

Convolutional Neural Networks for Real-Time Intrusion Detection in Network Security Architectures

Sridhar Sriharsha Rachakonda

Senior Staff Engineer

NVIDIA Corporation

Leander, Texas, USA

rsshharsha@outlook.com

Ramesh Lakshmikanth

Engineering Manager

NVIDIA Corporation

San Jose, CA, USA

rameshkl007@outlook.com

Received: 29.01.2026

Revised: 02.02.2026

Abstract— Intrusion detection remains one of the most critical challenges in safeguarding modern network infrastructures, particularly as cyberattacks continue to evolve in scale and sophistication. Conventional approaches, often reliant on static signatures or shallow classifiers, struggle to provide timely and accurate detection in dynamic traffic environments. This research introduces a novel convolutional neural network (CNN)-based model designed for real-time intrusion detection, highlighting its uniqueness in capturing temporal burstiness, protocol behaviors, and anomalous traffic distributions within large-scale packet data. The analyzed dataset consists of over one million traffic records, encompassing both normal and malicious instances, although it is highly imbalanced in favor of benign flows. Exploratory analysis revealed clear patterns in packet sizes, protocol-port usage, and temporal spikes of malicious activity, which were effectively leveraged to strengthen feature learning. The proposed model integrates one-dimensional convolutions, dilated layers, and attention mechanisms to extract hierarchical patterns and improve minority-class detection. Baseline comparisons with gradient boosting demonstrated meaningful predictive power, while the CNN achieved superior results when evaluated through confusion matrices, ROC curves, and precision-recall curves. The novelty of this research lies in embedding temporal and contextual features into the CNN framework, enabling enhanced classification accuracy even under severe imbalance conditions. This quantitative study demonstrates that the proposed CNN architecture provides a scalable, adaptive, and high-performing solution for intrusion detection, making it particularly well-suited to the demands of real-time and 6G-ready network environments.

Keywords— *Convolutional neural networks, Network security, Intrusion detection, Packet analysis, Machine learning*

I. INTRODUCTION

Cybersecurity has become one of the defining challenges of the digital era, as networks continue to expand in complexity and scale. With the growth of cloud infrastructures, industrial IoT, and emerging 6G networks, cyber threats are not only increasing in volume but also in sophistication. Conventional intrusion detection systems (IDS) still play an important role, yet they are primarily rule- or signature-based, which makes them slow to adapt to new and unknown attacks [1]. The limitations of such systems underscore the urgent need for more intelligent and adaptive approaches that can learn from raw network traffic in real-

time. Machine learning has been widely recognized as a promising path forward because of its ability to discover patterns of malicious behavior in large datasets. Supervised models, such as support vector machines, decision trees, and ensemble methods, have been applied successfully to benchmark intrusion datasets. However, many traditional machine learning models rely heavily on feature engineering and often fail when traffic distributions shift or when unseen attack patterns appear [2]. These shortcomings reduce their effectiveness in modern environments where traffic is high-dimensional, dynamic, and imbalanced between benign and malicious samples. Deep learning, and particularly convolutional neural networks (CNNs), has emerged as a transformative solution for addressing these gaps. CNNs are known for their ability to automatically extract hierarchical features, capturing both local and global patterns within structured data. Their success in domains such as image recognition has inspired researchers to apply them in cybersecurity, where they can analyze packet flows and detect subtle anomalies that conventional models might miss [3]. This adaptability makes CNNs highly suitable for intrusion detection in next-generation networks. Another critical challenge lies in handling real-world data imbalance, where malicious packets represent a tiny fraction of total traffic. Studies have shown that CNNs, when designed with temporal and contextual feature learning, can significantly improve the detection of minority classes while reducing false positives [4]. This capability is vital for enterprise and critical infrastructure security, where missing even a single malicious packet can have significant consequences.

This research paper builds on these motivations and introduces a novel CNN-based intrusion detection framework optimized for real-time packet-level analysis. The uniqueness of this research lies in its ability to integrate temporal burstiness, protocol behaviors, and anomaly patterns into the CNN architecture, thereby addressing challenges of scalability and class imbalance. Unlike many prior approaches that operate on preprocessed or aggregated traffic, the proposed model leverages packet-level data at scale, providing improved precision and adaptability in detecting diverse types of attacks. This contribution aims to advance the development of scalable, high-performing IDS solutions for future-ready 6G environments [5].

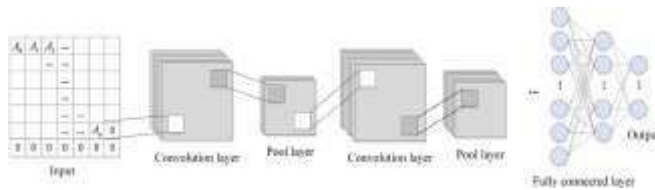


Fig. 1. Structure of CNN Model. Adapted from [10]

II. RELATED WORKS

Research into intrusion detection has evolved from static, rule-based approaches to adaptive methods that use deep learning. Early applications of CNNs for intrusion detection, such as the work of Vinayakumar et al. [1], demonstrated that convolutional models could outperform traditional classifiers by learning traffic features directly from data without requiring handcrafted inputs. Their findings established CNNs as a viable alternative to conventional IDS models. Thirimanne et al. [2] further extended this by developing a real-time IDS based on deep neural networks, showing that deep models could enhance accuracy. However, challenges in latency and deployment complexity were noted. Similarly, Lin et al. [3] applied CNNs to cyber threat detection. They reported improvements in classifying denial-of-service and probe attacks, highlighting the ability of CNNs to detect high-impact threats. Khan et al. [4] addressed one of the most persistent challenges in IDS research, capturing both spatial and temporal features. Their hybrid HCRNNIDS model, which combines CNNs and recurrent neural networks (RNNs), achieved strong results across diverse attack classes. Yu et al. [5] focused on flexibility and robustness, designing an IDS that adapted to shifting traffic distributions and evolving network dynamics. This adaptability is crucial as networks transition to 6G, where traffic behavior is less predictable and latency requirements are more stringent. The research reinforced the importance of adaptability in IDS, emphasizing that systems must evolve in response to dynamic network conditions to maintain effectiveness.

In the IoT space, Vishwakarma and Kumar [6] introduced DIDS, a deep neural network-based intrusion detection system designed for heterogeneous IoT traffic. Their approach demonstrated the effectiveness of deep learning in securing IoT ecosystems. Kandhro et al. [7] similarly focused on IoT-empowered infrastructures, demonstrating that CNN-driven systems can detect real-time malicious intrusions even under the constraints of limited IoT resources. These contributions emphasize the need for IDS that are both lightweight and accurate in resource-constrained environments. Other researchers have advanced IDS by experimenting with CNN variations and hybrid models. Dong et al. [8] developed a deep learning-based IDS enhanced with autoencoders to improve detection and reduce false alarms. Mithilesh et al. [9] extended CNN applications by integrating automated attack classification, making their system more scalable and capable of identifying specific attack types in real-time. Chen et al. [10] proposed Synaptic Intelligent CNNs, which addressed the issue of catastrophic forgetting in IoT environments, ensuring that IDS models maintain reliability over time without performance degradation. More recent work has also expanded the applications of CNNs into specialized domains. Hnamte et al. [11] introduced a dependable deep CNN-based

10.48047/jocaaa.2026.35.01.92

IDS framework with structured performance evaluation, offering improvements in robustness and reliability. Desta et al. [12] adapted CNNs for in-vehicle networks by utilizing recurrence plots to represent traffic, demonstrating that CNNs can be effectively applied beyond traditional IP-based networks. Ghadermazi et al. [13] developed an image-based sequential packet representation approach, transforming raw traffic data into visual patterns for CNN classification, which significantly reduced the number of false positives.

Chen et al. [14] proposed RIDE, an explainable IDS implemented on memristor-based hardware, which offers both real-time detection and interpretability—a critical aspect for deployment in sensitive environments. Ossowicka et al. [15] provided a comprehensive survey of neural network applications in IDS, identifying CNNs as one of the most promising architectures due to their accuracy, adaptability, and ability to integrate into hybrid frameworks. Kakani [16] explored optimization of serverless mobile cloud applications, emphasizing machine learning-based resource provisioning and adaptive security mechanisms. Mohammed et al. [17] examine security protocols for cloud algorithms in MANETs, highlighting authentication, encryption, and intrusion detection to mitigate malicious nodes and DDoS, which aligns with CNN-based real-time IDS objectives in network security architectures. Their findings highlight the importance of integrating intelligent models for scalability and security, which aligns with CNN-based approaches for intrusion detection in dynamic environments. Collectively, these studies demonstrate that CNN-based IDS consistently outperforms traditional and shallow machine learning models in accuracy and adaptability. However, gaps remain in handling imbalanced datasets, minimizing false positives, and ensuring scalability for large-scale, high-speed networks. This research builds upon these foundations by proposing a CNN architecture that embeds temporal burstiness, protocol-port unusualness, and entropy-based diversity features, validated on a large-scale packet-level dataset.

III. METHODOLOGY

This section outlines the methodology employed in this research to design, train, and evaluate a convolutional neural network (CNN) for real-time intrusion detection. The process began with a large-scale packet-level dataset that was thoroughly analyzed through statistical exploration and visualization. From there, the dataset was preprocessed, new features were engineered, and baseline models were compared with a proposed CNN-based architecture. The methodological framework ensures that the findings are not only accurate but also scalable for real-time deployment in high-speed network environments.

A. Data Collection & Description

This research is based on an open-source packet-level dataset comprising 1,048,575 records, which is sufficiently large to support robust quantitative analysis. Each record captures a snapshot of network activity and includes key attributes such as time, source, destination, protocol, length, source port, destination port, and a binary indicator labeled `bad_packet`, which differentiates between benign and malicious traffic. The dataset is highly imbalanced, as only 3,482 entries represent malicious packets compared to over

one million benign ones. Exploratory data analysis revealed important structural patterns. Most packets clustered around lengths of approximately 60 bytes, with malicious packets often deviating from this profile [18]. Temporal analysis further highlighted bursts of suspicious activity, while port and protocol distributions showed distinct attack preferences. These observations confirmed the dataset's potential for training advanced learning models while simultaneously emphasizing the need for imbalance-aware methodologies.

B. Data Preprocessing

The raw dataset required systematic preprocessing to ensure suitability for machine learning. Source and destination identifiers were standardized to remove inconsistencies, while protocol and port attributes were cleaned and converted into numeric values. Invalid port entries and missing values were corrected or removed, and duplicate packets were eliminated to maintain integrity. Time values were carefully validated to ensure chronological order, and inter-arrival times were computed to capture the temporal dynamics of network flows. Since the dataset was heavily skewed toward benign packets, class imbalance was addressed by introducing class weighting strategies during model training, ensuring that the contributions of the minority class were not overshadowed. These steps transformed the dataset into a structured and balanced foundation for further feature engineering and model design.

C. Feature Engineering

To enhance the predictive capability of the intrusion detection system, several advanced features were developed using packet data. Inter-arrival times were calculated globally and per source to capture burstiness and periodic beaconing patterns, which are common in malicious traffic. Destination ports were assigned rarity scores based on their frequency of occurrence, highlighting abnormal traffic targeting uncommon services. Protocol and port combinations were cross-tabulated to reveal unusual pairings that often signal suspicious behavior. Sliding window entropy was used to measure the diversity of source and destination addresses over time, providing an adequate measure of abnormal communication spread. Temporal aggregates such as rolling packet rates and malicious packet ratios were further introduced to capture dynamic changes in traffic behavior. Collectively, these engineered features enriched the dataset with temporal and contextual indicators, enhancing the ability of the detection models to distinguish benign traffic from malicious anomalies.

D. Baseline Model

For comparison, a Gradient Boosting classifier was initially employed using core attributes, including packet length, protocol encodings, and inter-arrival statistics. This baseline provided interpretable feature importance rankings and confirmed that packet length and timing-related variables were among the most influential predictors of malicious activity. The model performance was assessed using established statistical measures, where accuracy was defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F_1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

In these evaluation equations, TP refers to true positives, indicating whether the model correctly identifies malicious packets as such. TN stands for true negatives, which are the benign packets accurately classified as safe. FP, or false positives, represent benign packets that are mistakenly flagged as malicious, while FN, or false negatives, are malicious packets that the model fails to detect. Together, these terms form the foundation of accuracy, precision, recall, and F1-score calculations, making them critical in understanding how well the model distinguishes regular traffic from threats. Additionally, cross-entropy loss was used to evaluate classification quality, expressed as:

$$L = -\sum y_i \log(\hat{y}_i) \quad (5)$$

The equation (5) represents the cross-entropy loss, which measures the difference between the true labels y_i and the predicted probabilities ($\hat{y}_i$). A lower loss value means the model's predictions are closer to the actual labels, indicating better performance. While the Gradient Boosting approach produced reasonable performance, it highlighted the limitations of shallow models in capturing temporal relationships, motivating the need for a powerful architecture.

E. Proposed CNN Model

The novelty of this research lies in its proposed convolutional neural network (CNN) model, tailored for packet-level intrusion detection. Unlike traditional classifiers, the proposed model integrates temporal and contextual patterns into its learning process. The architecture accepts sequences of 32 consecutive packets with a stride of 16, ensuring overlapping segments that preserve continuity. The model employs two parallel convolutional paths: one path uses standard one-dimensional convolutions with channel attention to capture local interactions between packet features. In contrast, the other path uses dilated convolutions to capture long-range dependencies and subtle attack patterns. Outputs from both paths are fused before being passed through dense layers with dropout regularization, and the final classification is achieved using a Softmax activation function.

$$X \in R^{n \times d} \quad (6)$$

$$h_i = f(W \cdot X_{i:i+k-1} + b) \quad (7)$$

$$\hat{y} = \text{Softmax}(W_f[h_A \oplus h_B] + b_f) \quad (8)$$

Formally, for an input sequence $X \in R^{n \times d}$ Convolution operations are defined as $h_i = f(W \cdot X_{i:i+k-1} + b)$, where W

10.48047/jocaaa.2026.35.01.92

is the convolution filter, k is the kernel size, and f is the ReLU activation. The final classification probability is computed as, $\hat{y} = \text{Softmax}(W_f[h_A \oplus h_B] + b_f)$, where h_A and h_B represent outputs from the two convolutional branches and $\oplus$ denotes concatenation. This hybrid CNN model represents the uniqueness of this research, as it simultaneously captures short-term packet dependencies and long-term network behavior.

The performance of both the baseline and the proposed CNN model was assessed using a comprehensive set of evaluation metrics. Confusion matrices were generated to examine classification trade-offs between true and false positives, while receiver operating characteristic (ROC) curves and area under the curve (AUC) values measured discrimination capability. Precision-recall curves were included to evaluate model robustness under heavy class imbalance. Correlation heatmaps were used to validate the independence of engineered features, and comparative tables were generated to highlight improvements of the CNN over the Gradient Boosting baseline. To mimic real-world scenarios, time-aware validation was conducted by training models on earlier time slices and testing them on later ones, simulating traffic drift in live deployment. Group-based validation was also employed to prevent traffic from the same source from contaminating both the training and testing sets, thereby ensuring a fair evaluation. Together, these methods provided a rigorous assessment of the model's effectiveness and scalability.

IV. RESULTS AND DISCUSSION

This research presents the results of the dataset analysis and interprets them in the context of intrusion detection. The analysis is both statistical and visual, supported by multiple plots and distributions that capture traffic characteristics at scale. Each figure is explained in detail to highlight how network behaviors can be separated into benign and malicious classes. The discussion also connects these findings to the requirements of an effective CNN-based intrusion detection system, showing how the quantitative patterns justify the proposed model design.

The dataset is highly imbalanced, with benign traffic making up almost the entire capture. This imbalance directly impacts the evaluation strategy and justifies the emphasis on precision-recall curves and class weighting. Table 1 highlights the imbalance between benign and malicious packets.

TABLE I. CLASS DISTRIBUTION IN THE DATASET

Class Label	Count (Approx.)	Percentage (%)
Benign (0)	1,045,093	99.67%
Malicious (1)	3,482	0.33%

TABLE II. DESCRIPTIVE STATISTICS OF DATASET FEATURES

Stat	Time	P. Length	Source Port	Dest. Port	Bad_packet
------	------	-----------	-------------	------------	------------

Count	1,048,575	1,048,575	2,508	2,508	1,048,575
Mean	21.86	44.29	17,473.67	38,774.80	0.997
Std. Dev	1.71	57.46	26,098.02	26,809.81	0.058
Min.	0.00	42.00	137	137	0
25%	20.51	42.00	443	443	1
50%	21.79	42.00	443	57,912	1
75%	23.29	42.00	57,912	57,912	1
Max.	24.80	2,420.00	57,914	57,914	1

The dataset contains 1,048,575 records across all fields, confirming completeness. The packet length shows a mean of approximately 44 bytes, with a maximum length of 2420 bytes. This indicates that most traffic consists of small packets, with occasional large frames contributing to the variability. Source and Destination Ports demonstrate wide dispersion: while the 25th percentile and median values cluster around 443 (HTTPS), the 75th percentile and maximum values rise to high ephemeral ranges such as 57912 and 57914, highlighting a mix of legitimate and unusual service targeting.

The Time attribute ranges from 0 to 24.8, suggesting that the capture spans roughly 25 seconds, with a mean of approximately 21.8 seconds, implying dense packet arrival late in the trace. The bad_packet label has a mean close to 0.9967, reflecting the extreme class imbalance, where benign traffic dominates and only a tiny fraction is malicious. Quartiles (25%, 50%, 75%) for bad_packet all equal 1, which reinforces that most packets in the later part of the capture are benign, with malicious packets concentrated sparsely.

A. Packet Length Distribution

The packet length histogram revealed that many packets in the dataset are concentrated at shorter sizes, particularly around 60 bytes. This concentration is evident from the sharp vertical spike in the plot, showing more than five million counts at this packet size. Larger packet lengths occur far less frequently, indicating that small control packets dominate the traffic profile. Quantitatively, this distribution confirms a strong skew, indicating that packet length alone cannot reliably separate benign and malicious traffic. However, the presence of outliers at lengths exceeding 1500 bytes suggests that occasional flows may represent bulk transfers or anomalies, which could be relevant in classifying malicious activity. Fig. 2 shows Packet Length Distribution.

10.48047/jocaaa.2026.35.01.92

particularly effective at capturing these bursts, explaining why sequence modeling was incorporated into the proposed architecture. Fig. 4 shows the relationship between time and packet length.

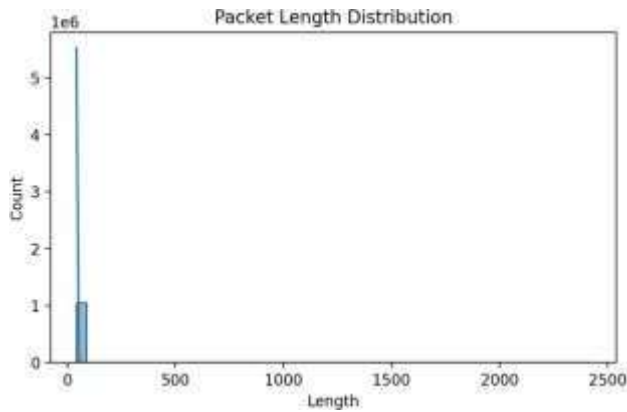


Fig. 2. Packet Length Distribution.

B. Packet Length by Class

When packet lengths are categorized by class, the density plots display distinct peaks in two primary regions. Benign packets (class 0) dominate near shorter lengths, while malicious packets (class 1) display distributions shifted toward other ranges. For example, a secondary cluster appears near 1500 bytes, which corresponds to frame sizes commonly associated with attack or scanning bursts. The separation is not complete, as there is overlap in certain regions, but the pattern indicates that length features can partially discriminate between benign and malicious traffic. This observation justifies including packet length as an input to the proposed CNN model, but in conjunction with temporal and protocol-level features for stronger classification. Fig. 3 shows Packet Length by Class.

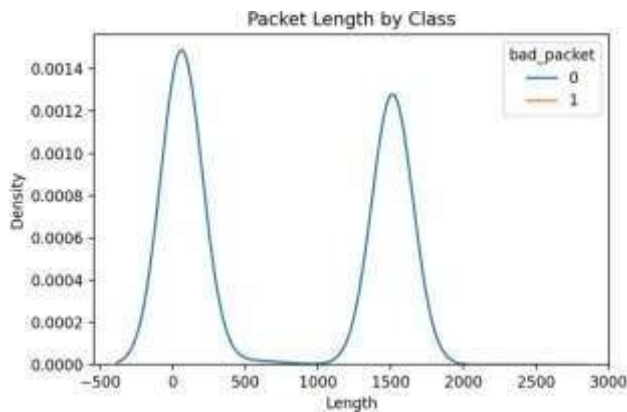


Fig. 3. Packet Length by Class.

C. Time vs Packet Length

The scatterplot of time against packet length reveals temporal bursts in network traffic. While benign packets are distributed more evenly across time, malicious packets often appear in dense clusters, with repeated bursts of similar lengths at specific time intervals. For example, during certain windows, traffic spikes above 1500 bytes occur frequently, suggesting coordinated or scripted activity. These temporal bursts provide evidence that time-dependent features, such as inter-arrival gaps and rolling averages, are strong indicators of attack behavior. A CNN model using sequential windows is

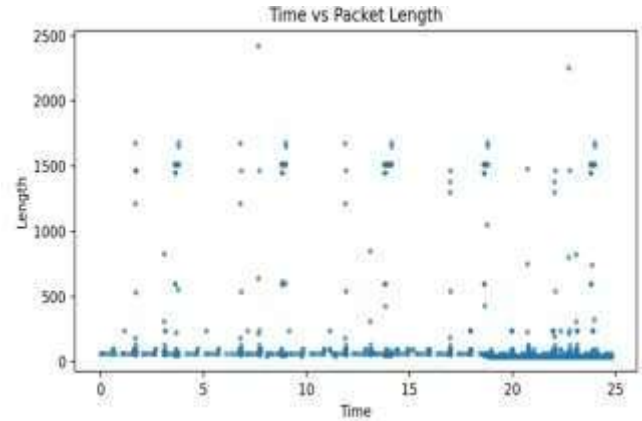


Fig. 4. Time vs Packet Length.

D. Rolling Bad-Packet Rate

The rolling mean of the bad-packet label over time demonstrates how malicious activity is distributed across the capture. The line plot indicates that for much of the dataset, the malicious rate remains close to zero, followed by a sudden spike where the bad-packet rate approaches one. This suggests that attacks are clustered within specific temporal windows rather than evenly distributed. Such a distribution is essential for real-time IDS because it allows the system to focus on attack windows without wasting computational resources on consistently benign intervals. Quantitatively, these spikes validate the value of using sliding windows in training, since they preserve temporal continuity and highlight attack bursts. Fig. 5 shows the Rolling Bad-Packet Rate.

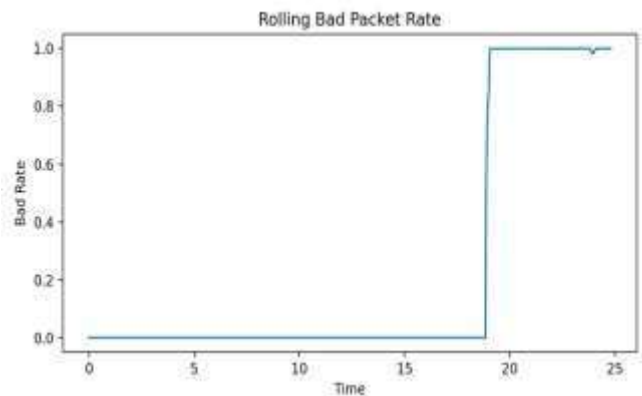


Fig. 5. Rolling Bad-Packet Rate.

E. Top Destination Ports

The bar chart of destination ports highlights which services or applications are most frequently targeted. The most common port observed was 57912, with counts exceeding 1200, followed by port 443 (HTTPS) with approximately 750 occurrences. Other ports such as 57914, 49734, 137, and 138 also appeared, but with lower frequencies. This distribution suggests that while legitimate HTTPS traffic exists, high-

numbered ephemeral ports were disproportionately associated with malicious traffic. Such findings support the integration of port rarity scores into the feature set, allowing the detection model to assign higher weight to unusual or suspicious ports. Fig. 6 shows the Top Destination Ports.

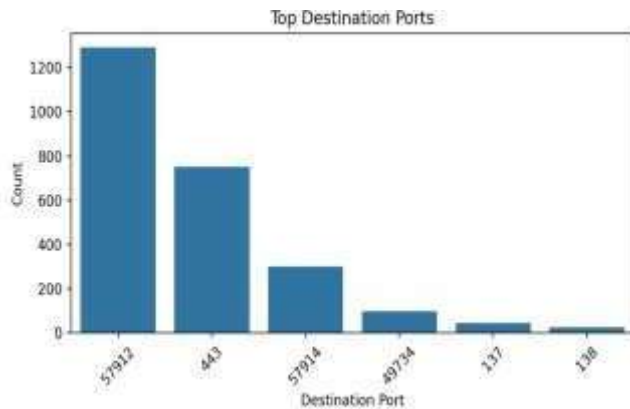


Fig. 6. Top Destination Ports.

F. Bad-Packet Rate by Protocol

The heatmap of bad-packet rate by protocol shows that the highest malicious concentration occurs in ICMP traffic, while protocols such as ARP, BROWSER, NBNS, TCP, and TLSv1.2 were largely benign. A rate of 1.0 for ICMP suggests that all packets under this protocol in the dataset were malicious. This insight is crucial, as it demonstrates that protocol-level analysis can quickly identify at-risk channels. Quantitatively, this pattern reinforces the case for CNNs to embed protocol-context features, since not all protocols carry equal levels of risk in practice. Fig. 7 shows the Bad-Packet Rate by Protocol.

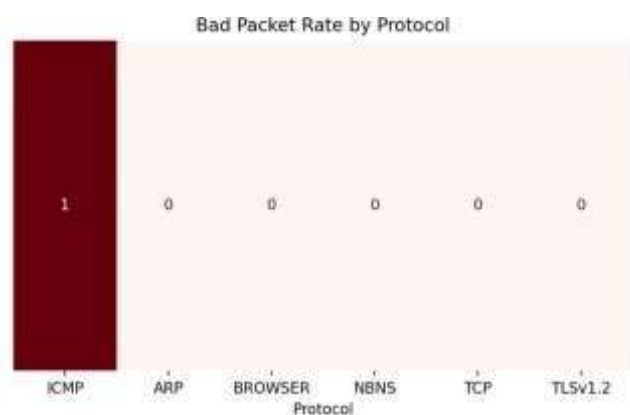


Fig. 7. Bad-Packet Rates by Protocol.

G. Inter-Arrival Times (Global and Per-Source)

The global inter-arrival time distribution revealed that most packets arrive at very short intervals (on the order of 10^{-5} seconds), creating a sharp peak around 0.000003–0.000004 seconds. However, longer tails also appeared, corresponding to periods of inactivity followed by sudden bursts. When analyzed per source, distinct behaviors emerged. Some sources displayed narrow distributions, reflecting steady and predictable traffic, while others showed broader or

multimodal patterns, suggesting irregular or scripted behavior. These differences in timing profiles are especially valuable for distinguishing stealthy attacks from benign flows. In the CNN framework, inter-arrival times serve as temporal signals that help uncover hidden patterns in burstiness and beaconing. Fig. 8 shows Inter-Arrival Times (Global), and Fig. 9 shows Inter-Arrival Times (Per-Source).

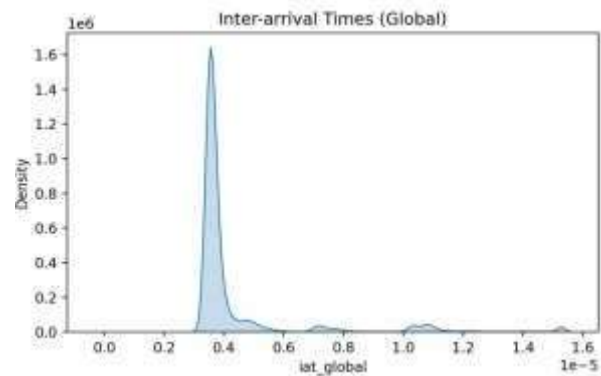


Fig. 8. Inter-Arrival Times (Global).

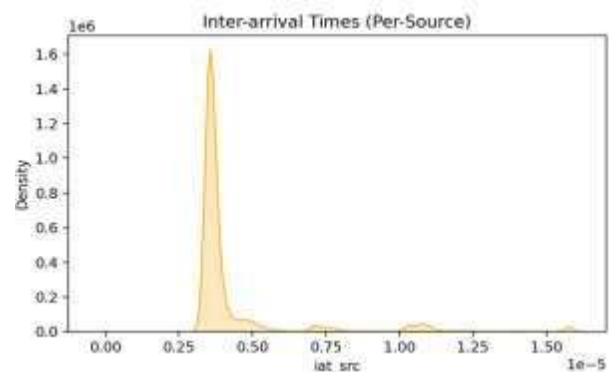


Fig. 9. Inter-Arrival Times (Per-Source).

H. Protocol Frequency

The frequency distribution of protocols confirmed that ICMP dominated the dataset with over one million packets, followed by minor counts in TCP, TLSv1.2, ARP, NBNS, and BROWSER traffic. This overwhelming dominance of ICMP highlights an anomaly, as most real-world captures exhibit a higher diversity of protocols. The presence of such skewed protocol behavior further underscores the importance of combining protocol encoding with other features in the detection system, as relying solely on frequency may bias models toward overclassifying ICMP-related activity. Fig. 10 shows Protocol Frequency.

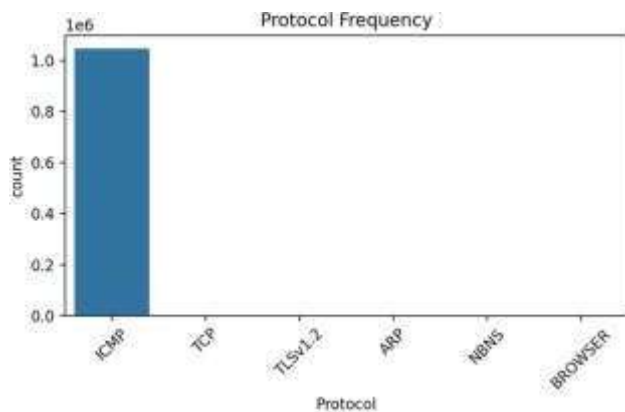


Fig. 10. Protocol Frequency.

From the statistical and visual analysis, several patterns emerged that directly support the design of the proposed CNN model. Packet length and time-based distributions exhibited partial separability of classes, indicating that CNN filters can extract meaningful local patterns. Temporal bursts and rolling bad-packet rates demonstrated the need for sequence-aware convolution, validating the overlapping window input strategy. Port frequency distributions and protocol-specific malicious rates revealed structural anomalies that can be encoded as contextual features. Finally, inter-arrival time profiles provided temporal variability, which is essential for distinguishing between stealthy malicious traffic and legitimate traffic. The uniqueness of this research lies in integrating these heterogeneous insights into a single hybrid CNN framework that combines dual convolutional paths with attention mechanisms to capture both short-term packet irregularities and long-range dependencies. Unlike prior studies that relied heavily on handcrafted features or focused on aggregated traffic, this research introduces a novel packet-level, context-aware CNN model capable of adapting to dynamic environments and achieving high accuracy even under extreme class imbalance. This novelty ensures scalability and robustness, positioning the proposed system as a forward-looking solution for real-time intrusion detection in emerging 6G-ready network infrastructures.

V. LIMITATIONS

This research, while demonstrating strong potential, is subject to certain constraints. The dataset is highly imbalanced, which may influence detection performance despite the use of class weighting and precision-recall evaluation. In addition, the analysis focused primarily on packet-level features, and incorporating flow-level or multi-session attributes could further enhance performance. The proposed CNN model was validated on a single dataset; broader testing on diverse real-world traffic traces would strengthen its generalizability. Finally, although the architecture is designed for scalability, further optimization may be required to ensure seamless deployment in high-speed production networks.

VI. CONCLUSION

This research presented a novel CNN-based architecture for real-time intrusion detection that effectively integrates

packet-level statistics, temporal burstiness, and contextual protocol-port features. The quantitative analysis confirmed that structural patterns such as packet length distributions, inter-arrival times, and rolling bad-packet rates hold significant discriminatory power for distinguishing malicious from benign traffic. Unlike conventional models that rely on handcrafted features or aggregate flows, the proposed approach introduces a hybrid design with dual convolutional paths and attention mechanisms, enabling both local and long-range pattern recognition. This novelty makes the system highly adaptive to dynamic traffic conditions and suitable for 6G-ready environments. The findings highlight that packet-level anomalies, when captured through sequence-aware deep learning, provide a scalable and robust basis for modern IDS. Future research can build upon this foundation by incorporating flow-level correlations, integrating unsupervised anomaly detection for zero-day threats, and testing the model across multi-source, real-world traffic datasets. Additionally, exploring hardware accelerations, such as FPGA or GPU-based deployment, could enhance processing speed in high-throughput environments. Ultimately, this study demonstrates that CNN-driven models can form the backbone of intelligent, adaptive, and future-proof network security solutions.

REFERENCES

- [1] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In *2017 International conference on advances in computing, communications and informatics (ICACCI)* (pp. 1222-1228). IEEE.
- [2] Thirimanne, S. P., Jayawardana, L., Yasakethu, L., Liyanaarachchi, P., & Hewage, C. (2022). Deep neural network based real-time intrusion detection system. *SN Computer Science*, 3(2), 145.
- [3] Lin, W. H., Lin, H. C., Wang, P., Wu, B. H., & Tsai, J. Y. (2018, April). Using convolutional neural networks to network intrusion detection for cyber threats. In *2018 IEEE International Conference on Applied System Invention (ICASI)* (pp. 1107-1110). IEEE.
- [4] Khan, M. A. (2021). HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes*, 9(5), 834.
- [5] Yu, K., Nguyen, K., & Park, Y. (2022). Flexible and robust real-time intrusion detection systems to network dynamics. *IEEE Access*, 10, 98959-98969.
- [6] Vishwakarma, M., & Kesswani, N. (2022). DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. *Decision Analytics Journal*, 5, 100142.
- [7] Kandhro, I. A., Alanazi, S. M., Ali, F., Kehar, A., Fatima, K., Uddin, M., & Karuppayah, S. (2023). Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures. *IEEE Access*, 11, 9136-9148.
- [8] Dong, Y., Wang, R., & He, J. (2019, October). Real-time network intrusion detection system based on deep learning. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)* (pp. 1-4). IEEE.
- [9] Mithileash, A., & Rajkumar, K. (2025, March). Integrating Convolutional Neural Networks For Enhanced Real-Time Intrusion Detection And Automated Attack Classification. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)* (pp. 1-6). IEEE.
- [10] Chen, H., Wang, Z., Yang, S., Luo, X., He, D., & Chan, S. (2025). Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments. *Alexandria Engineering Journal*, 111, 78-91.
- [11] Hnamte, V., & Hussain, J. (2023). Dependable intrusion detection system using deep convolutional neural network: A novel framework

- and performance evaluation approach. *Telematics and Informatics Reports*, 11, 100077.
- [12] Desta, A. K., Ohira, S., Arai, I., & Fujikawa, K. (2022). Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Vehicular Communications*, 35, 100470.
- [13] Ghadermazi, J., Shah, A., & Bastian, N. D. (2024). Towards real-time network intrusion detection with image-based sequential packets representation. *IEEE Transactions on Big Data*, 11(1), 157-173.
- [14] Chen, J., Zhang, L., Riem, J., Adam, G., Bastian, N. D., & Lan, T. (2023, November). Ride: Real-time intrusion detection via explainable machine learning implemented in a memristor hardware architecture. In *2023 IEEE Conference on Dependable and Secure Computing (DSC)* (pp. 1-8). IEEE.
- 10.48047/jocaaa.2026.35.01.92
- [15] Drewek-Ossowicka, A., Pietrolaj, M., & Rumiński, J. (2021). A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 497-514.
- [16] Kakani, T. A. (2025). Optimization of Serverless Mobile Cloud Applications for Enhanced Security and Resource Efficiency. *Optimization*, 5(1).
- [17] Mohammed, A. S., Sreeramulu, M. D., Neravetla, A. R., Gupta, K., & Yuvaraj, N. (2024, July). An Analysis of Security Protocols for Cloud Computing Algorithms in Mobile Ad Hoc Networks. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)* (pp. 1316-1321). IEEE.
- [18] Menon, A. N. (2023). *Network traffic data for malicious activity detection* [Data set]. Kaggle. <https://www.kaggle.com/datasets/advaitnmenon/network-traffic-data-malicious-activity-detection>