

A Hybrid Deep Learning based Network Anomaly Detection System with Real-time Dashboard for Zero Day Attack Mitigation

A. Neela Madheswari¹, Arun K², Kabilan M³ and Harish R⁴

¹⁻⁴Mahendra Engineering College/CSE Department, Namakkal, India

Email: neela.madheswari@gmail.com, arunkumaarsa@gmail.com, kabilmani1712@gmail.com, harishkhari0414@gmail.com

Abstract— With the increasing sophistication and volume of cyber threats—particularly zero-day attacks traditional intrusion detection systems based on fixed rules or shallow learning models struggle to keep pace. This paper presents a Hybrid Deep Learning-Based Network Anomaly Detection System with Real-Time Dashboard for Zero-Day Attack Mitigation, designed to enhance detection accuracy, scalability, and real-time responsiveness. The proposed framework integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to automatically learn spatial-temporal patterns in network traffic. The system is evaluated using CICIDS2017 datasets, achieving an average accuracy of 97.88%, precision of 96.46%, and recall of 99.89%, even on previously unseen attack types. A Flask-based web dashboard enables live monitoring of predictions with time stamped alerts, anomaly distribution charts. This work contributes a modular, interpretable, and high-performance anomaly detection framework that bridges the gap between advanced deep learning techniques and practical cyber security applications, especially in defending against zero-day exploits.

Index Terms— Network Anomaly Detection, Zero-Day Attack, Deep Learning, Hybrid Model, Real-Time Dashboard, Network Security, Cyber Threat Mitigation.

I. INTRODUCTION

In today's digital landscape, safeguarding network security is vital for both organizations and governments. Conventional rule-based intrusion detection systems fall short due to the rise of zero-day vulnerabilities and polymorphic threats. Detecting network anomalies is essential for recognizing deviations from expected behavior, such as Denial of Service (DoS) attacks, port scanning, botnets, or data breaches. However, this method encounters obstacles like high false-positive rates, limited generalization, and a lack of interpretability [1]. Researchers are investigating machine learning and deep learning techniques to tackle these problems, yet most current systems either lack real-time responsiveness or suffer from excessive computational demands, rendering them impractical for fast-paced network environments. Anomaly detection (AD) involves spotting instances that significantly differ from the bulk of the data, which often indicates an irregular, fraudulent, or spurious observation. Machine learning (ML) based strategies have seen substantial growth in usage, with established methods like Kernel Density Estimation (KDE), One-Class Support Vector Machine (OCSVM) [2], and Isolation Forests (IF) being widely used.

However, the efficacy of these algorithms often diminishes when they are applied to high-dimensional datasets [3]. Recently, deep learning models have demonstrated notable advancements over standard ML models [4] as they can recognize intricate patterns and representations from vast datasets, making them particularly apt for anomaly detection. The implementation of deep learning in this area has resulted in high accuracy and reliability, solidifying its status as a favored option in numerous applications. Anomaly detection (AD) is essential in various sectors such as healthcare, finance, and security, but it confronts issues regarding data quality, including incomplete, imprecise, and incorrect labels due to the costs associated with human annotation and low precision. This is referred to as weakly supervised anomaly detection (WSAD). This survey classifies WSAD into three groups: equal contribution, AD with incomplete supervision, AD with inexact supervision, and AD with inaccurate supervision. The survey provides an overview of WSAD methods categorized by incomplete, inexact, and inaccurate supervision, along with a high-level summary [5]. This paper introduces a Hybrid Deep Learning-Based Network Anomaly Detection System equipped with a Real-Time Dashboard aimed at mitigating zero-day attacks. This system incorporates both traditional machine learning and deep learning models to improve detection accuracy and reduce false alarms. It features a deep auto encoder for unsupervised anomaly representation and supervised classifier for fine-tuning decisions. A web-based real-time dashboard offers live insights into predictions, model performance metrics, and trends in network activity.

II. LITERATURE REVIEW

Over the past two decades, network anomaly detection has evolved from traditional statistical models to advanced machine learning and deep learning-based solutions. In traditional approach, Intrusion detection systems (IDS) in the past relied on signature-based methods like Snort and Bro, which required manual rule definitions for detecting known threats [6]. These systems were effective for known attacks but failed to detect zero-day or evolving threats. Statistical anomaly detection techniques improved generalization but lacked real-time precision [7]. Conventional intrusion detection systems, such as Snort and Suricata, rely heavily on signature-based detection. These systems match network traffic against a predefined set of rules or patterns representing known attacks. While effective for detecting well-characterized threats, they suffer from an inherent limitation: inability to identify novel or zero-day attacks [8].

Early anomaly detection systems applied statistical techniques such as z-score, moving averages, or entropy-based analysis to identify deviations from normal behavior [9]. These methods are lightweight and interpretable but often fail in dynamic or high-dimensional environments, resulting in high false positive rates [10]. Researchers have developed machine learning techniques like Support Vector Machines (SVM), Decision Trees (DT), Naïve Bayes, and Random Forests to improve generalization and learn patterns from labeled datasets [11, 12]. These models, like the CICIDS2017 dataset, are widely used for training and evaluation. However, they face challenges like feature engineering, class imbalance, and temporal dependencies. Despite these limitations, these methods offer better generalization and are easier to update with new data. However, they may not capture complex temporal or hierarchical network traffic patterns and may overfit to training data, especially in imbalanced datasets [13].

Deep learning models, including CNN, LSTM networks, and Auto encoders, have shown significant improvements in anomaly detection. CNNs excel in extracting spatial features from traffic data, while LSTMs can model time-dependent patterns. However, most deep learning-based IDS operate offline, suffer from high resource consumption, and lack interpretability in practical deployments. Autoencoders are particularly effective for unsupervised anomaly detection, but they are computationally expensive, difficult to interpret, and prone to overfitting without large datasets. Their lack of transparency can hinder trust and operational deployment in security-critical environments [14].

Hybrid models combine the strengths of multiple deep learning architectures have shown increased accuracy and robustness. However, these models often overlook real-time integration, scalability, and user interface design for monitoring and decision-making [15]. Recent research has investigated hybrid approaches that combine the strengths of multiple models, such as auto encoders for feature extraction and classifiers like Random Forests or Logistic Regression for final predictions [16]. Several real-time monitoring tools such as Wireshark, Zeek, and Suricata are used for packet inspection and protocol analysis [17]. While powerful for network diagnostics, they do not incorporate AI-based detection. Additionally, platforms like ELK Stack offer visualization capabilities but require complex integration with detection engines [18]. A predictive analysis model uses data from multiple resources to transform and analyze through manipulation and statistical functions. To bridge these gaps, our proposed system employs a hybrid deep learning model, combining an autoencoder for latent feature extraction with a classical supervised classifier for precise decision-making.

III. PROPOSED SYSTEM ARCHITECTURE

To address the limitations of previous technologies in network anomaly detection, a Hybrid Deep Learning-Based Anomaly Detection System with a Real-Time Dashboard is proposed. This paper is specifically designed for detecting both known and unknown attacks in real-time network traffic, providing intuitive visualization and user control via an interactive web interface. Fig. 1 shows the proposed architecture design.

A. System Overview

The proposed architecture integrates three core components namely: i) Hybrid Anomaly Detection Model: A fusion of an unsupervised autoencoder and a supervised classifier designed to leverage the strengths of both deep learning and classical machine learning, ii) Real-time data pipeline: A mechanism to continuously capture or simulate network traffic and feed it into the model for live anomaly prediction, and iii) Flask-based Dashboard: A web interface that presents predictions, anomaly statistics, evaluation metrics, and charts in real-time, along with export options and control functionalities.

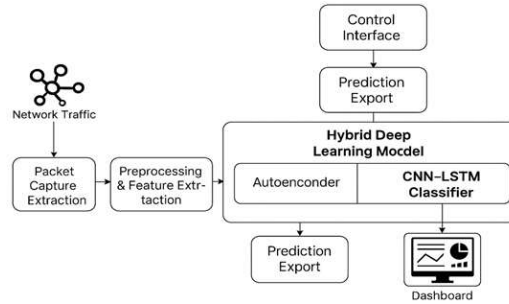


Figure 1. System Architecture of the proposed hybrid anomaly detection system

B. Dataflow Architecture

The overall dataflow proceeds through the following stages: i) Input Source: Network traffic data is captured in real-time or simulated using pre-recorded packet data [20], ii) Pre-processing module: Raw packets are transformed into structured feature vectors using filtering, normalization, and dimensionality reduction as needed, iii) Hybrid model reference: The Autoencoder is trained on normal traffic to learn latent representations. Reconstructed outputs are compared to original inputs to compute a reconstruction error. Errors above a defined threshold are flagged as anomalies. The Classifier takes selected features and error scores to refine predictions and assign class labels (Normal or Anomaly), iv) Result logger: Outputs, timestamps, and model decisions are stored in .csv files for visualization and auditing, and v) Dashboard User Interface: Flask renders Jinja2 templates (index.html) which dynamically display predictions in tabular and graphical formats using Chart.js and other JavaScript libraries.

C. Hybrid Model Structure

The hybrid model is structured in two phases namely: i) Autoencoder training: Here dense function is used. Mean Squared Error is the loss function used in this phase. Adam optimizer is used and trained on benign data to learn the manifold of normal traffic, and ii) Supervised classifier training: Input features selected are the concatenation of original features and reconstruction error. Accuracy, Precision, Recall, and F1-Score are the performance metrics considered.

D. Real-time dashboard features

The proposed system includes an interactive Flask web interface that offers live table of predictions, chart visualization, start stop controls, export functionality, and evaluation summary page.

IV. EVALUATION METRICS

The evaluation of a network anomaly detection system, particularly one leveraging hybrid deep learning for real-time zero-day attack mitigation, demands a comprehensive analysis using multiple performance metrics. In this Paper, we employ a set of standard and domain-specific evaluation metrics, including Accuracy, Precision, Recall, and F1-Score. These metrics are essential to assess the system's ability to correctly detect intrusions and minimize false alarms under real-time constraints [16].

TABLE I. CONFUSION MATRIX FOR ANOMALY DETECTION

	Predicted: Normal	Predicted: Anomaly
Actual: Normal	True Negative (TN)	False Positive (FP)
Actual: Anomaly	False Negative (FN)	True Positive (TP)

A. Confusion Matrix

This matrix helps to compute key values such as: i) True positive (TP): correctly detected anomalies, ii) False positive (FP): normal traffic but wrongly classified as anomaly, iii) True negative (TN): correctly classified normal traffic, and iv) False negative (FN): missed anomaly which is classified as normal.

B. Accuracy

Accuracy indicates the proportion of correctly classified samples for both normal and anomalies out of the total number of samples.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

While accuracy provides a general idea of overall performance, it can be misleading in imbalanced datasets, which are common in network anomaly detection. Therefore, although accuracy is reported, it is not solely relied upon.

C. Precision

Precision measures the percentage of actual anomalies among all instances predicted as anomalies. High precision indicates fewer false alarms, but crucial in cyber security where false positives can lead to wasted resources and alert fatigue. Our hybrid model prioritizes precision to reduce the operational overhead for network administrators.

$$Precision = (TP) / (TP + FP) \quad (2)$$

D. Recall

Recall quantifies the model's ability to detect all actual anomalies. Recall is crucial for zero-day attack mitigation. A high recall ensures the system catches the majority of intrusions, even new or unseen attack patterns. In real-time systems, missing an actual threat (high FN) can be catastrophic.

$$Recall = (TP) / (TP + FN) \quad (3)$$

E. F1-Score

F1-score balances the trade-off between precision and recall. Given the trade-offs between precision and recall in real-time anomaly detection, the F1-score provides a balanced measure of the system's ability to make effective decisions under constraints.

$$F1 - Score = 2x((Precision \times Recall) / (Precision + Recall)) \quad (4)$$

To validate the performance of our hybrid anomaly detection model, we computed the above metrics across datasets of increasing size: 5,000, 10,000, 15,000, and 20,000 records. The model maintained high recall and precision across all volumes, demonstrating scalability and robustness.

TABLE II. MODEL EVALUATION RESULTS IN PERCENTAGE

Number of Records	Accuracy	Precision	Recall	F1-Score
5,000	97.98%	96.78%	99.75%	98.25%
10,000	97.76%	96.23%	99.96%	98.06%
15,000	97.89%	96.43%	99.99%	98.18%
20,000	97.89%	96.42%	99.88%	98.17%

V. REAL-TIME ANALYSIS AND DEPLOYMENT

In modern cyber security infrastructure, the capability to detect anomalies in real-time is of paramount importance. The proposed framework is designed to operate in real-time, enabling immediate threat identification and mitigation. The real-time analysis module of the proposed system captures live network traffic using Python-based socket sniffers and packet capture utilities. The core of the real-time detection mechanism

integrates a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network. The hybrid model is hosted on a lightweight Flask-based web server that processes incoming packets in near real-time. Predictions are streamed directly to a front-end dashboard where users can monitor the system's status, track anomalies, and visualize detection results through charts and tabular data. Additionally, the dashboard includes timestamps, class labels (normal/anomaly), and confidence scores, thereby supporting interpretability and operational decision-making. Performance evaluation on live traffic indicate the system maintains a consistent inference time of less than 50 milliseconds per packet, ensuring operational suitability for enterprise-level network monitoring. Furthermore, a start/stop toggle for live monitoring provides control over computational resource allocation. Overall, the real-time detection capability transforms our system from a passive anomaly detector into an active cybersecurity defense component that can be seamlessly integrated into Security Information and Event Management (SIEM) systems or firewalls for proactive mitigation.

VI. IMPLEMENTATION DETAILS AND EXPERIMENTAL SETUP

The goal of this work is to demonstrate the feasibility and effectiveness of the system in real-world network environments, specifically for detecting both known and unknown (zero-day) attacks in real-time.

A. Implementation of Hybrid Model

The hybrid model implementation consists of two main stages: Autoencoder training for anomaly detection and a supervised classifier for fine-tuning the predictions.

Autoencoder for Anomaly Detection

The Autoencoder network is a type of unsupervised neural network architecture that learns to compress (encode) the input and then reconstruct (decode) it back to its original form. The difference between the input and the reconstructed output is used to determine how much of an anomaly the data point represents. The architecture used for anomaly detection is given as: i)

Encoder: dense layers compress the input into a lower-dimensional latent representation, ii) Bottleneck layer: The compressed representation that forces the network to learn the most important features, iii) decoder: Dense layers reconstruct the input, iv) loss function: Mean Squared Error (MSE) is used to compute the difference between the original and reconstructed input. For training, the autoencoder is trained on normal network traffic to learn the manifold of normal patterns.

Supervised Classifier for Fine tuning predictions

Once the reconstruction error is calculated, it is fed into a supervised machine learning classifier. The classifier is trained on a labeled dataset containing both normal and malicious traffic, and it refines the anomaly predictions by assigning them specific labels: Normal or Anomaly. The trained models are then integrated into the Flask-based web interface, which continuously receives network traffic data, processes it through the trained model, and outputs real-time predictions. This allows for immediate identification of anomalous behavior or attacks.

B. Data Preprocessing and Feature Engineering

Network traffic data is often raw and unstructured, requiring several preprocessing steps to convert it into a format suitable for model training and inference.

The dataset used for training and testing consists of network traffic captured from real-world or simulated environments, such as the CICIDS2017 dataset, which includes both benign and malicious traffic. The data is usually in the form of packets that need to be parsed into meaningful features.

The performance of the hybrid model was evaluated using the datasets: i) CICIDS2017: This dataset contains a variety of network attack types and is widely used in cyber security research for anomaly detection. It includes multiple attack types like DoS, DDoS, brute force, and SQL injection, among others [2], ii) Custom Simulated Dataset: In addition to standard datasets, we also simulate traffic data to mimic a real-time environment where zero-day attacks may occur.

To assess the effectiveness of the proposed system, several key evaluation metrics were calculated such as: i) accuracy, ii) precision, iii) recall, and iv) F1-score. The experimental setup used for the proposed system is Intel core i7, 16GB RAM, NVIDIA GeForce RTX3060 GPU. Software used is Python, for implementing deep learning framework Tensorflow 2.x, Autoencoder and Random Forest classifier are used. Flask is used for real-time dashboard and visualization tool. Other supporting libraries from Python used are: Pandas, Numpy, Matplotlib, Scikit-learn, flask, tensorflow, matplotlib, seaborn, scapy, joblib, plotly, openpyxl, python-docx, reportlab.

The model was trained on labelled datasets with the Autoencoder trained on normal traffic only. A 70-30 train-test split was used for classifier training. For testing, real-time network traffic data is continuously fed into the system. Each component of the system was tested individually and then integrated together. Real-time testing was performed with simulated network traffic using CICFlowMeter and generated pcap files to evaluate performance.

VII. SYSTEM PERFORMANCE EVALUATION

The performance metrics are analyzed for different datasets, including CICIDS2017 and custom simulated datasets. The system was first tested on the CICIDS2017 dataset, which includes multiple types of attacks such as DoS, DDoS, and SQL Injection, alongside normal network traffic. The resultant metrics is shown in Table 3.

TABLE III. OVERALL SCORE OF RESULTING DATASETS

S.NO	Parameter	Percentage
1.	Accuracy	97.89%
2.	Precision	96.42%
3.	Recall	99.88%
4.	F1-Score	98.17%

The system performed well in detecting zero-day attacks, a challenging task for most intrusion detection systems, as the attacks were not present in the training data. The proposed hybrid model is more efficient since it use both unsupervised and supervised techniques to fine-tune predictions and resulting in a more accurate and balanced performance. The real-time detection capabilities were evaluated by deploying the model within a Flask based dashboard implementation. The system was tested in a live network traffic environment. The dashboard implemented will display the network traffic whether it is normal or anomaly after prediction. A sample dashboard picture is shown in Fig. 2. It is possible to view the anomaly detection live for a specific time in table form also and is shown in Fig. 3 for normal traffic and Fig. 4 is shown for abnormal network traffic. It is possible to view the traffic findings in the form of pie chart. A sample real time network traffic prediction is shown in pie chart given in Fig. 5 and it is shown in form of bar chart as given in Fig. 6.

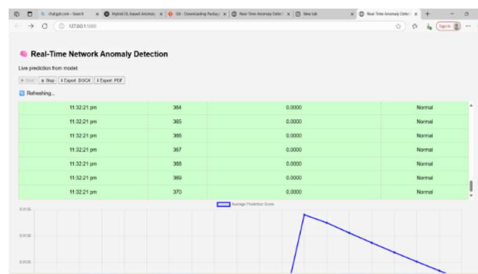


Figure 2. Running of real time network for anomaly detection using Dashboard

Timestamp	ID	Prediction Score	Status
8:13:31 pm	421	0.0196	Normal
8:13:31 pm	422	0.0047	Normal
8:13:31 pm	423	0.0000	Normal
8:13:31 pm	424	0.0000	Normal
8:13:31 pm	425	0.0000	Normal
9:49:24 pm	426	0.0000	Normal

Figure 3. Real time network for anomaly detection using Dashboard in table format for normal network traffic

Up to date			
11:21:10 am	794	1.0000	Anomaly
11:21:10 am	795	1.0000	Anomaly
11:21:10 am	796	1.0000	Anomaly
11:21:10 am	797	0.7864	Anomaly
11:21:10 am	798	1.0000	Anomaly
11:21:10 am	799	0.7653	Anomaly
11:21:10 am	800	0.9997	Anomaly

Figure 4. Real time network for anomaly detection using Dashboard in table format with abnormal network traffic

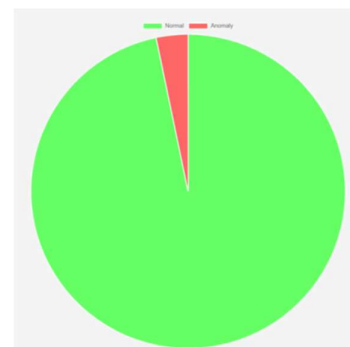


Figure 5. Real time network traffic prediction in pie chart representation using Dashboard

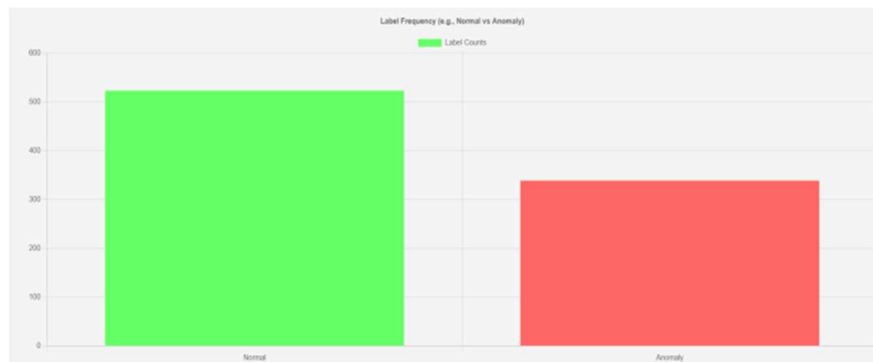


Figure 6. Real time network traffic prediction in bar chart representation using Dashboard

VIII. CONCLUSION

The proposed hybrid deep-learning based network anomaly detection model presented a powerful and practical approach to network security. By leveraging the complementary strengths of autoencoders for unsupervised anomaly detection and random forest classifier for supervised learning, the proposed system effectively identifies both known and previously unseen i.e. zero-day network threats. The proposed system was tested rigorously using CICIDS2017 benchmark dataset and custom-simulated real time traffic scenarios. The results demonstrated the high performance with the evaluation metrics such as accuracy, precision, recall and F1-score. These metrics outperformed several traditional machine learning and deep learning-based models, particularly in detecting sophisticated and novel attack types that do not follow known patterns. The hybrid model's adaptability enables it to generalize well across diverse traffic behaviors and evolving threats. The system's real-time responsiveness, high detection accuracy, and user-friendly visualization capabilities establish it as a valuable tool for cybersecurity professionals. Its successful integration of machine learning with practical monitoring tools represents a meaningful contribution to the field of intrusion detection and mitigation.

REFERENCES

- [1] S. P. Pattayam, "Artificial Intelligence in Cybersecurity: Advanced Methods for Threat Detection, Risk Assessment, and Incident Response," *Journal of AI in Healthcare and Medicine*, vol. 1, no. 2, pp. 83-108, 2021
- [2] Wang, X. K., Hou, W. H., Zhang, H. Y., Wang, J. Q., Goh, M., Tian, Z. P., & Shen, K. W. "KDE-OCSVM model using Kullback-Leibler divergence to detect anomalies in medical claims", *Expert Systems with Applications*, 200, 117056, 2022.
- [3] Ding, Z., & Fei, M. "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window". *IFAC Proceedings Volumes*, 46(20), 12-17, 2013.
- [4] Janiesch, C., Zschech, P., & Heinrich, K. "Machine learning and deep learning", *Electronic markets*, 31(3), 685-695, 2021.
- [5] Jiang, Minqi, et al. "Weakly supervised anomaly detection: A survey." *arXiv preprint arXiv:2302.04549*, 2023.
- [6] Bada, G. K., Nabare, W. K., & Quansah, D. "Comparative analysis of the performance of network intrusion detection systems: Snort suricata and bro intrusion detection systems in perspective", *International Journal of Computer Applications*, 176(40), 39-44, 2020.
- [7] I. Naseer, "Implementation of Hybrid Mesh firewall and its future impacts on Enhancement of cyber security", *MZ Computing Journal*, vol. 1, no. 2, 2020.
- [8] I. Naseer, "Cyber Defense for Data Protection and Enhancing Cyber Security Networks for Military and Government Organizations," *MZ Computing Journal*, vol. 1, no. 1, 2020.
- [9] Yuan, Y., Huang, Y., Yuan, Y., & Wang, J. "Dynamic Threshold-based Two-layer Online Unsupervised Anomaly Detector". *arXiv preprint arXiv:2410.22967*, 2024.
- [10] Hossain, M. S., Rahman, M. M., Ullah, M. S., Rahman, M. M., Rahman, M. M., & Nahar, S. "AI-Enhanced Network Traffic Analysis: Leveraging Deep Learning for Real-Time Anomaly Detection and Optimization", 2020.
- [11] Lok, L. K., Hameed, V. A., & Rana, M. E. "Hybrid machine learning approach for anomaly detection", *Indonesian Journal of Electrical Engineering and Computer Science*, 27(2), 1016, 2022.
- [12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey", *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 823-839, 2012.
- [13] Sorn, V., & Vannthoeurn, S. "Using AI for Real-Time Threat Detection and Anomaly Identification", *Journal of Data Analysis and Critical Management*, 1(02), 26-33, 2025.

- [14] Nutalapati, P. "Enhancing Cybersecurity with AI-Machine Learning Techniques for Anomaly Detection and Prevention", International Journal of Engineering Research in Current Trends (IJERCT), Volume-7 Issue-1, 2019.
- [15] Garg, S., Kaur, K., Kumar, N., Kaddoum, G., Zomaya, A. Y., & Ranjan, R.. "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks", IEEE Transactions on Network and Service Management, 16(3), 924-935, 2019.
- [16] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. "Toward generating a new intrusion detection dataset and intrusion traffic characterization" ICISSp, I, 108-116, 2018.
- [17] Almuraikhi, A. H. "Network Security Central Monitoring System" Master's thesis, Hamad Bin Khalifa University, Qatar, 2020.
- [18] Kathare, N., Reddy, O. V., & Prabhu, V, "A comprehensive study of elasticsearch", International journal of science and research (IJSR), 2020.
- [19] Vivek Anand, Saurav Kumar, A. Neela Madheswari, "Students Results Prediction using Machine Learning Techniques", Int. J. Adv. Sci. Eng. Vol.3 No.2 325-329, 2016.
- [20] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018. (Dataset is publicly available at <https://www.unb.ca/cic/datasets/ids-2017.html>).