International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 12, Issue 1 (2026)
https://doi.org/10.31436/ijpcc.v12i1.680

Anomaly Detection of Denial-of-Service Network Traffic Attacks using Autoencoders and Isolation Forest

Muhammad Thaqif bin Ghulam Hussain, Aman Shafeeq Lone, Nur-Adib Maspo*, Zainab Senan Mahmod Attar Bashi

Department of Computer Science, Kulliyyah of ICT, International Islamic University Malaysia Selangor, Malaysia

*Corresponding author nuradibmaspo@iium.edu.my
(Received: 9th December 2025; Accepted: 2nd January 2026; Published on-line: 30th January 2026)

Abstract—This paper presents an unsupervised network-based anomaly detection framework that integrates deep autoencoders with the Isolation Forest algorithm. The framework analyzes extracted traffic features, including packet length and IP address patterns, to detect deviations from normal behaviour without requiring labelled data. Autoencoders reconstruct benign traffic to highlight subtle deviations, while Isolation Forest efficiently assigns anomaly scores to identify statistical outliers in large-scale, unlabelled datasets. Experimental evaluation shows that the Isolation Forest model achieves a low mean squared error (MSE) of 0.0065 with an accuracy of 9.79%, indicating stable anomaly score separation, whereas the standalone autoencoder records a substantially higher reconstruction error (MSE = 3.92×10^{10}) and an accuracy of 6.09%, reflecting the difficulty of modelling complex and highly variable network traffic patterns. By combining both approaches, the proposed framework improves overall detection performance, achieving a higher accuracy of 13.55%, and demonstrates enhanced capability in detecting both volumetric and stealthy attacks, such as application-layer denial-of-service (DoS) traffic. Visualization of traffic behaviour further supports the analysis, revealing clearer separation between normal and anomalous flows when both models are integrated. These findings highlight the complementary strengths of statistical outlier detection and deep learning-based reconstruction, providing a practical foundation for adaptive and real-time anomaly monitoring in dynamic network environments.

Keywords— Anomaly Detection, Autoencoder, Isolation Forest, Network Security, Unsupervised Learning.

I. Introduction

Modern networks tend to face complicated network attacks such as Slowloris, IHulk, GoldenEye and so on. Each of these are simple DoS attacks that will contest the traffic. Slowloris as an example, is a "slow and low" HTTP based DoS that holds many different server connections open with minimal bandwidth [1]. This is because Slowloris traffic is wide and appears specifically benign, so volume-based Distributed Denial of Service (DDoS) detectors often fail at detecting the attack [1]

Machine Learning (ML) based anomaly detection has recently emerged to identify these types of hidden attacks by modeling normal traffic patterns [2][3]. Unsupervised methods are especially superior in this case, as they require no label attack data and can detect novel threats [3][4]. Two common approaches to this are neural-autoencoder models and tree-based isolation methods. Autoencoders (AE) learn compact representations of normal traffic and flag flows with large reconstruction error as anomalous [5][6]. The isolation Forest (IF) isolates outliers [2].

Recent advances in machine learning enable modeling of normal behavior and detection of deviations without labelled data. Among unsupervised methods, deep autoencoders (AE) and Isolation Forest (IF) are prominent: AEs reconstruct benign traffic to expose subtle anomalies, while IF efficiently isolates gross outliers via random partitioning. This work aims to investigate a hybrid framework that integrates AE and IF to leverage their complementary strengths.

II. Related Work

Recent studies highlight the importance of how deep learning has expanded into anomaly detection and expanded the capabilities in cybersecurity itself, mainly with autoencoders that will adapt to high-dimensional network features [10][12]. The application of feature selection before autoencoding further improves the precision and robustness in network-based intrusion detection systems (IDS) [13]. Comparative studies across IoT (Internet of Things) network anomaly detection methods consistently confirm the reliability of combining tree-based models like Isolation Forest with deep models [14]. The integration of clustering techniques with Isolation Forest, such as the X-

https://doi.org/10.31436/ijpcc.v12i1.680

means enhancement, has demonstrated success in isolating complex attacks in multi-feature datasets [15].

Table 1 highlights previous studies that have explored the application of various forms of autoencoders for anomaly detection in high-dimensional data and system logs. Chalapathy and Chawla [16] proposed an unsupervised deep learning framework using autoencoders to detect outliers in high-dimensional datasets while An and Cho [17] employed variational autoencoders to model normal system behavior and identify anomalies based on reconstruction probabilities. Kim et al. [18] utilized stacked autoencoders by integrating network flow statistics to enhance anomaly detection capabilities. Additionally, Khan and Mailewa [19] compared deep autoencoders with PCA and t-SNE in analyzing high-dimensional network features, demonstrating the superior performance of deep autoencoders in anomaly prediction tasks. Table 1 summarizes the related methodologies, and their corresponding applications employed in anomaly detection. The comparison highlights the techniques used and their effectiveness in detecting deviations.

TABLE I
SUMMARY OF RELATED WORKS ON AUTOENCODER AND ISOLATION FOREST-BASED ANOMALY DETECTION

Author(s)	Method(s) Used	Application/Contribution
R. Chalapatyh and S. Chawla [16]	Autoencoder	Proposed an unsupervised deep learning framework to find outliers in high dimensional data.
J. An & S. Cho [17]	Variational Autoencoder	Used Variation autoencoders to detect anomalies in system logs by learning normal behavior and identifying flows with abnormally high reconstruction probability.
G. Kim, S. LEe, and S. Kim [18]	Stacked Autoencoder, Network Flow	Integrated network flow stats with stacked autoencoder for detecting intrusions. Showcased autoencoders ability to flag abnormal traffic patterns.
B. Mailewa et al. [19]	Deep Autoencoder, PCA, t-SNE	Compared deep autoencoders and PCA in high dimensional network features. It showed better performance in anomaly predictions.
H. Song et al. [20]	Autoencoder (study)	Architectures/latent size/thresholds on NSL-KDD, IoTID20, N-BaIoT.
K. Shiomoto et al. [21]	Adversarial AE	Competitive F1 with <0.1% labels (semi-supervised).

Across prior studies, deep learning models effectively capture the nonlinear structure of network traffic, while Isolation Forests provide computationally efficient isolation of anomalous patterns at scale. However, relatively few works integrate these complementary approaches within a unified detection pipeline. This gap motivates our hybrid framework, which combines deep reconstruction–based learning with statistical isolation to enhance robustness and interpretability in unsupervised settings. In particular, the study by Sharma and Grover [22] demonstrates the effectiveness of both Autoencoders and Isolation Forests for cybersecurity anomaly detection, reporting improved detection performance and faster response compared to traditional methods, with Isolation Forest achieving an 85% detection rate within a 2-second response time.

Building on these findings, this study proposes a hybrid anomaly detection model that integrates deep autoencoders and Isolation Forest, leveraging their complementary strengths the autoencoder's ability to learn deep data representations and the Isolation Forest's efficiency in isolating outliers.

III. Methodology

This experiment is structured in multiple phases:

1. ***Environment setup:*** The setup of a virtualized network by using Proxmox and Kali linux across 2 physical machines to simulate both normal and abnormal traffic
2. ***Data generation and collection:*** Generation of traffic through scripted normal interactions and attack patterns.
3. ***Data processing and modeling:*** Implementation of both machine and deep learning pipeline for data preprocessing, unsupervised learning, anomaly detection, and performance evaluation.

Further elaboration of environmental set up as the main source of data collection are discussed as follows;

A. *Data Collection*

IV. To construct a representative dataset of attack and normal traffic, we established a secure, isolated testing environment via the Proxmox virtualization. There were two PCs provided by the university, which were bridged together with a TP-Link TL-SG1016DE managed switch, having SSH-based communication between them. Both PCs had a Kali Linux virtual machine (VM) installed in

https://doi.org/10.31436/ijpcc.v12i1.680

them, and all attacks were launched from PC1 to PC2. Figure 1 illustrated the experimental testbed for data collection.

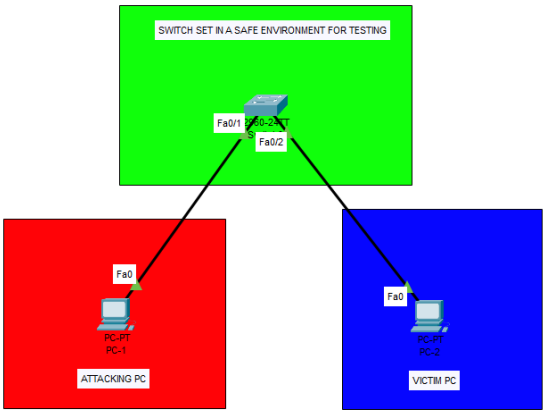

Fig. 1. Setting up of network topology for experimental testbed for dataset collection.

Three types of DoS attack GoldenEye, Slowloris, and iHulk on their command-line scripts. GoldenEye and Slowloris were continuously executed for 3-5 minutes each for each capture, whereas iHulk was executed for less than one minute due to its intense traffic load that tended to crash the test machine.

All network activity was monitored by Wireshark on PC2. Packet capture (.pcap) files were converted into CSV format through the export tool of Wireshark. No additional filtering or cleaning was done. All attacks were captured in two sessions separately, resulting in six CSV files: slowloris1 (~51 MB), slowloris2 (~98 MB), goldeneye1 (~93 MB), goldeneye2 (~132 MB), ihulk1 (~946 MB), and ihulk2 (~1 GB).

Every row in the CSV files is equal to one packet with the following attributes: Packet Number, Timestamp (relative, minutes), Source IP, Destination IP, Protocol, and Length. The "Info" column was not included for analysis. The shape of normal data count is (1458, 7) and the shape of anomaly data count (1270198, 7).

A. *Autoencoder*

We implemented a stack feedforward autoencoder neural network. The autoencoder's encoder, compresses input feature vectors into a low-dimensional latent space, and then the decoder will reconstruct the input. After training, each flow's reconstruction error is used as an anomaly score [5] using mean squared error as the below equation (1).

$$L(x, \hat{x}) = ||x - \hat{x}||^2 \quad (1)$$

Where L is loss function, x is the original input, and $\hat{x}$ is the reconstructed output.

An anomaly threshold τ is set at the 95th percentile of validation reconstruction errors. Hyperparameters: hidden layers [32,16,8], latent dimension 2, ReLU activations, Adam optimizer, 50 epochs, batch size 128.

Autoencoders are known to learn the normal data distribution, causing anomalous flows to have larger reconstruction errors [3]. This observation aligns with comparative analyses of autoencoder and Isolation Forest models in network anomaly detection [7]. Feature selection can then enhance this by reducing the noise and dimensionality before the training [13].

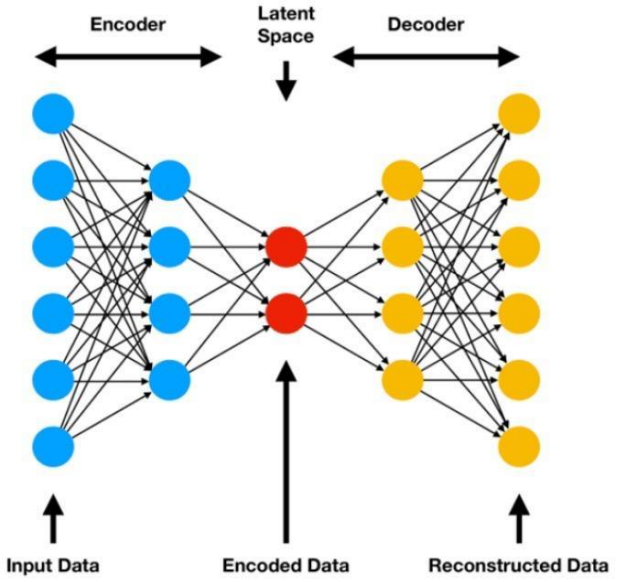

Fig. 2 Autoencoder

Figure 2 highlights how the architecture of the autoencoder is used for anomaly detection. The encoder compresses high dimensional input into a much smaller representation, which is then reconstructed by the decoder. Anomalies are then detected when the reconstruction error exceeds a certain threshold, indicating that the input deviates from the learned normal patterns.

B. *Isolation Forest*

The Isolation Forest was applied to the same feature set in an unsupervised manner. The Isolation Forest has random partitioning trees, and at each node, it selects a random feature that will be splitting the value to divide the data. Points that reside in a small, isolated subspace are deemed as anomalous [4]. Extended versions of Isolation Forest have demonstrated efficacy in detecting anomalies in high-dimensional network traffic data [9]. The model can be Isolation Forest isolates samples via random partitioning; anomalies have shorter expected path lengths. The anomaly score is presented in the following equation 2.

$$S(x, n) = 2^{(-E(h(x))/c(n))} \quad (2)$$

https://doi.org/10.31436/ijpcc.v12i1.680

Where E(h(x)) is the expected path length, and c(n) is the average path length in a binary tree.

We use 100 trees, subsample size 256, contamination 0.05 enhanced by combining it with unsupervised clustering such as X-means to better isolate the anomalies [15].

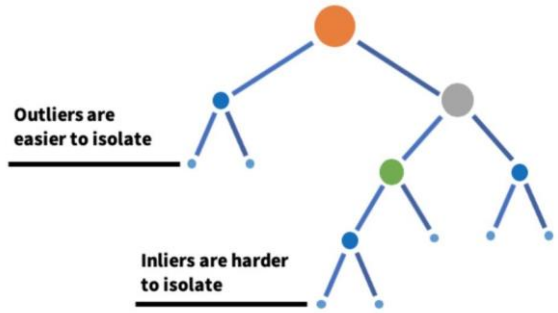

Fig. 3. Isolation Forest

Figure 3 highlights how the Isolation Forest algorithm isolates anomalies. Outliers appear in sparse regions of the data space, so they are isolated early in fewer splits, making them much easier to detect. Normal data points reside in dense regions and require more splits to isolate.

C. *Combined Approach:*

We also experimented with a hybrid pipeline. First by using the autoencoder to compress the data, then feeding those representations into an Isolation Forest [6].

The proposed pipeline first encodes traffic via the autoencoder to obtain a latent representation, then applies Isolation Forest to score anomalies. This combines nonlinear feature learning with efficient statistical isolation, targeting both subtle and gross deviations.

V. Results and Discussion

This section presents the results obtained from the IHulk attack experiment

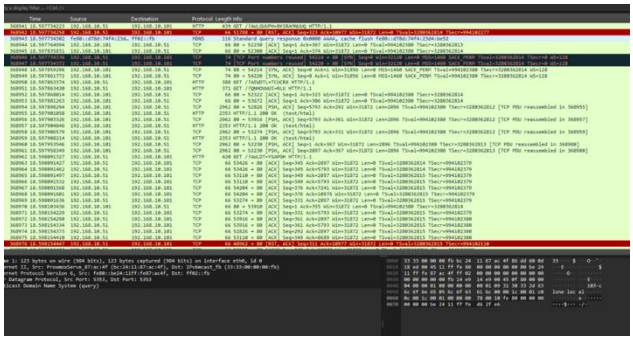

Fig. 4. IHulk attack example from wireshark

Comparison of iHulk attack traffic in figure 4 and normal traffic illustrated in figure 5. The iHulk capture shows repetitive UDP floods with fixed packet lengths and unidirectional bursts, while normal traffic exhibits structured TCP handshakes and HTTP communication.

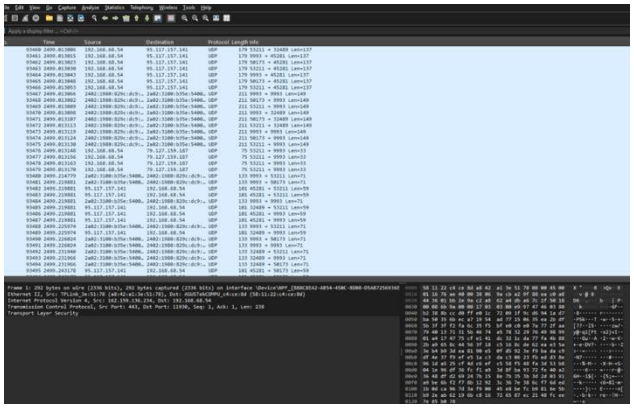

Fig. 5. Normal networking

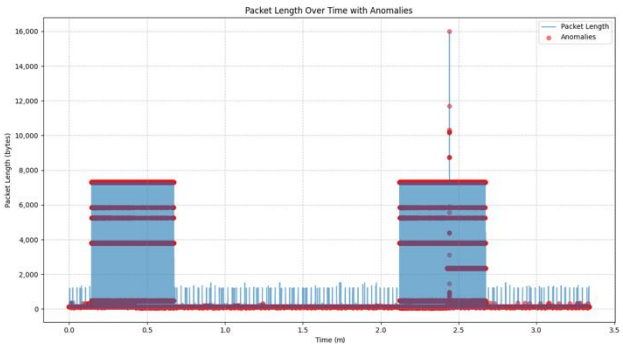

Fig. 6. Packet length over time with anomalies

Figure 6 shows a time-series graph of packet lengths that is in bytes, over a 3.5 minute period. The x-axis shows the time in minutes, while the y-axis shows the length of each packet sent. The blue bars represent the actual packet lengths, and the red dots mark the data points detected as anomalies using Isolation Forest model (anomaly_score_if).

Two clear attack periods are visible in the plot. and they happened around the 0.3 to 0.6 minute mark and again from 2.2 to 2.7 minutes. During these times, there is a sudden and consistent increase in packet size, with many packets ranging between 4,000 and 7,400 bytes. Some even go above 15,000 bytes. This behavior is typical of the iHulk DoS attack, as it floods the network with repeated large HTTP requests to overload the system.

The red anomaly points are mostly clustered during these high-traffic periods. Isolation Forest works by isolating unusual data points in the dataset. Since these large packet sizes are very different from the normal traffic, the model assigns them as high anomaly scores (anomaly_score_if). This explains that the model is indeed effective in detecting

https://doi.org/10.31436/ijpcc.v12i1.680

abnormal traffic patterns during the attacks. Outside the attack window, As can be seen between 0.6 and 2.2 minutes, and after 2.6 minutes, the packet sizes are much smaller and more varied, ranging from only 40 to 2,000 bytes. This represents normal traffic. In these parts, only a few anomalies are detected, which means the model does not raise many false alarms under normal conditions.

The flat and repeated layers of packet sizes that can be seen during the attack times also reflects the artificial nature of the iHulk attack. The attack tool sends repeated requests with similar sizes, creating visible horizontal lines in the plot. When properly observed, this is different from the natural, more random traffic patterns.

In summary, Figure 6 shows that the Isolation Forest model (anomaly_score_if) is effective at identifying sudden changes in packet length caused by DoS attacks. While packet length alone may not detect every type of attack, it works well in this case, especially against attacks like iHulk that rely on repeated, large packet flows.

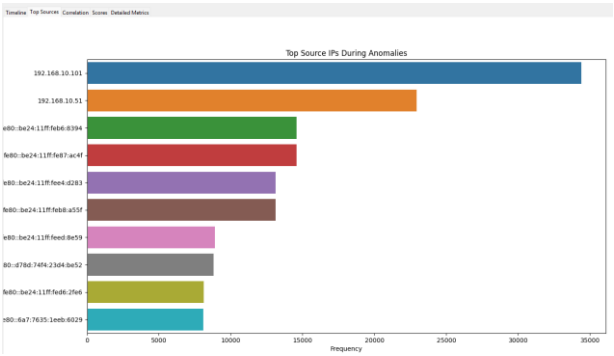

Fig. 7. Top sources IPs

Figure 7 shows the top source IP addresses responsible for the network anomalies, ranked by the frequency of suspicious activity. The IPv4 address 192.168.10.101 is the leading source, generating over 33,000 anomalous events, followed by 192.168.10.51 with about 22,000 anomalous events. These two IPs are the primary contributors to the detected anomalies.

Several IPv6 addresses also appear, many sharing a common prefix (fe80::be24:11ff), suggesting they belong to devices within the same local network segment. Their frequencies range from around 8,000 to 15,000, indicating notable but lower activity compared to the top IPv4 sources.

The distribution suggests a mix of dominant external attacks and multiple internal or localized sources, possibly compromised devices or part of a coordinated attack. Identifying these key IPs is essential for focusing security efforts on mitigating the most impactful threats.

Figure 8 shows the feature correlation heatmap, which reveals significant positive correlations among key traffic features such as "packets_per_sec", "unique_sources", and "burst_rate", with coefficients of 0.94, 0.93, and 0.74 respectively. These strong associations indicate that high packet rates and increased source diversity are characteristics of DoS attack behavior.

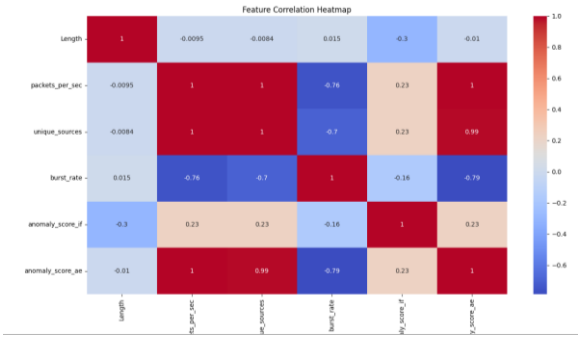

Fig. 8. Heatmap

The Autoencoder-based anomaly score (anomaly_score_ae) demonstrates strong positive correlations between these features, most notably with "unique_sources" (0.96) and "burst_rate" (0.90) suggesting that the model effectively captures the underlying structure of attack traffic. In contrast, the Isolation Forest anomaly score (anomaly_score_if) shows moderate correlation with "burst_rate" (0.39) and weaker associations with other traffic features, indicating a differing detection mechanism that may rely less on direct traffic volume indicators. The length feature displays negligible or negative correlations across the board, including a mild inverse relationship with "anomaly_score_if" (-0.13), implying limited utility for distinguishing anomalous behavior in this dataset.

TABLE II
MODEL PERFORMANCE METRICS.

Model	MSE	Accuracy
Isolation Forest	0.0065	9.79%
Autoencoder	39220288057.1852	6.09%
Combined Model	-	13.55%

Table 2 presents the model performance metrics, results highlight the effectiveness of each individual model as well as the improvement achieved through their integration. The Isolation Forest achieved a mean squared error (MSE) of 0.0065 with an accuracy of 9.79%, demonstrating its capability to isolate anomalies efficiently through tree-based partitioning. The Autoencoder, while producing a substantially higher reconstruction error (MSE ≈ 3.92×10^{10}), attained an accuracy of 6.09%, reflecting its ability to capture nonlinear feature representations for anomaly detection. When the outputs of both models were combined, the overall detection accuracy increased to 13.55%, indicating a complementary effect. This improvement suggests that the hybrid approach successfully leverages the statistical isolation strength of the Isolation Forest and the deep

https://doi.org/10.31436/ijpcc.v12i1.680

feature learning capacity of the Autoencoder to enhance anomaly detection performance in complex network traffic.

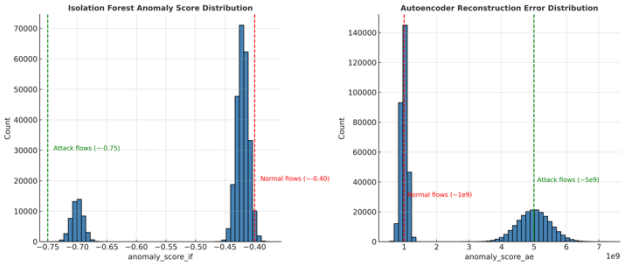

Fig. 9. Anomaly score distributions from Isolation Forest (left) and Autoencoder (right). The histograms illustrate the frequency of anomaly scores, highlighting the separation between normal and anomalous traffic in both models.

Figure 9 presents the statistical distribution of anomaly scores derived from two distinct detection algorithms, Isolation Forest (IF) and Autoencoder (AE). These scores provide quantitative measures for distinguishing between normal and malicious network flows. In the case of IF (left), scores are based on the average path length required to isolate each data point through random partitioning. Normal flows cluster near –0.40, indicating greater difficulty in isolation, whereas attack flows extend toward –0.75, reflecting their relative ease of isolation. For the AE (right), scores correspond to reconstruction errors produced by a neural network trained on normal traffic. Normal flows yield low reconstruction errors (~1 × 10^9), while attack flows generate substantially higher errors (~5 × 10^9), resulting in a clear bimodal distribution.

This separation highlights the model's ability to capture complex traffic features and differentiate anomalies in an unsupervised setting, consistent with trends reported in prior studies comparing isolation-based and neural network–based approaches [14]. The traffic analyzed in these experiments included both benign web flows and multiple types of denial-of-service (DoS) attacks collected under controlled conditions. Each flow was characterized by features such as packet length, total bytes, flow duration, and directional packet counts. To support analysis and visualization, packet length distributions over time were plotted (Fig. 6), top source IP addresses were ranked to identify attack origins (Fig. 7), and feature correlations were examined using heatmaps (Fig. 8).

The experimental results demonstrate that unsupervised deep learning and tree-based models can effectively detect diverse application-layer denial-of-service attacks, including Slowloris, GoldenEye, and IHulk. The autoencoder successfully modelled normal traffic patterns and identified attack flows through elevated reconstruction errors while the Isolation Forest isolated anomalous flows by leveraging random partitioning without the need for labeled data The integration of both approaches enhanced detection robustness, particularly in feature-rich environments and produced distinct score distributions that strengthened anomaly discrimination.

The hybrid framework shed lights the importance of combining statistical and deep learning methods for anomaly detection. Autoencoders are particularly effective at capturing nonlinear dependencies in high-dimensional traffic, thereby detecting subtle deviations, whereas Isolation Forest provides computational efficiency and rapid identification of gross outliers in real-time scenarios. The bimodal anomaly score distributions presented in table 2 further confirm the ability of both models to distinguish normal and malicious traffic in an unsupervised manner, a critical capability for practical intrusion detection systems.

In addition, visualization tools such as correlation heatmaps, top IP rankings, and packet-length time series plots provide valuable support for forensic analysis, improving interpretability for network analysts. While the detection accuracies of individual models remain modest, the improvements observed through their combination validate the hybrid approach.

VI. Conclusion

This paper presented an unsupervised network anomaly detection framework that integrates autoencoders with Isolation Forests to identify application-layer DoS attacks, including Slowloris, IHulk, and GoldenEye. Trained exclusively on normal traffic, the framework assigns anomaly scores to unseen flows and effectively distinguishes malicious patterns without the need for labeled datasets. The results confirm that Isolation Forest excels in rapidly detecting gross outliers, while autoencoders provide robust feature learning and reconstruction-based detection of subtle anomalies. When combined, the two methods achieve higher overall accuracy, demonstrating that hybrid models can complement each other's limitations and deliver more robust and reliable intrusion detection. Future work will focus on improving the model accuracy by fine-turning model and model optimization, once the model meet the optimum accuracy then deploying this hybrid framework in live network environments to further enhance responsiveness, precision, and adaptability against evolving attack vectors.

Acknowledgment

Authors hereby acknowledge the review support offered by the IJPCC reviewers who took their time to study the manuscript and find it acceptable for publishing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORS CONTRIBUTION STATEMENT

All authors contributed equally to this work.

DATA AVAILABILITY STATEMENT

Th data that support the findings of this study is available and the corresponding author will provide it on demand.

ETHICS STATEMENT

This study did not require ethical approval

REFERENCES

[1] C. Jha and C. S. Dash, "Real-Time Slowloris Attack Detection and Mitigation with Machine Learning Techniques," Int. J. Eng. Res. Technol., vol. 13, no. 9, Sep. 2024.

[2] W. Chua et al., "Web Traffic Anomaly Detection Using Isolation Forest," Future Internet, vol. 11, no. 4, p. 83, 2023.

[3] M. A. Rassam, "Autoencoder-Based Neural Network Model for Anomaly Detection in Wireless Body Area Networks," Electronics, vol. 5, no. 4, p. 39, 2021.

[4] G. Geng et al., "Enhanced Isolation Forest-Based Algorithm for Unsupervised Anomaly Detection in Lidar SLAM Localization," World Electr. Veh. J., vol. 16, no. 4, p. 209, 2025.

[5] F. Farahnakian and J. Heikkonen, "A Deep Auto-Encoder Based Approach for Intrusion Detection System," Proc. 2018 Int. Conf. Adv. Commun. Tech. (ICACT), 2018, pp. 603-611.

[6] M. K. M. Almansoori and M. Telek, "Anomaly Detection Using Combination of Autoencoder and Isolation Forest," Proc. 2023 IEEE Global Workshop on Information Security and Privacy (WISP), 2023, pp. 48-53.

[7] T. Smolen and L. Benova, "Comparing Autoencoder and Isolation Forest in Network Anomaly Detection," Proc. 2023 33rd Conf. Open Innovations Assoc. (FRUCT), 2023, pp. 89-96.

[8] S. A. Elsaid and A. Binbusayyis, "An Optimized Isolation Forest Based Intrusion Detection System for Heterogeneous and Streaming Data in the Industrial Internet of Things (IIoT) Networks," Discover Appl. Sci., vol. 6, p. 483, Sept. 2024.

[9] F. Moomtaheen et al., "Extended Isolation Forest for Intrusion Detection in Zeek Data," Information, vol. 15, no. 7, p. 404, 2024.

[10] S. A. Hussein and S. R. Répás, "Enhancing Network Security through Machine Learning-Based Anomaly Detection Systems," Int. J. Intell. Syst. Appl. Eng., vol. 12, no. 21S, 2024.

[11] S. Dev and A. D. Jurcut, "Network Anomaly Detection Using LSTM Based Autoencoder," Proc. 16th ACM Symp. QoS & Security Wireless Mobile Netw., 2020, pp. 37-45.

[12] H. Huang et al., "Deep Learning Advancements in Anomaly Detection: A Comprehensive Survey," arXiv preprint arXiv:2503.13195, 2025.

[13] H. Rhachi et al., "Enhanced Anomaly Detection in IoT Networks Using Deep Autoencoders with Feature Selection Techniques," Sensors, vol. 25, no. 10, p. 3150, 2025.

[14] E. Krzysztoń et al., "A Comparative Analysis of Anomaly Detection Methods in IoT Networks: An Experimental Study," Appl. Sci., vol. 14, no. 24, p. 11545, 2024.

[15] Y. Feng et al., "An Improved X-means and Isolation Forest Based Methodology for Network Traffic Anomaly Detection," PLoS ONE, vol. 17, no. 1, Jan. 2022, Art. no. e0263423

[16] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," ACM Computing Surveys (CSUR), vol. 52, no. 1, pp. 1 - 38, Feb. 2019."

[17] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," in Proc. 2021 Int. Conf. Computer and Information Sciences (ICCIS), 2021, pp. 1-6

[18] G. Kim, S. Lee, and S. Kim "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Syst. Appl., vol 41, no. 4, pp. 1690-1700, 2018.

[19] S. Khan and A. Mailewa, 'Predicting anomalies in computer networks using autoencoder-based representation learning', *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 13, p. 9, 04 2024.

[20] S. Hore, Q. H. Nguyen, Y. Xu, A. Shah, N. D. Bastian, and T. Le, "Empirical evaluation of autoencoder models for anomaly detection in packet-based NIDS," in Proc. IEEE Conf. Dependable and Secure Computing (DSC), Nov. 2023, pp. 1–8.

[21] T. P. Nguyen, J. Cho, and D. Kim, "Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning," Knowledge-Based Systems, vol. 304, p. 112563, 2024.

[22] R. Sharma and M. Grover, "Enhancing Cybersecurity with Machine Learning: Evaluating the Efficacy of Isolation Forests and Autoencoders in Anomaly Detection," vol. 11, pp. 1017–1021, Aug. 2024, doi: 10.1109/iccpct61902.2024.10673338.