International Journal of Advanced Trends in Engineering and Management
(ISSN 2583-7052)

Vol. V, No. 01, pp. 31-38, January

An Efficient Intrusion Detection Framework for Software-Defined Networks Using Chaotic Panda Optimization and Multi-Scale Transformer Attention Network

P. Kavitha[3]

[3]Department of Electrical and Electronics Engineering, Government College of Engineering Tirunelveli, Tamilnadu, India.

ABSTRACT: Software-Defined Networks (SDN) have emerged as a flexible networking paradigm by separating the mechanism plane from the data plane, allowing centralized network managing as well as dynamic traffic control. In this paper an Efficient Intrusion Detection (ID) Framework based on Chaotic Panda Optimization (CPO) and a Multi-Scale Transformer Attention Network (MSTAN) Intrusion Detection (ID) for SDN environments. A network traffic data are collected and pre-processed through data cleaning and missing value handling to improve data quality. Data visualization is then performed to analyse traffic patterns and identify abnormal behaviours, followed by feature selection to retain the most informative attributes. The proposed MSTAN captures both local as well as global dependences in network traffic through multi-scale attention mechanisms for accurate ID. Further to improve MSTAN parameter the CPO algorithm is used enhancing convergence speed and detection performance. Experimental results is demonstrated in Python software using SDN Intrusion Detection dataset the proposed structure attains an accuracy of 99.3%, outperforming existing ID approaches. The proposed framework provides an intelligent, reliable, and computationally efficient solution for real-time ID in SDN.

Keywords: Software-Defined Networks, Intrusion Detection, Chaotic Panda Optimization, Multi-Scale Transformer Attention Network

1. Introduction

Computer networks have develop an important part of present life, supporting communication, business, healthcare, education, and many other services [1-2]. As Internet usage continues to grow rapidly, SDN systems are increasingly exposed to a wide range of security threats [3]. At the same time, the expansion and complexity of the Internet have directed to a significant increase in the number as well as sophistication of cyber-attacks and vulnerabilities [4-5]. SDN is important because it serves as the central control unit of the network [6]. SDN is compromised by an attacker, the entire network become unavailable its ability to operate securely and efficiently [7]. Because of this reason numerous techniques are utilized various IDS have been developed to identify and mitigate cyber-attacks in SDN and other similar network environments.

2. Related Works

Several techniques are utilized ID for SDN. In [8] Swilled et al (2025) have proposed Natural Language Processing (NLP) and BERT to evaluate network traffic patterns and recognize potential cyber-attacks in SDN. However, BERT contains a large number of parameters and requires substantial computational resources. Consequently, in [9] John et al (2026) have invented DenseNet–Alex Net Integrated Vision with Attention-Based ID (DAIV-ABID) to improve ID. Nevertheless, the complex architecture overfit when the intrusion dataset is

highly imbalanced. Additionally, in [10] Alotaibi and Jamal (2025) have developed SDN traffic-control approach combines a Squeeze-and-Excitation (SE) enhanced Bidirectional Long Short-Term Memory (Bi-LSTM) network with the Honey Badger Algorithm (HBA). HBA automatically search for suitable model parameters, reducing manual tuning. Nonetheless, performance depends on the configuration of both the neural network and HBA. Furthermore, in [11] Alsulami et al (2024) have proposed Hybrid Genetic Fire Hawk Optimizer (HGFHO) with Residual Dense-assisted Multi-Attention Transformer (Op-Red MAT) is used for ID. HGFHO identify the most relevant features and reduce redundant information. Nevertheless, it increase the time required to obtain the optimal feature subset. Moreover, in [12] Veers et al (2025) have presented a Trident Auto former Fusion Network (Trident-AFNet) with the LyrebirdNizar Integrated Tuner (LyNi-Tuner) to improve ID in SDN. Trident-AFNet capture traffic patterns at different temporal scales. However, introduce latency in rapidly changing SDN environments. Existing DL techniques are struggle to handle the ID for SDN to overcome these issue this paper propose an MSTAN-CPO.

2.1 Research Contribution

- Data pre-processing based on data cleaning as well as missing-value handling is used to improve the quality and reliability of the collected network traffic data.
- Data visualization and feature selection is utilized to preserve the most significant network traffic characteristics.
- The proposed MSTAN captures both local as well as global dependences in network traffic using multi-scale attention mechanisms.
- CPO algorithm is used to optimize MSTAN parameters, improving convergence and classification performance whereas reducing manual parameter tuning.

3. Proposed Methodology

For SDN detection at first, data pre-processing based on data cleaning as well as handling missing values, removing inconsistent records to improves data quality and model reliability. Data Visualization is used to examine traffic distributions, feature variations, and abnormal network behaviours, identify patterns associated with normal and attack traffic. Feature Selection is used to select relevant network traffic features to remove redundant and less informative attributes. Then the proposed MSTAN technique learns local traffic characteristics and long-range global dependencies, allowing the model to separate complex intrusion patterns from legitimate network activities. CPO algorithm is incorporated to optimize MSTAN parameters. The optimization process searches for an effective parameter configuration that improves model convergence and intrusion classification performance. The proposed block diagram is presented in figure 1 ID for SDN using CPO-MSTAN method.

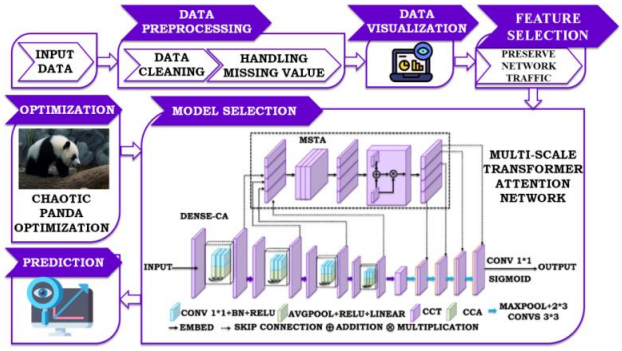

Figure 1: *Proposed block diagram*

3.1 Data Pre-processing

In ID for SDN the collected SDN Intrusion Detection dataset are pre-processed through data cleaning and missing-value handling to improve data quality. Inconsistent, duplicate, and erroneous records are identified and removed, whereas missing values are appropriately treated to maintain dataset completeness. This pre-processing step reduces noise and improves the

reliability, consistency, and learning performance for further visualization.

3.2 Data Visualization

Data visualization is used to analyse the distribution of SDN network traffic features and identify variations among different traffic patterns. It displays abnormal behaviours and unusual patterns that indicate potential intrusion activity. The visualized data is fed to feature selection.

3.3 Feature Selection

In ID for SDN feature selection identifies the most relevant network traffic features whereas removing redundant, irrelevant, and less informative attributes. It reduces the dimensionality of the input data, lowers computational complexity, and effectively distinguishes normal traffic from malicious activities. The selected features are fed to model selection based on MSTAN-CPO.

3.4 Model Selection

The proposed MSTAN is used to analyse selected SDN network traffic features at multiple scales and identify complex intrusion patterns. Transformer block in MSTAN enhances the embedded patches by learning the relationships between different feature representations. The structure of MSTAN is displayed in figure 2.

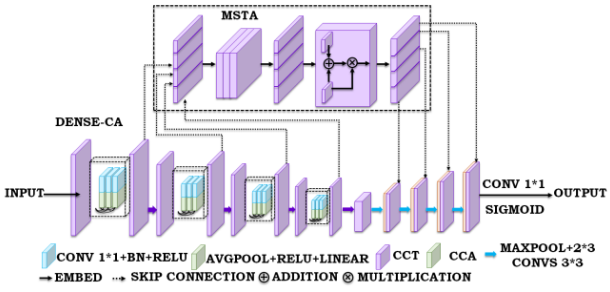

Figure 1: *Structure of MSTAN*

The block employs a self-attention mechanism to detect and gives the most informative features for SDN ID, which is defined as follows:

$$Attention\ (Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_{head}}}\right)V \quad (1)$$

Wherever, $Q, K, \& V$ signify the query, key, as well as value matrices, d_{head} indicate the dimension of every attention head as well as controls the level of detail captured during local feature interactions. The $Q, K, \& V$ matrices have the same dimensions of $HW \times C$, resulting in a computational complexity of$O((HW)^2)$. As the resolution increases, the number of spatial elements grows substantially, leading to higher computational and memory requirements and making efficient processing more challenging. For fine tuning the MSTAN parameter it is fed to CPO.

3.4 Chaotic Panda Optimization

The chaotic search strategy improves population diversity and CPO avoids premature convergence, allowing the model to identify a better parameter configuration for accurate intrusion classification. The flowchart for CPO is shown in figure 3.

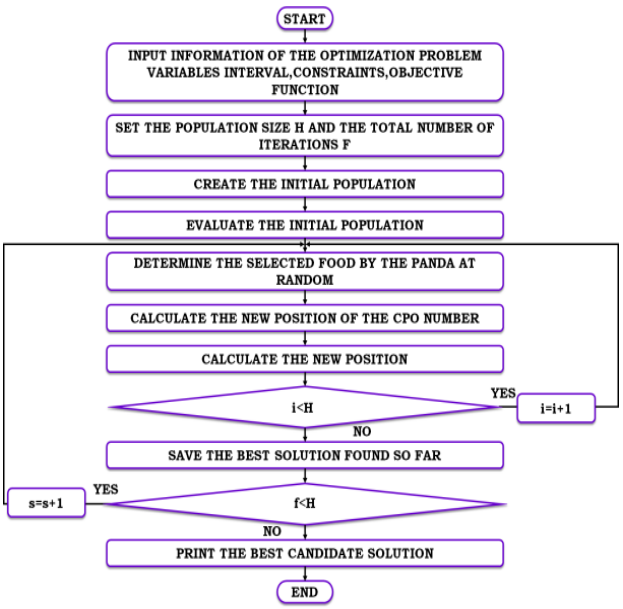

Figure 2: *Flowchart for CPO*

CPO is used in the proposed SDN ID framework to optimize the important parameters of the MSTAN

$$\lambda_{p,q} = \beta_q + l_{p,q}.(\mu_q - \beta_q) \quad (2)$$

$$(\lambda_{p,q})^{u1} = \lambda_{p,q} + \hbar\nabla * \left| \varsigma_{p,q}\left(\Theta\ (R)\right) - \rho \times \lambda_{p,q} \right| \quad (3)$$

$$b\lambda_{p,q} = \begin{cases} \{IF((\varsigma_{fit})^{u1} < \varsigma_{fit}), (\lambda_{p,q})^{u1} \\ ELSE, \quad \lambda_{p,q} \end{cases} \quad (4)$$

Where, R signifies the population matrix containing the positions of the pandas. The index $q = 1,2, \ldots, Q$ indicates the decision variables, whereas $p = 1,2, \ldots, P$ symbolizes the number of data transmissions between the global leader (GL) and sub-leader (SL). The parameters $\mu_q - \beta_q$ designate the lower as well as upper bounds of the q^{th} decision variable, individually then $l_{p,q}$ randomly generated value within the range[0,1]. Thus the MSTAN parameter is tuned using the CPO.

4. Result and discussion

In ID for SDN the work is implemented in Python software using SDN Intrusion Detection dataset with X shapes of 1188333, 40 and Y shapes of 1188333. The data is spit as 80% for train then 20 % for test with X train shape of 950666, 40, and then Y train shape of 950666. Also the X test shape of 237667, 40 and Y test shape of 237667 respectively.

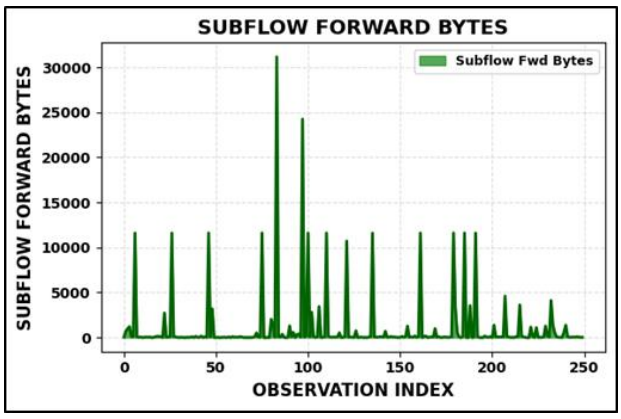

Figure 3: *Sub flow forward bytes*

Figure 4 displays sub flow forward bytes. Across 250 consecutive observation indices, it highlighting a maximum data transmission peak of over 30,000 bytes near index 85 along with several recurring spikes around 10,000 to 12,000 bytes.

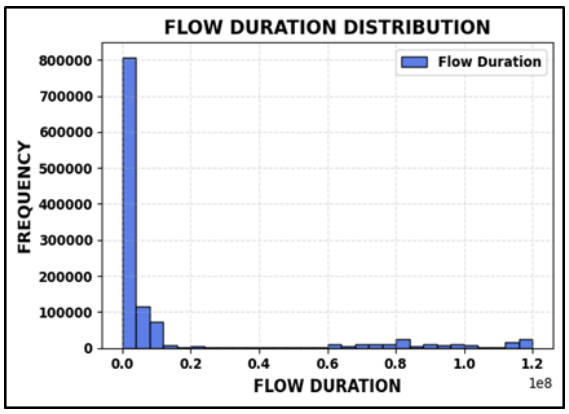

Figure 4: *Flow duration distribution*

Figure 5 shows the flow duration distribution. The heavily right-skewed frequency distribution histogram of network flow durations is happen with a massive concentration of over 800,000 flows occurring near the0 and a long, spare tail spreading out to 1.2×10^8 units.

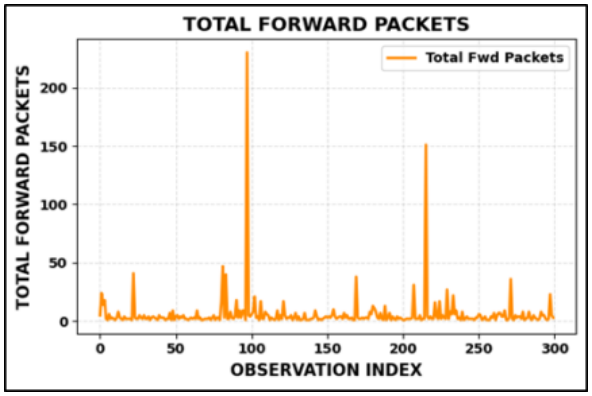

Figure 5: *Total forward packets*

Figure 6 displays the total forward packets across 300 observation directories. It features a stable baseline with distinct large spikes reaching over 200 packets near index 100 and around 150 packets near index 215 respectively.

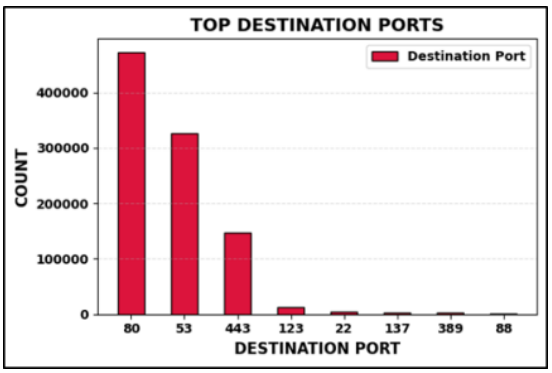

Figure 6: *Top distribution ports*

Figure 7 shows the top distribution ports. The count of network traffic characterized by the top destination port numbers, with port 80 (HTTP) showing the highest frequency monitored by port 53 (DNS) and port 443 (HTTPS).

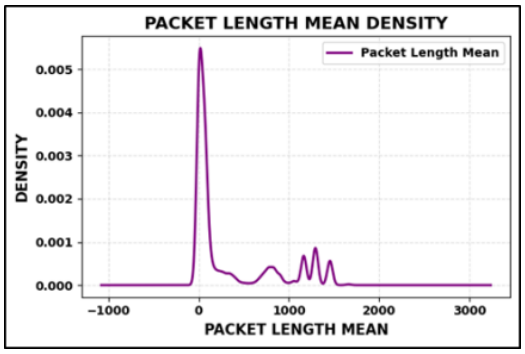

Figure 7: *Packet length mean density*

Figure 8 displays packet length mean density. A massive concentration of data values near 0 to 100 bytes, presenting a high peak density of approximately 0.0055.

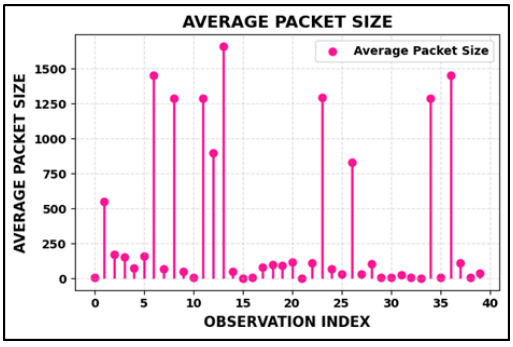

Figure 8: *Average packet size*

Figure 9 shows average packet size. The maximum peak occurs at observation index 13, then vertical line extends past the 1500 packet size on the y-axis, reaching exactly halfway among 1500 and the top grid line area, mapping directly to 1650.

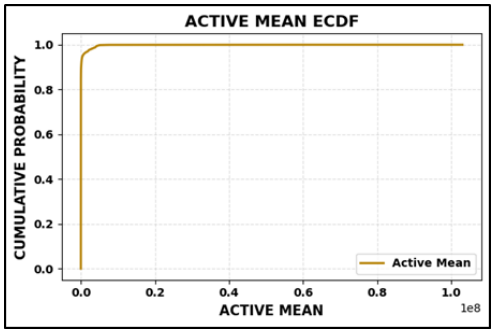

Figure 9: *Active mean ECDF*

Figure 10 displays active mean ECDF. It rises sharply to approximately 1.0 cumulative probability at an Active Mean of nearly $0.05\times10^{8}$and remains close to 1.0 up to $1.0\times10^{8.}$

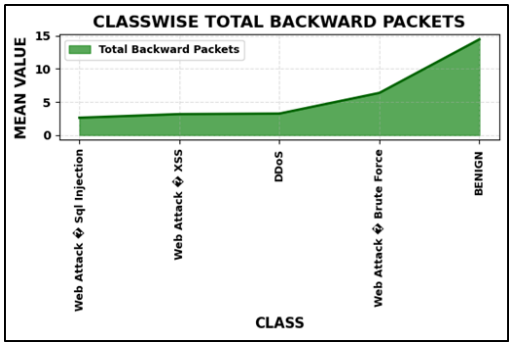

Figure 10: *Class wise total backward packets*

Figure 11 shows class wise total backward packets. It increase from nearly 2.5 for Web Attack–Sql Injection to 14.5 for BENIGN, through intermediary values of about 3.0 (Web Attack–XSS), 3.2 (DDoS), and 6.2 (Web Attack–Brute Force).

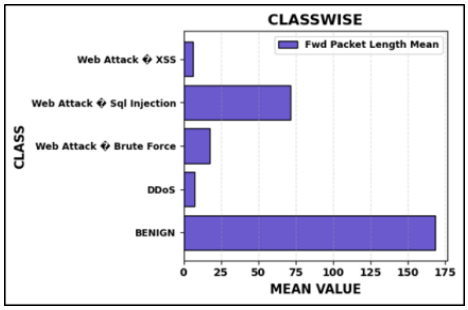

Figure 11: *Class wise*

Figure 12 displays class wise. The network traffic classes ranked from maximum to lowermost by their average forward packet length BENIGN is 168, Web Attack Sql Injection is 72, Web Attack Brute Force is 18, DDoS is 8 and Web Attack XSS is 6 respectively.

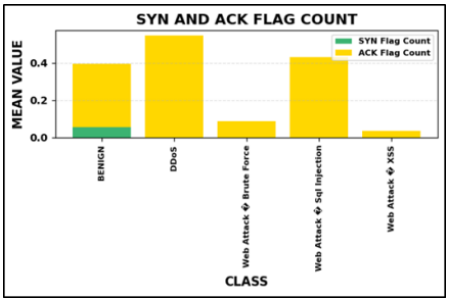

Figure 12: *SYN and ACK flag count*

Figure 13 shows SYN and ACK flag count. It have highest for DDoS with 0.55, followed by SQL Injection with 0.43 and BENIGN with 0.40, whereas Brute Force with 0.09 and XSS with 0.03 display much lower values.

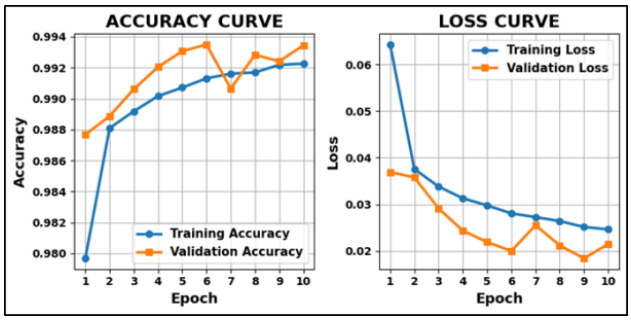

Figure 13: *Accuracy & loss curve*

Figure 14 displays accuracy & loss curve for proposed MSTAN-CPO. The training and validation accuracy increase sharply by epoch 2, then steady between 99.1% and 99.3% accuracy. The training and validation losses drop suddenly from epoch 1 to epoch 3, continuing a gradual downward trend with minor fluctuations like validation loss spikes at epochs 7 and 10. The close tracking of training and validation metrics without severe divergence proposes a well-generalized model without overfitting.

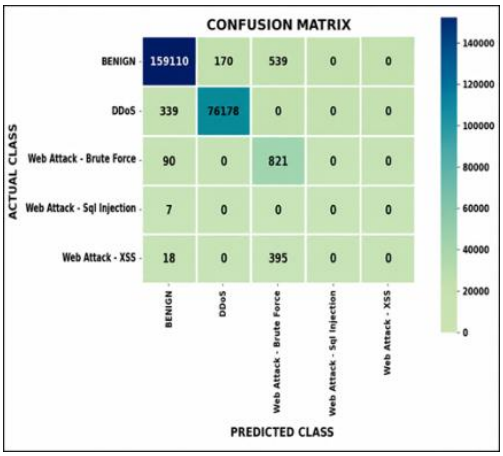

Figure 14: *Confusion matrix*

Figure 15 displays the confusion matrix for proposed MSTAN-CPO. The model correctly identifies 159,110 benign instances and 76,178 DDoS instances on the main diagonal. A significant number of actual benign records are misclassified as Web Attack - Brute Force of 539 & DDoS of 339. Web Attack - Sql Injection shows zero correct predictions misclassifications mapped across the matrix entries for that row

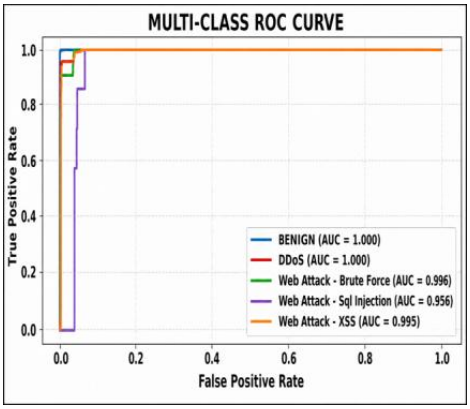

Figure 15: *Multi-class ROC-Curve.*

Figure 16 displays the Multi-class ROC-Curve for proposed MSTAN-CPO. The BENIGN and DDoS have the AUC of 1, Web Attack-Brute Force as well as Web Attack-XSS have the AUC of 0.99 the Web Attack-Sql injection have the AUC value of 0.95 it is low compared to other classes.

4.1 Comparisons with proposed technique

The proposed MSTAN-CPO technique is compared with the existing technique. The accuracy for Wahab et al [3] is 92.5%, Ahmed et al [2]is 98.6%, Sankar et al [1] is 98.9%, John et al [9]is 99%, Alsulami et al [11] is 99.1% and proposed MSTAN-CPO have the accuracy of 99.3% it is high compared with the existing techniques as presented in table 1.

Table 1: *Evaluation with the proposed methods*

Study	Accuracy (%)
Wahab et al [3]	92.5
Ahmed et al [2]	98.6
Sankar et al [1]	98.9
John et al [9]	99
Alsulami et al [11]	99.1
Proposed technique	99.3

5. Conclusion

In this paper an MSTAN-CPO structure is proposed for ID in SDN. The data pre-processing, improve the quality, data visualization technique

visualize the processed data, and feature selection detects the input traffic data. MSTAN model captures both local and global dependencies in network traffic through multi-scale attention, whereas the CPO algorithm optimizes the model parameters to improve convergence and detection performance. Experimental evaluation using the SDN Intrusion Detection dataset in Python demonstrates that the proposed framework attains an accuracy of 99.3%, outperforming the considered existing ID methods. The MSTAN-CPO framework offers an accurate, reliable, and computationally efficient solution for SDN security. Future work focuses on incorporating explainable AI (XAI) for security administrators understand the reasons behind attack predictions.

References

1. Sasi Rekha Sankar; B. Jothi., Year: 2026, “EGOA-DL-IDS: explainable intrusion detection in SDN-based industrial cyber physical system with optimized deep learning techniques,” Cluster Computing, Vol: 29, No: 4, pp. 242.
2. USMAN Ahmed; Muhammad Tariq Sadiq, Year: 2026, “Enhancing SDN Intrusion Detection via Multi-Hybrid Deep Learning Fusion and Explainable AI,” Mathema*tics*, Vol: 14, No: 9, pp. 1498.
3. Sheikh Abdul Wahab; Saira Sultana; Noshina Tariq; Maleeha Mujahid; Javed Ali Khan; Alexios Mylonas, Year: 2025, “A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers,” Sensors 25, No: 15,pp. 4845.
4. K. Sivakumar; Kakoli Banerjee; R. Vibin; B. Saravanan; Satyajee Srivastava; R. Anand; V. Bhoopathy, Year: 2024, “AI-driven green network management for future Internet and software-defined networking (SDN),”.
5. Ayşe Okutan Kara; Aytuğ Boyacı, Year: 2026, “Transformer–DDQN-Based Explainable and Active Intrusion Detection Architecture for Network Traffic Analysis,”Applied Sciences, Vol: 16, No: 12, pp. 5912.
6. Laraib Sana; Muhammad Mohsin Nazir; Jing Yang; Lal Hussain; Yen-Lin Chen; Chin Soon Ku; Mohammed Alatiyyah, Sulaiman Abdullah Alateyah, and Lip Yee Por, Year: 2024, “Securing the IoT cyber environment: Enhancing intrusion anomaly detection with vision transformers,” IEEE Access, Vol: 12, pp. 82443-82468.
7. Zhenyue Long; Huiru Yan; Guiquan Shen; Xiaolu Zhang; Haoyang He; Long Cheng, Year: 2024, “A Transformer-based network intrusion detection approach for cloud security,” Journal of Cloud Computin*g*, Vol: 13, No: 1, pp. 5.
8. Mohammed N Swileh; Shengli Zhang, Year: 2025, “Unseen attack detection in software-defined networking using a BERT-based large language model,”AI, Vol: 6, No: 7, pp. 154.
9. Sirajudeen Ameer John; Senthilnathan Palaniappan P; Gopichand G; Ganesan G, Year: 2026, “Advanced IoT intrusion detection using hybrid attention-based deep learning with DenseNet-ViT and AlexNet-BERT,” Journal of Cloud Computing.
10. Jamal Alotaibi, Year: 2025, “A hybrid software-defined networking approach for enhancing IoT cybersecurity with deep learning and blockchain in smart cities,” Peer-to-Peer networking and Applications, Vol: 18, No: 3, pp. 123.
11. Majid H Alsulami, Year: 2024, “Residual dense optimization-based multi-attention transformer to detect network intrusion against cyber-attacks,” Applied Sciences, Vol: 14, No: 17, pp. 7763.
12. G. B Veeresh; Vanita Jaitly; V. Lokeswara Reddy, Year: 2025, “Swish-Optimized trident fusion

network for precise attack classification in software-defined networks," Knowledge-Based Systems, pp. 114860.