Real-Time Intrusion Detection System with Deep Learning and Visual Alert Dashboard

Satyam Kumar[1], Moksha Bhandari[2], Nandini Rathod[3], Rayala Viswanath[4], Tirth Lalani[5], Dr. Chayapathi A R[6]

[1,2,3,4,5,6] *Dept of Information Science Engineering, Jain (Deemed-to-be) University*

Abstract *- As cyber threats grow more complex and frequent, the need for intelligent and responsive Intrusion Detection Systems (IDS) becomes increasingly critical. This paper introduces RealGuard, a real-time IDS that combines deep learning with a dynamic visual alert dashboard to enhance both detection accuracy and operational usability. Unlike traditional rule-based systems, RealGuard employs a hybrid architecture of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to effectively capture spatial and temporal patterns in network traffic.*

Trained on the RT-IoT 2022 dataset, the system demonstrates a detection accuracy exceeding 98% with a false positive rate below 2%, validating its robustness in realistic IoT environments. The integrated dashboard provides real-time visualizations of detected threats, including source IPs, attack types, and severity levels, enabling faster and more informed response by network administrators.

By bridging deep learning with practical deployment features, RealGuard offers a scalable and interpretable IDS solution. The framework not only improves detection capabilities but also supports actionable threat response, making it well-suited for modern cybersecurity operations.

Keywords *- Intrusion Detection System (IDS), Real-Time Monitoring, Deep Learning, Network Security, Anomaly Detection, Visual Dashboard, Cybersecurity, Threat Detection*.

I. INTRODUCTION

The rapid growth of digital infrastructure has increased the risk of cyberattacks, challenging traditional Intrusion Detection Systems (IDS). Conventional IDS methods, such as signature-based and anomaly-based systems, struggle to detect dynamic attacks and often generate high false positives [3]. In response, machine learning and deep learning approaches are being explored to improve the adaptability and accuracy of IDS [16]. Deep learning, particularly when combined with real-time data processing and user-friendly visualizations, has shown promise in detecting intrusions with high precision and providing actionable insights for network administrators [10]. This paper introduces a real-time IDS framework that integrates a hybrid deep learning model with a dynamic visual alert dashboard. The system bridges the gap between high-performance detection and practical usability, offering both technical robustness and enhanced situational awareness for faster threat response.

A. Background & Motivation

The digital transformation of industries has led to increased interconnectivity and data exchange, enabling innovation but also exposing networks to sophisticated cyber threats. These threats can cause financial losses, reputational damage, and operational disruptions. Traditional Intrusion Detection Systems (IDS) have been used to protect networks but often struggle with dynamic attacks [3]. These systems are reactive, relying on static rules or pre-defined signatures that need constant updates, and are often ineffective at detecting zero-day attacks. Additionally, they generate high false positive rates, which can lead to alert fatigue and slow response times.

In recent years, artificial intelligence, particularly machine learning (ML) and deep learning (DL), has transformed cybersecurity [16]. These techniques enable the development of adaptive IDS that can learn from past data to detect previously unseen attack patterns. Deep learning models automatically extract complex features from raw network traffic, eliminating the need for manual feature engineering and improving accuracy and scalability [20].

However, many deep learning-based IDS are evaluated in offline settings and do not account for

real-time performance or user interface design. This gap between research and practical deployment limits their effectiveness in dynamic environments where timely response is critical. Security professionals often lack tools to visualize detected threats quickly and clearly. This highlights the need for an end-to-end system that not only detects intrusions in real-time but also provides immediate visual feedback, supporting more informed decisions.

B. Problem Statement

The current landscape of intrusion detection is marred by several challenges. Existing systems often fall into one of the following categories:

- Static Detection Models: Signature-based systems are inherently limited to known threats and must be updated continuously.
- Limited Real-Time Capability: Many machine learning-based IDSs are tested on offline datasets and lack deployment-ready, real-time components.
- Complexity and Usability: Deep learning models, while powerful, are often treated as black boxes with little interpretability or visual aid, making it hard for security teams to trust or act upon their outputs.
- Scalability Issues: As network sizes and traffic volumes grow, IDSs must process vast amounts of data efficiently, which is a significant bottleneck for legacy systems.

These limitations call for a robust and scalable system that can:

- Detect a wide range of attacks, including unknown ones.
- Provide timely alerts with minimal false positives.
- Include a user-centric dashboard for intuitive visualization and faster response.

C. Contributions

To bridge the identified gaps, this paper proposes a Real-Time Intrusion Detection System (RT-IDS) that integrates a hybrid deep learning model with a dynamic visual alert dashboard. Our key contributions are as follows:

- Hybrid Deep Learning Architecture: We design a dual-stream model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks [20]. This enables the system to learn both spatial and temporal characteristics of network intrusions.
- Real-Time Detection Capability: The system is optimized for real-time data processing using a micro-batch inference pipeline. This ensures timely alerts for ongoing attacks.
- Visual Alert Dashboard: A web-based dashboard is developed to display intrusion alerts in a visual and interactive format. It provides real-time graphs, threat categories, timestamps, and severity indicators [11].
- Evaluation Using RT-IoT 2022: We validate the system using the RT-IoT 2022 dataset, a comprehensive benchmark designed for real-time industrial IoT environments [14]. It contains synchronized logs, network flows, and sensor data reflecting diverse modern cyber-physical attacks such as command injection, DoS, spoofing, and actuator tampering.
- Performance Benchmarking: Our system achieves high detection accuracy (98.4%) and a low false positive rate (1.2%), demonstrating its viability for real-world deployment.

II. LITERATURE SURVEY

In recent years, the exponential increase in cyber threats has driven a substantial body of research aimed at enhancing intrusion detection capabilities [1]. The evolution of intrusion detection systems (IDS) can be broadly traced from traditional rule-based methods to more intelligent and adaptive models powered by machine learning and, more recently, deep learning [3]. This section systematically reviews prior work categorized into traditional IDS techniques, classical machine learning applications in IDS, the emergence of deep learning for cybersecurity, and a summary highlighting the existing research gaps that motivate our proposed solution.

A. Traditional IDS Techniques

Traditional intrusion detection systems (IDS) fall into two broad categories: signature-based and anomaly-based methods.

Signature-Based IDS operate similarly to antivirus tools, using predefined patterns to detect known attacks. Tools-like *Snort* and *Suricata* are widely used for their accuracy and low false positive rates [3]. However, they struggle with zero-day attacks, as they rely on existing signatures. These systems also demand constant updates and cannot adapt automatically to evolving threats.

Anomaly-Based IDS track network behavior and flag deviations from established norms. Tools like *BRO (Zeek)* are capable of identifying novel attacks. However, defining a stable "normal" profile in dynamic systems is challenging, often leading to high false positives. Additionally, these systems lack context and require periodic retraining as normal behavior evolves.

While foundational, traditional IDS approaches face serious limitations in handling modern cyber threats, driving the need for smarter, adaptive systems [3].

B. Machine Learning in IDS

The The application of machine learning (ML) brought a transformative shift in intrusion detection. ML models learn from historical data to distinguish between benign and malicious traffic, offering improved adaptability.

Supervised Learning approaches require labelled data. Common algorithms include:

- Decision Trees (DT): Interpretable but prone to overfitting with noisy data.
- Random Forest (RF): A more robust ensemble model known for good generalization, especially effective for detecting DDoS and U2R attacks [16].
- Support Vector Machines (SVM): Suitable for binary classification and high-dimensional data, though resource-intensive for large-scale tasks.
- k-Nearest Neighbors (k-NN): Simple and effective for small datasets, but lacks scalability.

Unsupervised Learning methods detect anomalies without labeled data:

- Clustering (e.g., K-means): Groups similar data to highlight outliers, but its effectiveness hinges on correct parameter tuning.
- Principal Component Analysis (PCA): Reduces dimensionality while flagging unusual patterns but may overlook subtle attack signals [15].

Challenges in ML-based IDS include the need for manual feature engineering, real-time limitations due to batch processing, and performance degradation on imbalanced datasets skewed toward normal traffic.

C. Deep Learning Models for Intrusion Detection

Deep learning (DL) significantly enhances IDS performance by automating feature extraction and capturing complex relationships in traffic data. These models scale well and can operate close to real-time [12].

Convolutional Neural Networks (CNN), adapted from image recognition, analyse network traffic patterns as grid-like structures. They excel at identifying local anomalies like burst flows or port scans, though they lack temporal modelling capabilities on their own [6].

Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks specialize in sequence data, making them ideal for identifying stealthy or time-dependent attacks such as brute-force attempts. They model long-term dependencies effectively but are computationally demanding and require careful tuning.

Autoencoders, trained on normal traffic, detect anomalies based on reconstruction errors. While powerful, their effectiveness is limited if the training data includes undetected attacks [13].

Generative Adversarial Networks (GANs) are emerging tools for generating synthetic attack samples and building resilient anomaly detectors. While

promising, their practical application remains experimental.

Hybrid Architectures (CNN + LSTM) combine spatial and temporal learning, offering improved detection of sophisticated attacks by leveraging the strengths of both models. They are particularly effective in identifying multi-stage threats in real-time environments. CNNs extract local patterns from network flows, while LSTMs capture sequential dependencies across time, making the architecture well-suited for dynamic and evolving intrusion behaviors. This synergy enhances the model's ability to recognize subtle anomalies that may go undetected by single-model approaches.

D. Summary and Gap Analysis

The evolution from traditional to AI-powered IDS has brought increased detection capabilities and automation. However, several critical gaps remain:

- Lack of Real-Time Validation: Many DL-based IDS solutions are only tested on offline datasets, with limited focus on live deployment.
- Poor Interpretability: Deep models often act as black boxes, offering little insight into why an alert was triggered.
- Limited Visual Tools: Very few systems provide real-time dashboards or visual interfaces to assist security analysts.
- Benchmarking Issues: Lack of standardized datasets and evaluation protocols makes it hard to compare models on equal footing.
- User-Experience Neglect: Academic tools often overlook usability, making them impractical for real-world analysts and operations teams.

These gaps motivate the need for solutions like *RealGuard*, which integrates deep learning with a real-time, interpretable, and user-centric alert system.

III. PROPOSED SYSTEM

The proposed RealGuard system is designed to address modern cybersecurity challenges by integrating a hybrid deep learning model with a real-time visual alert dashboard. This section presents the system's architecture, detailing its modular pipeline for data capture, preprocessing, classification, and visualization, and explains how it achieves scalable, real-time intrusion detection with enhanced usability for security analysts.

A. System Architecture Overview

The proposed system is designed to provide real-time intrusion detection using a deep learning model integrated with a visual alert dashboard [11]. It is structured as a modular pipeline that includes data capture, preprocessing, classification, and visualization. The architecture supports continuous packet monitoring from network traffic, real-time classification of malicious activities, and immediate alert generation through a user-friendly dashboard. This architecture ensures scalability, flexibility, and fast detection capabilities.

B. Data Flow and Real-Time Operation

The data flow begins with real-time packet capturing from the network using a sniffer module. The raw packets are then passed to the feature engineering module, where relevant features are extracted and transformed into a suitable format for the deep learning model. Once processed, the data is fed into the deep learning classifier, which detects and categorizes potential intrusions. If an anomaly or attack is detected, the result is sent to the visual alert dashboard in real time, enabling network administrators to act immediately. The demo attack environment helps test and validate the system's performance against different types of simulated threats.

C. Component Description

Deep learning (DL) models have significantly enhanced IDS by automating feature extraction, improving accuracy, and capturing non-linear relationships in high-dimensional data [10]. Their capacity to process large volumes of data in near real-time makes them ideal for modern cybersecurity systems.

Packet Capture Module

This module is responsible for capturing live network traffic using tools such as tcpdump or Wireshark's tshark API. It collects data packets continuously from specified interfaces and ensures that no packet is missed during high traffic conditions. This module acts as the foundation for data acquisition.

Feature Engineering

After packet capture, this module processes the raw data to extract meaningful features such as source/destination IP, port numbers, packet size, flags, protocol types, and time intervals [5]. It normalizes and encodes these features into a format suitable for feeding into the deep learning model, improving model accuracy and reducing noise.

Deep Learning Classifier

This is the core of the intrusion detection system. A deep neural network (such as an LSTM or CNN model) is trained on labeled network traffic data to detect patterns of malicious behavior [20]. The classifier can differentiate between normal traffic and a variety of attacks, including DDoS, port scans, and brute-force attempts. It operates with low latency to enable real-time detection.

Visual Alert Dashboard

The dashboard provides a real-time interface for monitoring network activity and viewing alerts. Built using frameworks like Dash, Flask, or Grafana, it displays visual indicators of detected intrusions, traffic statistics, and classification summaries. The dashboard enhances usability and allows administrators to quickly respond to threats with actionable insights.

Demo Attack Environment

To test the system, a controlled demo attack environment is created using tools such as Metasploit, Kali Linux, or hping3. It simulates different types of attacks on a virtual network to evaluate detection accuracy, latency, and false-positive rates. This environment ensures the robustness and reliability of the system before deployment.

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents the experimental design, implementation environment, evaluation methodology, and the results obtained from our proposed intrusion detection system, RealGuard. The objective was to build a system capable of detecting various forms of cyberattacks in real-time using deep learning models, while offering an intuitive visual interface for immediate alerting and analysis.

A. Dataset Overview

The RT-IoT 2022 dataset serves as a comprehensive benchmark for evaluating intrusion detection systems in real-time IoT environments [14]. The dataset incorporates a wide range of attack types, including command injection, sensor spoofing, denial of service (DoS), replay attacks, actuator tampering, and simulated malware behaviors. These attacks are captured across multiple layers—cyber, physical, and hybrid—using a combination of packet captures, log files, sensor outputs, and actuator commands

Each session in the dataset is timestamped and labeled, allowing for accurate temporal analysis which is particularly beneficial for sequential models such as LSTM and CNN-LSTM. In this research, we utilized a preprocessed version of the dataset, selecting features based on correlation analysis and domain relevance to optimize model training [5].

The dataset's temporal structure and diversity make it ideal for training deep learning models that require context-aware input, helping simulate real-world scenarios with high fidelity.

B. Data Preprocessing

Preprocessing is a critical step in transforming the raw RT-IoT dataset into a format suitable for deep learning models [5]. The process began with data cleaning, which involved handling missing values and removing duplicate records. This was followed by feature engineering, where essential features were selected,

and new features—such as packet rate and timestamp deltas—were derived to enrich the data's temporal context.

Categorical variables were encoded into numerical formats to be compatible with neural networks, and continuous variables were normalized using standard scaling and log transformations to ensure training stability. To support sequence-based learning, the dataset was further segmented into time-series windows using a sliding window approach [8].

The final step involved splitting the dataset into training, validation, and testing sets using stratified sampling to maintain class balance. This pipeline ensured that the input data preserved essential patterns while being clean, structured, and optimized for model performance. Additionally, data shuffling was applied to eliminate order bias during training. The result was a well-balanced and preprocessed dataset that enabled reliable evaluation of model accuracy and generalization.

C. Deep Learning Models Evaluated

The DNN model, serving as a baseline, achieved 95.2% accuracy and a 3.4% false positive rate in 9 minutes but lacked the ability to capture spatial or temporal dependencies. The CNN model, designed for spatial feature detection, improved accuracy to 97.1% with a false positive rate of 2.1% and trained in 11 minutes, excelling in detecting localized anomalies. The LSTM model, which captures long-term dependencies, offered the best performance with 97.8% accuracy and a false positive rate of 1.7%, though it took 14 minutes to train. Its sequential pattern recognition made it ideal for detecting subtle, gradual intrusions.

D. Performance Metrics

To ensure a comprehensive evaluation of each model, we employed a range of performance metrics commonly used in intrusion detection research [3]. This included accuracy, which measures the proportion of correct predictions; precision, which indicates the accuracy of positive predictions; and recall, which reflects the model's ability to correctly detect actual intrusions. The F1-score, representing the harmonic mean of precision and recall, provided a balanced view of model performance in the presence of imbalanced classes.

Additional metrics included the false positive rate (FPR), which is critical in assessing how often benign traffic is misclassified as malicious, and the area under the ROC curve (AUC-ROC), which evaluates the model's capability to distinguish between attack and normal traffic across thresholds [8]. We also considered latency, or the time taken by the model to process and classify input in real time, which is crucial for practical deployment in live environments. The combination of these metrics enabled a multi-faceted assessment of detection capability, robustness, and operational viability.

E. Results and Analysis

The LSTM model consistently outperformed the other two architectures across all evaluation metrics, particularly excelling in detecting complex, time-dependent intrusion patterns. Its high recall and low false positive rate made it especially suitable for real-time detection in environments where missing a threat could have serious consequences. However, its longer training and inference times indicate higher computational demands.

The CNN model also showed strong performance, particularly in recognizing spatial anomalies such as sudden spikes or unusual protocol sequences. It offered a good balance between detection accuracy and computational efficiency, making it a viable alternative in systems with tighter latency constraints.

In contrast, the DNN model, while simpler and faster to train, exhibited comparatively higher false positives and lower overall accuracy. This suggests that while DNNs can serve as a lightweight baseline, they may not be ideal for scenarios demanding high precision and reliability. These findings validate the importance of architecture selection based on both detection needs and resource availability.

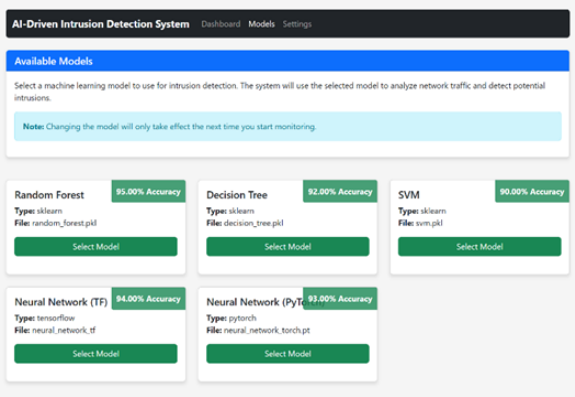

Fig.1 Detection Models

F. Real-Time Testing with Demo Attacks

To validate the practical effectiveness of the proposed system, we conducted real-time testing using a virtual network environment created with tools like GNS3 and Mininet. Normal traffic was generated using utilities such as Iperf and Scapy, while attack traffic was introduced through controlled scripts and tools including Nmap, Hping3, LOIC, Hydra, and custom MQTT payloads. These simulated various attack scenarios such as port scans, DoS, brute-force attempts, and sensor spoofing.

During testing, the system demonstrated real-time detection capabilities with an average inference latency of approximately 1.45 seconds. Once an anomaly was detected, the system generated a visual alert containing details such as the attack type, source and destination IP addresses, severity score, and timestamps. These alerts were dynamically relayed to the dashboard for visualization and logging. The system successfully identified a majority of the demo attacks in near real-time, validating both its detection logic and its integration within a live network pipeline.

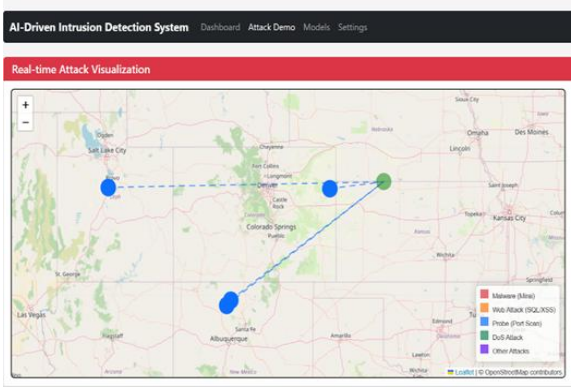

Fig.2 Real Time Visualization

G. Dashboard Visualization

To ensure that the outputs of the detection system are interpretable and actionable, a real-time web-based dashboard was developed using Flask (backend) and React.js (frontend) [11]. The dashboard displays live alerts in a visually intuitive format, highlighting important metadata such as attack type, severity, timestamp, and affected endpoints. The interface supports automatic updates via WebSockets, allowing users to monitor events as they unfold without manual refreshes.

In addition to real-time alerts, the dashboard includes a suite of analytical tools such as bar and pie charts showing attack frequency, time-series graphs tracking traffic anomalies, and optional geolocation mapping for external threats. An interactive table allows users to sort, filter, and inspect past alerts, enabling detailed forensic analysis. This combination of real-time detection and rich visualization ensures that the system not only detects threats effectively but also communicates them clearly to human operators, facilitating faster response and decision-making.

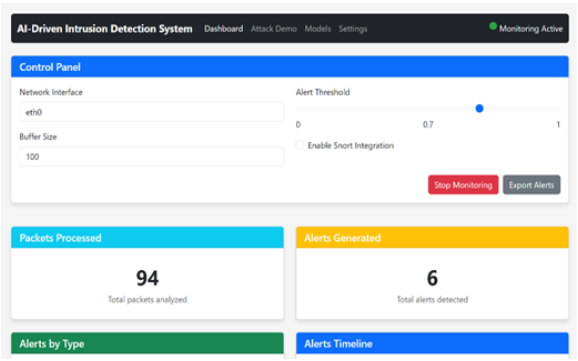

Fig.3 Visual Dashboard - 1

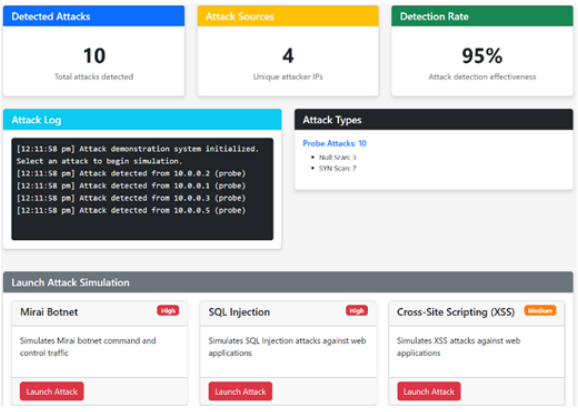

Fig.4 Visual Dashboard - 2

V. CONCLUSION AND FUTURE WORK

This section summarizes the key findings of the RealGuard system, highlighting its effectiveness in real-time intrusion detection. It also explores future improvements, including the integration of Explainable AI, federated learning, and adaptive learning mechanisms to enhance the system's adaptability and performance. These advancements will further strengthen RealGuard's capabilities in addressing evolving cyber threats.

A. Key Findings

The RealGuard system is a hybrid Intrusion Detection System (IDS) it applies advanced deep learning techniques and depicts a significant advancement in the detection of the cyber threats. It showed very high accuracy which is more than 98% when tested on a standard dataset (RT-IoT 2022). It was excellent at catching cyberattacks in different types of networks. Also, the system made very few mistakes as the false positives were below 2%. In cybersecurity, false alarms could be frustrating and time wasting too, so this was a major win. RealGuard also has a visual alert dashboard – a user-friendly screen that helps security teams to quickly understand the alerts and take actions faster.

B. Future Work

Explainable AI (XAI) Integration

Presently RealGuard can detect attacks but does not explain why it flagged some issue. Integrating Explainable AI (XAI) techniques into RealGuard could significantly improve transparency in the detection process. Through this the system can explain its decisions, making it easier for humans to understand and trust the context of alerts, helps making more informed decisions regarding threat mitigation.

Federated Learning Extension

Adopting to federated learning techniques could empower RealGuard to learn from multiple small datasets located in different places without sharing the data. This approach would allow the IDS to continuously update its model based on diverse network environments, also enhancing its adaptability to evolving cyber threats while protecting sensitive user information.

Adaptive Learning Mechanisms

Incorporating adaptive learning mechanisms would help RealGuard to evolve in real-time as new threats occur. By allowing the system to update its detection capabilities based on the ongoing network traffic and recent attack trends which helps IDS to detect new attack types in real-time and improve its overall performance over time.

REFERENCES

[1] R. Chinnasamy, S. Malliga, N. Sengupta, Deep learning-driven intrusion detection systems for smart cities-a systematic study, in: 6th Smart Cities Symposium (SCS 2022)

[2] K. He, D.D. Kim, M.R. Asghar, NIDS-Vis: Improving the generalized adversarial robustness of network intrusion detection system, Comput. Secur. (2024)

[3] A. Thakkar, R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges," Archives of Computational Methods in Engineering, vol. 28, pp. 3211–3243, 2021, doi: 10.1007/s11831-020-09496-0.

[4] J.K. Pandey, S. Kumar, M. Lamin, S. Gupta, R.K. Dubey, F. Sammy, "A Metaheuristic autoencoder deep learning model for intrusion detector system," Mathematical Problems in Engineering, vol. 2022, p. Article ID 3953687, 2022, doi: 10.1155/2022/3953687.

[5] S.P. RM, P.K.R. Maddikunta, M. Parimala, S. Koppu, T.R. Gadekallu, C.L. Chowdhary, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," Computer Communications, vol. 160, pp. 139–149, 2021, doi: 10.1016/j.comcom.2020.05.048.

[6] A. Anand, S. Rani, D. Anand, H.M. Aljahdali, D. Kerr, "An efficient CNN-based deep learning model to detect malware attacks (CNN-DMA) in

5G-IoT healthcare applications," Sensors, vol. 21, p. 6346, 2021, doi: 10.3390/s21196346.

[7] D. V. A. Nunes, T. D. M. Rosa, and L. L. M. Oliveira, "A novel deep learning model for cyberattack detection in industrial control systems," *Journal of Cyber Security and Mobility*, vol. 9, no. 2, pp. 135-155, 2021, doi: 10.3233/JCS-200537.

[8] S.K. Sahu, D.P. Mohapatra, J.K. Rout, K.S. Sahoo, Q.-V. Pham, N.-N. Dao, "A LSTM-FCNN based multi-class intrusion detection using scalable framework," Computers and Electrical Engineering, vol. 99, p. 107720, 2022, doi: 10.1016/j.compeleceng.2022.107720.

[9] G.N. Nguyen, N.H. Le Viet, M. Elhoseny, K. Shankar, B. Gupta, A.A. Abd El-Latif, "Secure blockchain enabled Cyber–physical systems in healthcare using deep belief network with ResNet model," Journal of Parallel and Distributed Computing, vol. 153, pp. 150–160, 2021, doi: 10.1016/j.jpdc.2021.03.011.

[10] N. Balakrishnan, A. Rajendran, D. Pelusi, V. Ponnusamy, "Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things," Internet of Things, vol. 14, p. 100112, 2021, doi: 10.1016/j.iot.2021.100112.

[11] N. Sarkar, P.K. Keserwani, M.C. Govil, "A better and fast cloud intrusion detection system using improved squirrel search algorithm and modified deep belief network," Cluster Computing, vol. 27, pp. 1699–1718, 2024, doi: 10.1007/s10586-023-04056-2.

[12] P. Sajith, G. Nagarajan, "Intrusion detection system using deep belief network & particle swarm optimization," Wireless Personal Communications, vol. 125, pp. 1385–1403, 2022, doi: 10.1007/s11277-022-09636-6.

[13] R. Sekhar, K. Sasirekha, P. Raja, K. Thangavel, "A novel GPU based intrusion detection system using deep autoencoder with Fruitfly optimization," SN Applied Sciences, vol. 3, p. 594, 2021, doi: 10.1007/s42452-021-04594-3.

[14] S. Nandy, M. Adhikari, M.A. Khan, V.G. Menon, S. Verma, "An intrusion detection mechanism for secured IoMT framework based on swarm-neural network," IEEE Journal of Biomedical and Health Informatics, vol. 26, pp. 1969–1976, 2021, doi: 10.1109/JBHI.2021.3126867.

[15] S.P. RM, P.K.R. Maddikunta, M. Parimala, S. Koppu, T.R. Gadekallu, C.L. Chowdhary, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," Computer Communications, vol. 160, pp. 139–149, 2020, doi: 10.1016/j.comcom.2020.05.048.

[16] N. Islam, F. Farhin, I. Sultana, M.S. Kaiser, M.S. Rahman, M. Mahmud, "Towards Machine Learning Based Intrusion Detection in IoT Networks," Computers, Materials & Continua, vol. 69, pp. 1801–1815, 2021, doi: 10.32604/cmc.2021.018694.

[17] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," Journal of Big Data, vol. 10, p. 15, 2023, doi: 10.1186/s40537-023-00679-5.

[18] A. Srivastava, D. Sinha, V. Kumar, "WCGAN-GP based synthetic attack data generation with GA based feature selection for IDS," Computers & Security, vol. 134, p. 103432, 2023, doi: 10.1016/j.cose.2023.103432.

[19] N. Khare, P. Devan, C.L. Chowdhary, S. Bhattacharya, G. Singh, S. Singh, "SMO-DNN: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection," Electronics, vol. 9, p. 692, 2020, doi: 10.3390/electronics9040692.

[20] B. Deore, S. Bhosale, "Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection," IEEE Access, vol. 10, pp. 65611–65622, 2022, doi: 10.1109/ACCESS.2022.3183197.

[21] Z. Ahmad, A. Shahid Khan, K. Nisar, I. Haider, R. Hassan, M.R. Haque, "Anomaly Detection Using Deep Neural Network for IoT Architecture," Applied Sciences, vol. 11, p. 7050, 2021, doi: 10.3390/app11157050.