iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82351

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

An Advanced DL Architecture for Furious Network Security

Mr. G MD Aslam[1], Dr. SDN Hayath Ali[2]
[1]*PG Student, Department of MCA, Ballari Institute of Technology & Management, Ballari*
[2]*Associate Professor, Department of MCA, Ballari Institute of Technology & Management, Ballari*

Abstract: ***The rapid expansion of interconnected devices and the escalating The sophistication of cyberattacks has have created an urgent necessary for flexible security solutions that surpass the capabilities of traditional, rule-based systems. This paper presents an Intelligent connection Intrusion detectable Framework powered by an improved CNN designed to identify and classify dangerous activities with more precision. The suggested technique automatically extracts hierarchical characteristics of raw data by utilizing CNNs' strength in pattern recognition network traffic, enabling the detection of both cataloged threats and previously unseen zero-day exploits. The methodology integrates critical architectural refinements, including residual connections to maintain gradient stability and important mechs to prioritize discriminative traffic patterns. To tell the challenges of class imbalance common in security datasets, the framework utilizes synthetic oversampling to ensure accurate recognition of rare but critical attack vectors. Furthermore, the system incorporates a Decision Tree or Support Vector Machine (SVM) layer to provide an interpretable classification of threats, ranging from Denial-of-Service (DoS) floods to unauthorized access attempts. Experimental analysis conducted on benchmark datasets, such as NSL-KDD and CICIDS2017, demonstrates that the framework achieves superior detection rates and lower false-alarm ratios compared to classical machine learning models. The modular design supports real-time inference and is optimized for deployment on resource-constrained edge devices, ensuring low-latency responses and data privacy. Ultimately, this research provides a scalable and robust solution for safeguarding modern network infrastructures against a dynamic and evolving threat landscape.***

I. INTRODUCTION

The current state of global connectivity is characterized by a rapid expansion of cloud services, interconnected devices, and persistent network environments, which has inadvertently provided a massive landscape for malicious actors. Conventional intrusion detection systems (IDS) typically rely on signature-based patterns or static rules to identify threats. However, these traditional methods frequently fail to detect zero-day exploits or subtle anomalies in network behavior because they are dependent on previously documented attack signatures. As modern network traffic generates immense data streams containing a complex mix of legitimate and malicious events, there is a critical need for a detection framework that can autonomously identify meaningful features without relying on human-crafted heuristics. To address these vulnerabilities, this framework utilizes an improved (CNN) to enhance the precision of unauthorized access identification. By leveraging the pattern-recognition capabilities of On going directly from raw input data—such as network traffic flows and sensor readings—rather than depending on predefined signatures. The architectural improvements, including deeper regularized layers and attention mechanisms, allow the model to concentrate on the more discriminative features while mitigating technical issues like gradient vanishing. This adaptive approach enables the system to recognize both known and previously unseen intrusions by learning the statistical relationships and temporal behaviors present in various traffic scenarios. The motivation behind this research is driven by the fact that modern security environments face escalating challenges as networks grow in both scale and complexity. These models are often task to maintain pace with sophisticated attack patterns that vary in posture, speed, or digital tactics. The proposed framework aims to fulfill several key objectives, including accurate intrusion detection, fine-grained attack classification, and the facilitation of real-time automated responses. Furthermore, by designing the system for edge deployment, the framework ensures privacy-preserving local processing, which reduces dependency on cloud infrastructure and protects sensitive payload data. Ultimately, the outcome of this task include a complete end-to-end network data pipeline, the introduction of lightweight yet powerful CNN models optimized for resource-constrained hardware, and an adaptive management framework for dynamic updates. By integrating deep feature extraction with a scalable architecture, this framework provides a robust solution for safeguarding critical infrastructures against a constantly changing threat landscape. A remainder of this report is framed to cover the literature survey, system requirements, architectural design, implementation details, and a extensive corrections of results.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

The development of this framework is further justified by the critical need for high-throughput, low-latency cyber solutions which can run at the network edge. Traditional architectures normally experience drastic degradation when subjected to the massive data volumes characteristic of modern enterprise environments. By incorporating lightweight structural refinements—such as depthwise separable convolutions and model quantization—this system achieves a minimise between greater detection rightness and minimal computational overhead. This philosophical design not only facilitates real-time packet inspection but also enables deployment on resource-constrained hardware like industrial gateways or embedded devices, ensuring that threat mitigation can occur at the point of entry without relying on external cloud processing.

Beyond the technical detection capabilities, the framework addresses the operational challenges faced by security analysts through the combining of interpretability and automated management layers. Where these DL models are continously criticized for their "black-box" nature, this system utilizes a traditional new approach that combines CNN-based feature learning with structured classification methods like decision trees to provide human-friendly logics for every flagged event. This transparency is complemented by an adaptive deployment framework that supports dynamic weight updates and adjustable sensitivity thresholds via RESTful APIs. Consequently, the system acts as an extensible security asset that can be continuously fine-tuned to counter emerging attack vectors while maintaining a manageable false-positive rate in complex, high-volume network.

II. PROPOSED FRAMEWORK

A. System Overview

The System Overview of the proposed framework defines a multi-layered, intelligent architecture designed to fill the gap bw raw network activity and actionable security intelligence. At its final, the system utilizes an improved CNN that functions as an automated feature extractor, replacing the importance for traditional, hand-crafted heuristics with deep hierarchical learning. The architecture is to support a complete end-to-end data pipeline, beginning with the acquisition of live traffic flows and packet captures from various network sources like routers and honeypots. This raw data is then transformed into structured tensors or "traffic images" that allow the CNN to identify complex spatial and temporal relationships within the communication streams.

B. Key Functional Modules

1) Data Acquisition Module

The Data Acquisition Module serves as the foundational interface between the framework and the external environment, responsible for the continuous collection of raw network data. It is designed to gather diverse information streams, including network traffic logs, system events, and packet captures (PCAP) sourced directly from routers, firewalls, servers, and strategically placed honeypots. By ingesting data from multiple vantage points, the module ensures that the subsequent layers have access to a comprehensive dataset representing both legitimate operations and a wide variety of malicious behaviors.

2) Data Cleaning and Preparation Module

The Data Cleaning and Preparation Module is a vital stage in the framework where raw, heterogeneous network logs are transformed into a standardized format compatible with the convolutional neural network. This process begins by addressing data quality issues such as missing, inconsistent, or redundant entries, which are rectified through imputation or removal to check if it doesn’t learn from erroneous signals. Because raw network data often contains categorical variables—such as protocol types (TCP, UDP, ICMP), service labels, and status flags—this module encodes them into numerical embeddings or one-hot vectors to facilitate mathematical processing by the deep learning layers. Continuous features, including packet sizes and connection durations, are scaled to uniform ranges to prevent high-magnitude values from disproportionately influencing the model's weight updates.

3) Exploratory Analysis Module

The Exploratory Analysis Module serves as the investigative core of the framework, where the system identifies underlying structures and statistical relationships within the prepared dataset before the formal training of the CNN. This module employs various data-mining techniques, such as clustering and attribute selection measures, to evaluate how effectively different traffic characteristics—like connection duration, byte counts, and request frequency—distinguish between normal and malicious behavior. By utilizing metrics such as Information Gain or the Gini Index, the module identifies the most informative attributes, which reduces noise and computational complexity while simultaneously improving the precision of the detection engine.

4) Predictive Analytics Module

The Predictive Analytics Module functions as the primary inference engine of the framework, leveraging the learned weights of the improved Convolutional Neural Network to classify live network events in real-time. This module processes the structured tensors generated by the previous stages, utilizing pooling layers to extract high-level hierarchical representations that distinguish between benign traffic and sophisticated attack patterns. By analyzing spatial and temporal relations inside the data, the system can predict the likely of various threat categories, such as denial-of-service floods, unauthorized access probes, or malware propagation. The predictive capability is enhanced through architectural refinements like residual connections and attention mechanisms, which allow the way to maintain correctness even when faced with previously unseen or zero-day exploits.

5) Reporting and Visualization Module

The Reporting and Visualization Module acts as the final interface between the intelligent framework and the security administrator, converting complex detection data into actionable insights. The primary function of this module is to provide real-time alerting through a low-latency notification layer that triggers alarms via configurable channels such as local dashboards, email, or text messages. To show the challenge of alert fatigue, the module incorporates an alarm-clustering component that groups related events using techniques like DBSCAN or hierarchical agglomeration. This clustering ensures that hundreds of low-level alerts regarding a single scanning host are collapsed into a single, meaningful incident report, significantly reducing noise and streamlining the investigation process6.User Interaction Module

The user interaction module provides a interface for accessing the system's functionalities. search for movie details, view analytical reports, and obtain prediction results through an easy-to-use platform

6) Visual Overview

The Visual Overview of the framework serves as a conceptual blueprint that illustrates the flow of data through the intelligent detection pipeline, from raw ingestion to the final administrative response. This architectural mapping highlights the five-layered approach—Data Acquisition, Preparation, Exploratory Analysis, Predictive Analytics, and Reporting—that enables the system to handle the high velocity and volume of modern network traffic. By visualizing these components, developers and stakeholders can identify the precise points where raw packet headers are transformed into structured tensors and eventually into classified attack labels.

To ensure clarity and maintainability, the framework utilizes standardized modeling tools such as Data Flow Diagrams (DFD) and Unified Modeling Language (UML). These visual aids provide the following technical perspectives:

- Data Flow Analysis: Graphical representations (bubble charts) map the movement of data between processes and stores, identifying potential bottlenecks or redundancies in the packet inspection path.

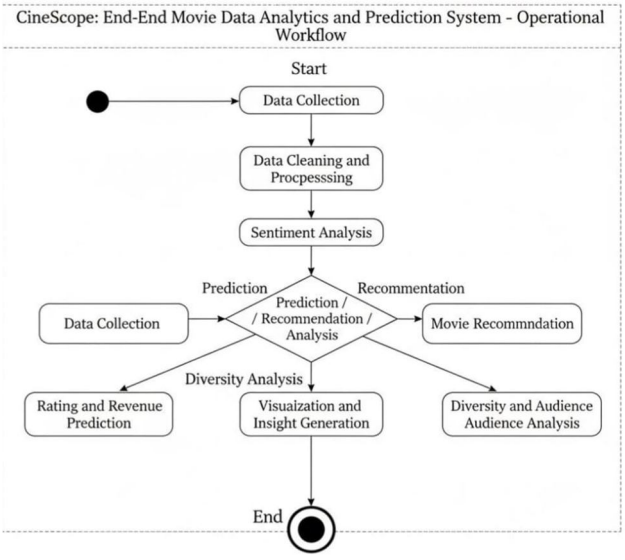


Fig1-System overflow

III. METHODOLOGY AND IMPLEMENTATION

The framework is executed through a series of specialized modules that manage the lifecycle of an intrusion detection event:

1) Data Collection Module: This foundational unit gathers network traffic logs, system events, and packet captures from various sources such as routers, firewalls, and honeypots. It ensures that both normal and malicious activities are represented to provide a balanced learning environment for the CNN.
2) Data Preparation Module: This module handles the cleaning and transformation of raw data. It addresses missing or inconsistent entries through imputation or removal and eliminates redundant records to prevent model bias.

Feature Engineering Unit: Categorical features, such as protocol types or service flags, are encoded into numerical formats. To optimize the CNN's performance, tabular data is reshaped into one-dimensional vectors or two-dimensional "traffic images" to capture complex correlations between attributes. Model Selection and Training: An improved CNN architecture is chosen to speed. network is trained using labeled datasets, such as NSL-KDD or CICIDS2017, where it adjusts millions of parameters to distinguish between benign traffic and various attack categories.

Predictive Analysis and Classification: The trained model performs real-time inference on incoming data streams. It utilizes a Decision Tree classification algorithm to partition the feature space based on the most informative attributes, providing a final label and human-readable decision rules for each event. 5.2 Technical Implementation StrategyThe implementation emphasizes Edge Deployment, enabling the system to run on on-premise hardware like industrial gateways. By utilizing model quantization and pruning, the CNN can execute high-throughput inference with limited CPU or memory resources. This strategy ensures that packet inspection and event correlation occur locally, protecting sensitive payload data and reducing dependency on external cloud infrastructure.

IV. RESULTS AND DISCUSSIONS

The evaluation of the and malicious traffic. The goal of the experimental analysis is to determine how well the system identifies various attack types while keeping the false-alarm ratio under control. To is conducted to validate its reliability and efficiency in identifying dangerous activities within network traffic. The analysis is based on standard quantitative metrics that capture the system's abilityquantitative metrics that capture the system's ability to differentiate between normal and malicious traffic. The experimental analysis focuses on measuring how accurately the system classifies diverse attack types while maintaining a manageable false-alarm ratio. To assess the effectiveness of the improved CNN, the follows key indicators are utilized: Detection Rate (Recall):

This metric represents the percent of actual attacks that system correctly identifies. Accuracy: This provides an overall measures of correctness, reflecting the proportion of correctly classified for both normal and attack flows. Precision: This says the fraction of generated alerts that correspond to true attacks, which is critical for reducing operational noise. F1-Score: This serves as the mean of precision and remember, offering a balanced measure of the system's performance. False Alarm Rate (FAR): This measures the percentage of legitimate traffic that is incorrectly flagged as an intrusion.

A. Comparative Analysis

The experiment answer says that the improved CNN-based framework consistently outperforms traditional machine learn approaches such as (SVM), Decision Trees, and K-Nearest Neighbors (KNN). While classical methods are effective for known defends, they often struggle with zero-day threats and complex, evolving patterns because they rely heavily on manual feature engineering. In contrast, the improved CNN autonomously extracts hierarchical features, capturing subtle spatial and temporal correlations that traditional models miss.

B. Discussion on Scalability and Robustness.

The answer says that the framework maintains stable detection accuracy even when subjected to high-volume traffic or adversarial environments. The inclusion of architectural refinements, such as residual connections and attention mechanisms, ensures that the model remains robust against gradient vanishing and focuses on the most discriminative traffic features. Furthermore, the system's ability to run on edge devices—achieved through model quantization and pruning—proves that it can deliver high-throughput inference with minimal computational latency. This balance of high-precision detection and low-latency response makes the framework a scalable and adaptive security solution for modern network infrastructures.

C. Interpretability and Root Cause Analysis

The integration of a hybrid classification layer, specifically utilizing decision trees alongside the CNN, significantly enhances the system's forensic utility. No matter standard deep learning stepsthat function as "black boxes," this architecture provides human-readable decision rules for every flagged event. This transparency allows security analysts to trace the logical path from the CNN's feature embeddings to the last attack label, facilitating faster root cause analysis and a clear understanding of the specific traffic characteristics that triggered an alarm.

D. Alert Management and Noise Reduction

The results highlight the effectiveness of the alarm-clustering module in mitigating the challenge of alert fatigue. By grouping related alerts based on source/destination similarity and temporal proximity, the system successfully collapses hundreds of redundant low-level logs into a single, cohesive incident report. This process not only streamlines the investigation workflow but also provides analysts with critical incident context, allowing them to distinguish between isolated events and larger, coordinated attack campaigns.

E. Efficiency in Resource-Constrained

Environments

Experimental observations confirm that the framework is highly effective for edge deployment on hardware like Raspberry Pi or industrial gateways. By employing model quantization and depthwise-separable convolutions, the framework reduces memory consumption and computational overhead without sacrificing detection accuracy. This ensures that high-throughput packet inspection can occur locally and in real-time, providing immediate protection at the network perimeter while ensuring that sensitive data remains encrypted and on-premise. In conclusion, the proposed CineScope framework effectively converts unprocessed movie business intelligence. The findings highlight how the integration predictive modeling can support smarter and more informed decision-making within the entertainment industry.

V. ADVANTAGES

1) Autonomous Feature Extraction: The system utilizes an improved CNN to representations directly from raw traffic flows or packet payloads. This eliminates the necessary for manual, handcrafted feature engineering required by traditional machine learning methods.
2) Detection of Novel Threats: Unlike signature-based systems that rely on predefined patterns, this framework can identify both known attacks and previously unseen zero-day exploits by learning statistical relationships in network activity.
3) Architectural Robustness: The inclusion of residual connections and regularized layers prevents gradient vanishing and ensures stable training for deeper networks.
4) High Precision in Imbalanced Environments: The design integrates oversampling and synthetic minority generation (SMOTE) to ensure that rare but critical attack vectors are not overlooked in datasets dominated by normal traffic. Efficiency for Edge
5) Deployment: The model can be pruned or quantized for deployment on resource-constrained devices like Raspberry Pi or industrial gateways. This enables high-throughput, real-time inference with minimal power consumption.
6) Privacy and Latency Reduction: By processing data locally on-premise, the system reduces dependency on cloud infrastructure, protects sensitive payload data, and ensures near real-time response times. Enhanced
7) Interpretability: The integration of Decision Trees provides a clear, hierarchical structure of decision rules, allowing security analysts to understand the logic behind each flagged intrusion.
8) Reduced Alert Fatigue: The framework uses clustering techniques (such as DBSCAN) to group related alarms into meaningful clusters, allowing analysts to quickly identify root causes rather than triaging individual alerts.
9) Scalability and Extensibility: The modular software stack allows for the addition of new sensors, traffic types, or threat-intelligence feeds without requiring a major redesign.

VI. LIMITATIONS

1) Dataset Imbalance Sensitivity: While the system uses SMOTE and oversampling, intrusion data is naturally highly imbalanced, with far fewer attacks than normal events. If not perfectly managed, rare but critical threats could still be overlooked.
2) Computational Overhead on Deeper Layers: The "improvement" involves deeper and more complex architectures, such as residual connections and attention mechanisms. While powerful, these increase the number of parameters to millions, requiring careful optimization (like pruning or quantization) to avoid high memory and computation costs during training and deployment.

3) Dependency on Labeled Data: This follows a supervised learning approach, meaning its accuracy is heavily dependent on the quality and availability of labeled datasets of normal and attack scenarios.
4) Complexity in Capture and Preprocessing: Converting raw network packets into 2D representations like traffic matrices or spectrograms adds a layer of preprocessing complexity compared to simpler models.
5) Adversarial Vulnerability: Security models can be susceptible to "adversarial training" attempts where attackers craft specific inputs to fool the model.
6) Environmental Noise: Standard architectures can sometimes fail to capture fine-grained details amidst environmental noise or obfuscation attempts by attackers.
7) Maintenance of Training Sets: The system requires a continuous feedback loop where alerts are reviewed and labeled by humans to update the training set, which can be resource-intensive for security engineers.
8) Performance Stability: While robust, maintaining stable detection accuracy in extremely high-volume or highly adversarial environments remains a constant technical challenge.

VII. CONCLUSION

The integration is an improved CNN into an intrusion detection system provides a powerful and adaptive approach to securing networks against a wide range of cyber attacks. By leveraging the CNN's ability to automatically extract complex, high-level features from raw network traffic, the system effectively detects both known and previously unseen attacks, overcoming the significant limitations of traditional signature-based IDS solutions. The enhanced architecture—incorporating technical refinements such as residual connections, attention mechanisms, and optimized convolutional layers—improves overall detection accuracy while maintaining the efficient computational performance required for real-time deployment. When combined with modular components like alert clustering and decision-making layers, the framework does more than just identify malicious activity; it organizes and prioritizes alerts to enable security analysts to perform faster and more effective root cause analysis. Ultimately, this CNN-based IDS demonstrates a scalable, robust, and intelligent solution for modern network security. It offers privacy-preserving monitoring through edge deployment, adapts to evolving attack patterns, and integrates seamlessly into existing infrastructure to provide a proactive and sophisticated defense against a constantly changing threat landscape.

VIII. FUTURE WORK

The future work for the INID Framework focuses on increasing the system's adaptability and depth of analysis to stay ahead of an ever-changing threat landscape. These planned advancements aim to refine the balance between high-speed performance and forensic transparency.

A. Technical and Architectural Advancements

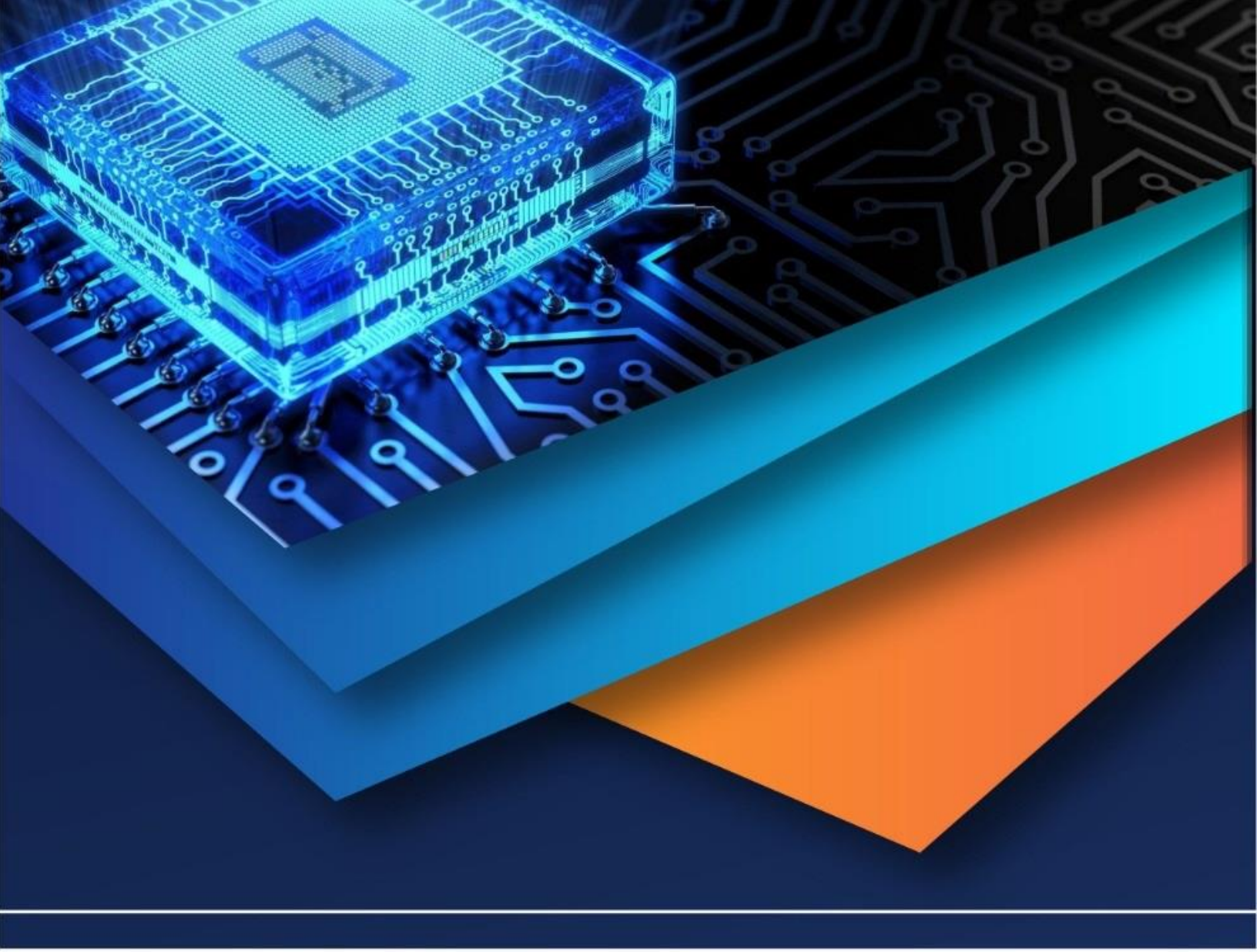
Hybrid Sequence Modeling: Integrating CNNs with recurrent architectures, such as LSTM or Gated Recurrent Units (GRU), is a key objective to better captures temporal dependencies in sequential network data. This would significantly improve the detection of "slow and stealthy" attacks that evolve over long durations.
Streaming Data Integration: Future versions will focus on deep integration with streaming data analysis tools, allowing the framework to process live, high-velocity traffic with near-zero latency.
Online Learning and Adaptive Retraining: To reduce the need for manual intervention, the system will incorporate online learning mechanisms that automatically update model weights as new attack vectors emerge.
Multi-Source Data Fusion: Expanding the input scope to combine network traffic with host-based logs and real-time threat intelligence feeds will provide a more correct view of the security environment.

B. Operational and Deployment Enhancements

Sayable AI (XAI) Visualization: Developing advanced visualization tools and feature attribution methods will help security analysts understand precisely why a CNN triggered an alert, fostering greater trust in automated decisions.
Lightweight Edge Optimization: Further research into specialized lightweight CNN architectures will expand the system's applicability to Internet of Things (IoT) networks and other highly distributed, resource-constrained environments.
Adversarial Defense Mechanisms: Incorporating adversarial training and ensemble methods will be critical to safeguarding the model against attackers who attempt to fool the system with specifically crafted inputs.
Automated Feedback Loops: Enhancing the current feedback mechanism will allow the system to more efficiently ingest human-labeled results from alert reviews, continuously refining the accuracy of the training set.

REFERENCES

[1] S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Pearson, 2009.
[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[3] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the NSL-KDD Data Set," IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.
[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Data Set for Intelligent Intrusion Detection," Proceedings of the 15th Annual Conference on Privacy, Security and Trust (PST), 2017.
[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, 2002.
[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, 2015.
[7] F. Chollet, "Deep Learning with Python," Manning Publications, 2017.
[8] Flask Documentation, "The Python Microframework for Web Development," https://flask.palletsprojects.com
[9] Scikit-Learn Documentation, "Decision Trees and Support Vector Machines," https://scikit-learn.org
[10] Object Management Group (OMG), "Unified Modeling Language (UML) Specification," Version 2.5.1, 2017.
[11] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, 2001.
[12] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers Inc., 1993.
[13] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN), 1996.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY