* menunjukkan stabilitas dan performa yang lebih superior. Seluruh konfigurasi dua lapisan, mulai dari 64–32 hingga 256–128, secara konsisten menghasilkan akurasi di atas 99,59%, melampaui performa terbaik dari model satu lapisan. Peningkatan ini menunjukkan bahwa penambahan lapisan membantu model mempelajari fitur secara bertahap, sehingga pola yang kompleks menjadi lebih mudah dipisahkan.

Puncak performa deteksi tercapai pada kategori arsitektur tiga lapisan tersembunyi. Konfigurasi 256–128–32 berhasil mencatatkan diri sebagai model terbaik dengan akurasi 99,84% dan F1-score 99,80%. Arsitektur ini menerapkan pola *funneling*, yaitu jumlah neuron yang berkurang secara bertahap dari lapisan awal menuju lapisan akhir. Pola tersebut memungkinkan model memiliki kapasitas representasi yang lebih besar pada lapisan awal untuk mempelajari hubungan nonlinier yang kompleks, sekaligus mengurangi kompleksitas model pada lapisan berikutnya.

melalui pengurangan jumlah neuron. Kombinasi ini menunjukkan keseimbangan yang baik antara kapasitas pembelajaran dan kompleksitas model, sehingga menghasilkan kinerja yang lebih tinggi dibandingkan konfigurasi lain yang diuji pada penelitian ini.

Perbandingan antara arsitektur terbaik (256–128–32) dan konfigurasi paling sederhana (32) menunjukkan perbedaan akurasi sebesar 1,49%. Meskipun perbedaannya tampak kecil secara persentase, peningkatan ini tetap penting dalam keamanan jaringan, di mana sistem harus memproses ribuan hingga jutaan paket per detik. Selisih akurasi tersebut secara langsung mengurangi jumlah paket yang salah diklasifikasikan, sehingga meningkatkan keandalan mekanisme deteksi DDoS pada lingkungan SDN. Temuan ini menegaskan bahwa pemilihan arsitektur MLP yang tepat berperan penting dalam meningkatkan performa deteksi serangan DDoS.

3.2 Analisis Confusion Matrix

Evaluasi model pada bidang keamanan jaringan tidak cukup hanya mengandalkan akurasi melainkan harus mencakup analisis kesalahan pada confusion matrix. Tabel 5 menyajikan perbandingan confusion matrix dari berbagai arsitektur MLP yang digunakan dalam penelitian ini, dengan kelas *malicious* ditetapkan sebagai kelas positif dan kelas *benign* sebagai kelas negatif. Pada lingkungan SDN, terdapat *trade-off* penting antara meminimalkan *False Positive* (FP) yaitu pemblokiran lalu lintas yang sah serta meminimalkan *False Negative* (FN) yaitu serangan yang tidak terdeteksi.

Tabel 5. Confusion Matrix Berbagai Arsitektur MLP

Arsitektur	TP	FN	FP	TN
32	8.023	134	210	12.502
64	8.112	45	111	12.601
128	8.120	37	60	12.652
256	8.111	46	39	12.673
64–32	8.130	27	29	12.683
128–32	8.133	24	39	12.673
128–64	8.146	11	61	12.651
256–32	8.143	14	43	12.669
256–64	8.149	8	37	12.675
256–128	8.107	50	9	12.703
128–64–32	8.100	57	5	12.707
256–128–32	8.145	12	21	12.691
256–128–64	8.136	21	39	12.673
256–64–32	8.133	24	48	12.664

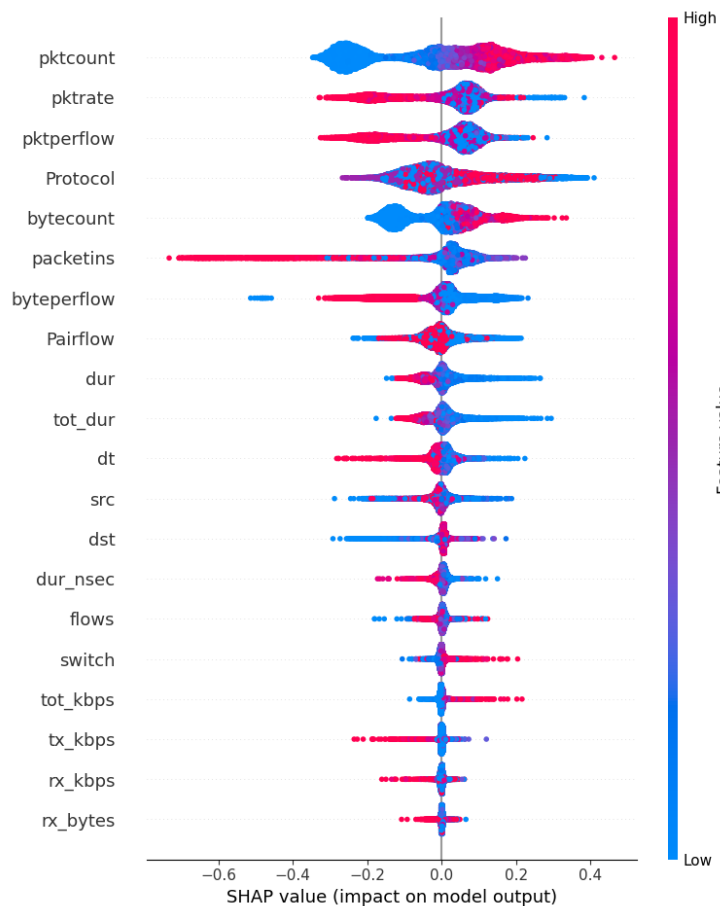
Hasil eksperimen menunjukkan perbedaan karakteristik yang cukup jelas antar variasi arsitektur MLP. Arsitektur 128–64–32 menghasilkan jumlah FP paling rendah (5) dan TN paling tinggi (12.707), sehingga sangat baik dalam meminimalkan kesalahan klasifikasi pada lalu lintas normal. Namun, model ini juga menghasilkan FN yang relatif tinggi (57), sehingga berisiko meloloskan lebih banyak serangan DDoS. Sebaliknya, arsitektur 256–64 menghasilkan FN paling rendah (8) dan TP paling tinggi (12.675), menunjukkan kemampuan deteksi serangan yang sangat baik. Akan tetapi, model ini menghasilkan FP yang lebih tinggi (37). Perbedaan tersebut menunjukkan adanya *trade-off* antara kemampuan mendeteksi serangan dan meminimalkan gangguan terhadap lalu lintas normal.

Untuk kebutuhan sistem yang memerlukan keseimbangan antara kedua aspek tersebut, arsitektur 256–128–32 menunjukkan kinerja yang paling baik. Model ini mampu menjaga FN dan FP tetap rendah, yaitu masing-masing 12 dan 21 dari total 20.869 sampel uji. Dibandingkan dengan model sederhana berlapis tunggal dengan 32 neuron, yang menghasilkan 344 total kesalahan prediksi, peningkatan performa pada arsitektur 256–128–32 terlihat jauh lebih baik. Kombinasi sensitivitas yang tinggi terhadap serangan siber, dan akurasi yang baik dalam mengenali lalu lintas normal menjadikan arsitektur ini paling robust dalam mendeteksi serangan DDoS pada SDN.

3.3 Analisis SHAP

Meskipun model MLP memiliki performa klasifikasi yang sangat tinggi, sifatnya sebagai *black box* seringkali menyulitkan administrator jaringan untuk memahami dan memverifikasi logika di balik deteksi serangan. Untuk

mengatasi hal ini, penelitian menggunakan pendekatan SHAP pada model terbaik (256–128–32) guna memvisualisasikan kontribusi setiap fitur terhadap keputusan prediksi. Hasil analisis ini disajikan dalam SHAP Summary Plot pada Gambar 2.



Gambar 2. SHAP Summary Plot pada Model Terbaik

Gambar 2 mengurutkan fitur berdasarkan tingkat kepentingannya dari atas ke bawah, di mana sumbu x merepresentasikan nilai SHAP. Nilai SHAP positif mendorong prediksi ke arah kelas serangan DDoS (*malicious*), sedangkan nilai negatif mendorong ke arah lalu lintas normal (*benign*) dengan warna merah merepresentasikan nilai tinggi dan biru merepresentasikan nilai rendah. Berdasarkan plot tersebut, *pktcount* (jumlah paket), *pktrate* (laju paket), dan *pktperflow* (paket per aliran) teridentifikasi sebagai tiga fitur paling dominan yang menjadi penentu utama dalam mendeteksi DDoS. Temuan ini menegaskan bahwa model MLP secara efektif mempelajari karakteristik dasar serangan DDoS. Analisis mendalam terhadap fitur *pktcount* dan *bytecount* memperlihatkan pola distribusi yang konsisten. Titik-titik berwarna merah pada kedua fitur ini terkonsentrasi di sisi positif. Hal ini mengindikasikan bahwa lonjakan jumlah paket dan ukuran byte yang besar merupakan indikator kuat yang digunakan model untuk mengklasifikasikannya sebagai serangan [24]. Logika ini selaras dengan mekanisme serangan DDoS tipe *flooding*, di mana penyerang membanjiri jaringan dengan volume lalu lintas yang masif.

Sebaliknya, fitur *pktrate* dan *pktperflow* menunjukkan perilaku yang berlawanan. Pada kedua fitur ini, titik berwarna merah justru berkumpul di sisi negatif, sementara titik berwarna biru menyebar di sisi positif. Pola ini menegaskan bahwa model mengenali aliran dengan jumlah paket per aliran yang sedikit dan laju paket yang rendah sebagai karakteristik utama serangan. Sementara itu, aliran dengan jumlah paket banyak dan laju tinggi cenderung dianggap sebagai lalu lintas normal.

Temuan ini selaras dengan Wang et al. [24] yang menyatakan bahwa nilai *packet rate* dan *packet per flow* justru menurun drastis saat terjadi serangan DDoS. Hal ini disebabkan oleh mekanisme serangan DDoS dengan teknik *spoofing*, di mana penyerang menggunakan ribuan alamat IP palsu untuk membanjiri *flow table*. Akibatnya, tercipta ribuan aliran koneksi pendek yang masing-masingnya hanya membawa sedikit paket. Hal tersebut berbeda dengan lalu lintas normal, seperti streaming, yang memiliki durasi aliran stabil dengan laju paket yang tinggi. Interpretasi

visual melalui SHAP ini menunjukkan bahwa model MLP yang dikembangkan tidak sekadar menghafal data latih, melainkan berhasil menangkap pola perilaku lalu lintas jaringan yang valid. Kemampuan model membedakan antar karakteristik fitur yang menegaskan keandalannya dalam mendeteksi serangan DDoS pada SDN.

3.4 Perbandingan dengan Penelitian Sebelumnya

Untuk melakukan perbandingan dengan penelitian sebelumnya, hasil penelitian ini dibandingkan dengan dua studi terdahulu yang juga menggunakan dataset DDoS attack SDN dengan proporsi pembagian data yang identik, yaitu 80% train dan 20% test. Ringkasan perbandingan kinerja dengan penelitian sebelumnya disajikan pada Tabel 6.

Tabel 6. Perbandingan Kinerja dengan Penelitian Sebelumnya

Model	Akurasi (%)	Presisi (%)	Recall (%)	F1-score (%)
MLP (Penelitian ini)	99,84	99,74	99,85	99,80
MLP [12]	99,05	98,66	98,92	98,79
Logistic Regression [11]	83,69	83,31	82,46	82,26
Support Vector Classifier (SVC) [11]	85,83	85,79	87,46	86,61
K-Nearest Neighbor [11]	95,22	96,83	94,37	95,58
Random Forest (RF) [11]	97,2	96,56	95,45	96,23
Ensemble Classifier [11]	97,5	96,43	96,43	96,72
ANN [11]	98,2	97,43	97,84	97,12
SVC-RF [11]	98,8	98,27	97,91	97,65

Berdasarkan pada Tabel 5, terlihat bahwa pendekatan MLP yang diusulkan secara umum mampu mengungguli metode machine learning tradisional maupun metode ensemble learning. Penelitian oleh Ahuja et al. [11] menunjukkan bahwa algoritma klasik seperti Logistic Regression dan SVC memiliki keterbatasan dalam menangani kompleksitas fitur serangan DDoS, dengan hasil akurasi masing-masing hanya 83,69% dan 85,83%. Bahkan pendekatan hybrid SVC-RF yang diusulkan oleh Ahuja et al. [11] yang sebelumnya mencatatkan performa terbaik dalam studi mereka (98,80%), masih berada di bawah kinerja model MLP yang diusulkan dalam penelitian ini dengan selisih akurasi sebesar 1,04%.

Perbandingan yang lebih spesifik dilakukan terhadap penelitian Rahman & Sutabri [12], yang juga menerapkan algoritma MLP. Meskipun menggunakan algoritma yang sama, model MLP dengan arsitektur 256–128–32, yang dikembangkan dalam penelitian ini mampu melampaui kinerja MLP standar yang digunakan oleh Rahman & Sutabri [12]. Tercatat peningkatan akurasi dari 99,05% menjadi 99,84%, serta peningkatan F1-score dari 98,79% menjadi 99,80%. Peningkatan ini membuktikan bahwa konfigurasi jumlah layer dan neuron yang tepat, khususnya penggunaan struktur tiga lapisan dengan pola *funneling* dapat berperan krusial dalam meningkatkan kapasitas model untuk mempelajari representasi fitur yang lebih detail dibandingkan arsitektur standar.

4. KESIMPULAN

Penelitian menunjukkan bahwa evaluasi arsitektur MLP melalui variasi kedalaman lapisan dan jumlah neuron berpengaruh terhadap kinerja deteksi serangan DDoS pada lingkungan SDN. Berdasarkan evaluasi komprehensif, arsitektur MLP dengan tiga lapisan tersembunyi menunjukkan superioritas dibandingkan arsitektur dangkal, di mana konfigurasi 256–128–32 menjadi model terbaik. Arsitektur ini tidak hanya mencapai akurasi tertinggi sebesar 99,84% dan F1-score 99,80%, tetapi juga menawarkan stabilitas dengan menekan tingkat kesalahan prediksi seminimal mungkin, yaitu hanya menghasilkan 21 FP dan 12 FN. Temuan ini menegaskan bahwa penggunaan struktur jaringan saraf tiruan dengan pola *funneling* bekerja efektif dalam menyaring fitur-fitur kompleks dari lalu lintas jaringan. Penelitian ini juga berhasil mengatasi tantangan *black box* melalui pendekatan SHAP. Analisis interpretabilitas mengungkap bahwa fitur *pktcount*, *pktrate*, dan *pktperflow* merupakan indikator paling dominan yang digunakan model dalam mengidentifikasi serangan DDoS. Selain itu, pendekatan MLP yang diusulkan terbukti unggul secara konsisten di seluruh metrik evaluasi dibandingkan penelitian-penelitian sebelumnya. Penelitian selanjutnya disarankan untuk mengimplementasikan model secara *real-time* pada lingkungan SDN serta melakukan evaluasi menggunakan *cross-validation* guna menguji konsistensi performa dan potensi *overfitting* model.

DAFTAR PUSTAKA

- [1] U. B. Clinton, N. Hoque, and K. Robindro Singh, "Classification of DDoS attack traffic on SDN network environment using deep learning," *Cybersecurity*, vol. 7, no. 1, 2024, doi: 10.1186/s42400-024-00219-7.
- [2] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A New Framework for DDoS Attack Detection and Defense in SDN Environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020, doi: 10.1109/ACCESS.2020.3021435.
- [3] N. S. Shaji and R. Muthalagu, "Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN," *Digital Communications and Networks*, vol. 10, no. 6, pp. 1716–1731, 2024, doi: <https://doi.org/10.1016/j.dcan.2023.09.004>.
- [4] N. M. Kaliyamurthy, S. Taterh, S. Shanmugasundaram, A. Saxena, O. Cheikhrouhou, and H. Ben Elhadj, "Software-Defined Networking: An Evolving Network Architecture - Programmability and Security Perspective," *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/9971705.
- [5] W. Hill *et al.*, "DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies," *Discover Applied Sciences*, vol. 6, no. 9, p. 472, 2024, doi: 10.1007/s42452-024-06172-x.
- [6] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS Attack Detection Method Based on SVM in Software Defined Network," *Security and Communication Networks*, vol. 2018, no. 1, p. 9804061, 2018, doi: <https://doi.org/10.1155/2018/9804061>.
- [7] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Detection of DDoS Attack on SDN Control plane using Hybrid Machine Learning Techniques," in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2018, pp. 299–303. doi: 10.1109/ICSSIT.2018.8748836.
- [8] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking (SDN)," *Journal of Computer Networks and Communications*, vol. 2019, no. 1, p. 8012568, 2019, doi: <https://doi.org/10.1155/2019/8012568>.
- [9] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.
- [10] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–5. doi: 10.1109/ICCCI50826.2021.9402517.
- [11] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, p. 103108, 2021, doi: <https://doi.org/10.1016/j.jnca.2021.103108>.
- [12] H. Rahman and T. Sutabri, "Analysis DDoS Attack Using Machine Learning On Software-Defined Network Architectures," *Jurnal Sistem dan Informatika (JSAI)*, vol. 7, no. 3, Nov. 2024, doi: 10.36085/jsai.v7i3.7301.
- [13] S. Ahmed *et al.*, "Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron," *Future Internet*, vol. 15, no. 2, 2023, doi: 10.3390/fi15020076.
- [14] W.-W. Tay, S.-C. Chong, and L.-Y. Chong, "DDoS Attack Detection with Machine Learning," *Journal of Informatics and Web Engineering*, vol. 3, no. 3, pp. 190–207, 2024, doi: 10.33093/jiwe.2024.3.3.12.
- [15] A. Sanmorino, L. Marnisah, and H. D. Kesuma, "Detection of DDoS Attacks using Fine-Tuned Multi-Layer Perceptron Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16444–16449, 2024, doi: 10.48084/etasr.8362.
- [16] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Comput. Secur.*, vol. 88, p. 101645, 2020, doi: <https://doi.org/10.1016/j.cose.2019.101645>.
- [17] P. Prasad, M. Tahir, and J. Isoaho, "Enhancing Explainability of Artificial Intelligence for Threat Detection in SDN-based Multicast Systems," *Procedia Comput. Sci.*, vol. 257, pp. 569–574, 2025, doi: <https://doi.org/10.1016/j.procs.2025.03.073>.
- [18] M. A. Setitra, M. Fan, B. L. Y. Agbley, and Z. E. A. Bensalem, "Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment," *Network*, vol. 3, no. 4, pp. 538–562, 2023, doi: 10.3390/network3040024.

- [19] N. Ahuja, G. Singal, and D. Mukhopadhyay, "DDOS attack SDN Dataset," 2020, *Mendeley Data*. doi: 10.17632/jxpfjc64kr.1.
- [20] S. Alam, M. S. Ayub, S. Arora, and M. A. Khan, "An investigation of the imputation techniques for missing values in ordinal data enhancing clustering and classification analysis validity," *Decision Analytics Journal*, vol. 9, p. 100341, 2023, doi: <https://doi.org/10.1016/j.dajour.2023.100341>.
- [21] Q. Zhao, F. Wang, W. Wang, T. Zhang, H. Wu, and W. Ning, "Research on intrusion detection model based on improved MLP algorithm," *Sci. Rep.*, vol. 15, no. 1, p. 5159, 2025, doi: 10.1038/s41598-025-89798-0.
- [22] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7499–7519, 2024, doi: 10.1109/TNNLS.2022.3229161.
- [23] A. Sanmorino, L. Marnisah, and H. Di Kesuma, "Detection of DDoS Attacks using Fine-Tuned Multi-Layer Perceptron Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16444–16449, Oct. 2024, doi: 10.48084/etasr.8362.
- [24] S. Wang *et al.*, "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques," *Engineering Science and Technology, an International Journal*, vol. 35, p. 101176, 2022, doi: <https://doi.org/10.1016/j.jestch.2022.101176>.