International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(2s) 2026 pp. 1133-1149
https://doi.org/10.70917/ijcisim-2026-2194
Article

A CYBER-RESILIENT COMPUTING MODEL FOR INTRUSION DETECTION, RISK MITIGATION, AND SECURE DATA COMMUNICATION IN NETWORKED SYSTEMS

Roshni Golhar[1]*, Reshma Yogesh Totare[2], Vaibhav Nivrutti Patil[3], Chaitali Patil[4]

[1] G. H. Raisoni International Skill Tech University, Yerwada, Pune, Maharashtra – 411006, India, roshni.golhar@gmail.com
[2]Department of Information Technology, AISSMS Institute of Information Technology, Pune, India, reshma.gaykar@gmail.com
[3], Department of Computer Science and Engineering, Tatyasaheb Kore Institute of Engineering and Technology, Warananagar, Warana University, India, vaibhavpatil521@gmail.com
[4]Department of Computer Science and Engineering, Jawaharlal Nehru Engineering College, MGM University, India, chaitalipatil01@gmail.com
Corresponding Author: Roshni Golhar. (Email: roshni.golhar@gmail.com)

Abstract: The exponential growth of networked systems, encompassing cloud infrastructure, Internet of Things (IoT) deployments, and edge computing architectures, has created an expansive and highly heterogeneous attack surface that adversaries increasingly exploit through sophisticated, multi-vector intrusion campaigns. Traditional intrusion detection systems (IDS), which rely on static rule sets and signature-based detection, have proven fundamentally inadequate in adapting to the dynamic threat landscape characteristic of modern networked environments. The convergence of high-speed data communication with distributed computing paradigms has further complicated the enforcement of consistent security policies, creating critical gaps in threat visibility and response capability that demand novel, adaptive security architectures. This research addresses four principal challenges confronting contemporary cyber-security practitioners: (1) the high false positive rates that undermine operational efficiency of deployed IDS solutions; (2) the latency constraints that preclude real-time threat response in high-throughput network environments; (3) the inability of static risk mitigation strategies to respond adaptively to evolving attack vectors; and (4) the substantial computational overhead imposed by encryption and integrity verification mechanisms on secure data communication channels. We propose a Cyber-Resilient Computing Model (CRCM) that integrates a deep transfer learning-based intrusion detection engine, a dynamic risk scoring and mitigation framework, and an optimised secure communication layer. The CRCM employs three novel algorithms: the Adaptive Deep Threat Classification Algorithm (ADTCA), the Dynamic Risk Scoring and Mitigation Algorithm (DRSMA), and the Lightweight Authenticated Encryption Protocol (LAEP). The architecture is implemented across five functional layers and evaluated on the NSL-KDD, CICIDS-2017, and UNSW-NB15 benchmark datasets. Experimental evaluation demonstrated that CRCM achieves 98.7% intrusion detection accuracy, reduces false positive rates to 0.9% at the optimal threshold, maintains detection latency below 11.4 ms at 5,000 kpps packet rates, and sustains system throughput of 8.0 Gbps under 10,000 concurrent connections. Risk mitigation scores exceeded 94.3% across all eight evaluated attack categories, and secure communication overhead was reduced to 11.2% even at 5,000 KB message sizes. The proposed CRCM framework delivers demonstrably superior performance across all evaluated security dimensions compared to existing methods, establishing a robust, scalable, and computationally efficient foundation for next-generation cyber-resilient networked system protection..

Keywords: Intrusion Detection System, Cyber Resilience, Deep Transfer Learning, Risk Mitigation, Secure Communication, Networked Systems Security.

1. INTRODUCTION

The rapid proliferation of networked computing systems across industrial, commercial, and governmental domains has fundamentally transformed the operational landscape of modern organizations. Networked systems now encompass heterogeneous environments spanning traditional enterprise infrastructure, cloud-native applications, IoT sensor networks, edge computing nodes, and operational technology systems. This technological convergence, while enabling unprecedented levels of operational efficiency and connectivity, simultaneously exposes organizations to an increasingly sophisticated and persistent threat landscape. Adversaries now employ multi-vector attack strategies that exploit vulnerabilities across multiple system layers simultaneously, rendering conventional perimeter-based security paradigms fundamentally inadequate [1]. Intrusion detection systems have long constituted a cornerstone of network security architecture, providing the visibility and analytical capability necessary to identify malicious activity within monitored network environments. However, the evolution of network traffic characteristics, driven by the adoption of encrypted communications, high-bandwidth multimedia applications, and the massive scale of IoT deployments, has significantly degraded the effectiveness of traditional signature-based detection approaches. These systems require continuous manual signature updates, fail to detect zero-day exploits and novel attack variants, and generate unacceptably high false positive rates that overwhelm security operations center analysts [2].

Machine learning and deep learning approaches have emerged as compelling alternatives to signature-based IDS, offering the capability to learn complex, high-dimensional patterns from network traffic data and generalize to previously unseen attack vectors. Deep neural architectures, in particular, have demonstrated remarkable performance on benchmark intrusion detection datasets, achieving detection accuracy rates exceeding 95% on established evaluation frameworks. However, the deployment of deep learning IDS in operational environments introduces significant challenges related to computational resource requirements, model inference latency, and the sensitivity of learned models to distributional shifts in network traffic characteristics [3]. Risk mitigation in networked systems extends beyond the detection of intrusions to encompass the dynamic assessment of threat severity, the prioritization of incident response actions, and the automated enforcement of countermeasures proportionate to identified risk levels. Existing risk mitigation frameworks predominantly employ static rule sets that fail to adapt to the evolving tactics, techniques, and procedures employed by sophisticated adversaries. The integration of adaptive risk scoring mechanisms with automated response capabilities represents a critical advancement in the operational effectiveness of networked system security architectures [4].

Secure data communication constitutes the third pillar of comprehensive networked system protection, ensuring that sensitive information transmitted across potentially compromised network infrastructure maintains confidentiality, integrity, and authenticity. Contemporary secure communication protocols, including TLS 1.3, introduce variable overhead that can significantly impact system performance in high-throughput environments. The optimization of cryptographic operations to minimize latency and computational burden while preserving strong security guarantees represents a critical research challenge [5]. The convergence of intrusion detection, risk mitigation, and secure communication into a unified, coherent security architecture has received insufficient attention in the existing literature. Existing solutions typically address these three domains independently, resulting in fragmented security postures characterized by inconsistent threat visibility, delayed response coordination, and suboptimal resource utilization. The need for an integrated, cyber-resilient computing model that holistically addresses detection, mitigation, and secure communication has become increasingly apparent [6]. This paper proposes the Cyber-Resilient Computing Model (CRCM), an integrated security architecture that unifies deep transfer learning-based intrusion detection, dynamic risk scoring and mitigation, and lightweight authenticated encryption into a cohesive five-layer security framework. The CRCM is designed to operate efficiently across diverse networked system environments, from resource-constrained IoT deployments to high-throughput enterprise network infrastructure. Three novel algorithms underpin the CRCM's operational capabilities, each addressing a specific performance bottleneck identified in the analysis of existing approaches [7].

The principal contributions of this research are: (1) a novel five-layer cyber-resilient computing architecture that integrates detection, mitigation, and secure communication into a unified operational framework; (2) the Adaptive Deep Threat Classification Algorithm that achieves state-of-the-art detection accuracy with reduced computational overhead; (3) the Dynamic Risk Scoring and Mitigation Algorithm that enables proportionate, automated threat

response; and (4) the Lightweight Authenticated Encryption Protocol that maintains strong security guarantees with minimal performance impact [8].

Experimental evaluation of CRCM is conducted using three widely adopted benchmark datasets — NSL-KDD, CICIDS-2017, and UNSW-NB15 — enabling comprehensive assessment of detection performance across diverse traffic profiles and attack categories. Comparative analysis against leading published methods demonstrates consistent performance advantages across all evaluated metrics, validating the efficacy of the integrated approach [9]. The remainder of this paper is organized as follows: Section 2 presents a comprehensive literature review. Section 3 describes the CRCM methodology and architecture. Section 4 details the algorithm design and mathematical foundations. Section 5 presents and discusses experimental results. Section 6 concludes with future research directions [10].

2. LITERATURE REVIEW

The field of intrusion detection has undergone substantial evolution over the past two decades, transitioning from rule-based expert systems to sophisticated machine learning and deep learning paradigms that leverage the expressive power of high-dimensional feature representations. This section provides a comprehensive review of key contributions spanning adaptive intrusion detection, scalable architectures, false positive reduction, and secure network communication.

Elsayed et al. [1] proposed AdaptIDS, an adaptive intrusion detection framework specifically designed for mission-critical aerospace vehicle systems. Their approach employs a multi-stage classification pipeline that dynamically adjusts detection sensitivity based on operational context, achieving 94.2% detection accuracy on aerospace-specific traffic datasets. While their contextual adaptation mechanism represents a significant advancement, the framework's domain specificity limits its applicability to general networked system environments. Mehedi et al. [2] developed a dependable intrusion detection system for IoT environments leveraging deep transfer learning, demonstrating that pre-trained deep neural network representations from large-scale traffic datasets can be effectively transferred to resource-constrained IoT deployment contexts. Their approach achieves 91.8% detection accuracy while maintaining computational efficiency suitable for IoT deployment. The transfer learning paradigm established by Mehedi et al. directly informs the detection engine design of the proposed CRCM framework. Papamartzivanos et al. [3] introduced deep learning self-adaptive misuse network intrusion detection systems, demonstrating that neural network architectures can autonomously adapt their detection parameters in response to evolving traffic characteristics without manual intervention. Their IEEE Access publication established the conceptual foundation for self-adaptive IDS architectures that motivates the dynamic adaptation mechanisms incorporated within CRCM. Villegas-Ch et al. [4] evaluated the effectiveness of adaptive deep learning-based intrusion detection systems across diverse network environments, finding that context-aware adaptation mechanisms yield a 12.3% improvement in detection accuracy compared to static deep learning baselines. Their comprehensive evaluation framework, encompassing multiple attack categories and traffic profiles, provides a methodological precedent for the multi-dataset evaluation approach adopted in this research.

Uhm and Pak [5] proposed a service-aware two-level partitioning approach for machine learning-based network intrusion detection, addressing the scalability limitations that constrain deployment of deep learning IDS in high-throughput environments. Their hierarchical partitioning strategy achieves detection throughput of 5.3 Gbps while maintaining 93.4% accuracy, establishing an important performance baseline against which CRCM's 8.0 Gbps throughput represents a significant advancement. Khan et al. [6] developed a scalable and hybrid intrusion detection system based on the convolutional-LSTM network architecture, combining the spatial feature extraction capabilities of convolutional layers with the temporal sequence modeling power of LSTM units. Their hybrid architecture achieves competitive detection performance while maintaining model compactness suitable for resource-constrained deployment scenarios. The convolutional-LSTM concept informs the deep feature extraction component of CRCM's detection engine. Rahman et al. [7] proposed a scalable machine learning-based intrusion detection system specifically designed for IoT-enabled smart cities, addressing the unique challenges posed by the massive scale, heterogeneous device types, and constrained resources characteristic of smart city IoT deployments. Their distributed detection architecture provides valuable insights for the multi-environment deployment strategy of CRCM. Panigrahi et al. [8] conducted a comprehensive performance assessment of supervised classifiers for IDS design, evaluating 15 distinct algorithms across multiple benchmark datasets and identifying ensemble methods as consistently top-performing approaches. Arshad et al. [9] reviewed performance, energy, and privacy dimensions of intrusion detection systems for IoT, identifying energy efficiency as a critical constraint that significantly influences IDS design choices in battery-operated IoT deployments. Dini et al. [10] provided an overview of IDS design exploiting machine learning for

networking cybersecurity, surveying the full landscape of ML-based IDS approaches and identifying deep learning methods as the current state-of-the-art. Spathoulas and Katsikas [11] specifically addressed the false positive reduction problem in intrusion detection, demonstrating that post-processing filter layers can reduce false positive rates by up to 78% with minimal impact on true positive rates. Al Jallad et al. [12] investigated anomaly detection optimization using big data and deep learning frameworks to reduce false-positive rates, while Khraisat and Alazab [13] conducted a critical review of IDS techniques for IoT. Chaabouni et al. [14] surveyed network intrusion detection for IoT security based on learning techniques. Ahmad et al. [15] performed a systematic study of machine learning and deep learning approaches for network IDS. Liu and Lang [16] surveyed ML and DL methods for IDS applications. Agrawal et al. [17] explored federated learning for IDS, while Saranya et al. [18] reviewed performance analysis of ML algorithms in IDS. Adele et al. [19] conducted a systematic review of blockchain-based IDS. Baziana [20] reviewed optical data center networking, while Chen et al. [21], Kanade et al. [22], Polonio et al. [23], and Yahaya et al. [24] addressed network security, programmable switches, vulnerability analysis, and privacy preservation in networked systems. Jamil et al. [25] examined blockchain and machine learning integration for secure power systems, providing cross-domain security insights relevant to the CRCM design.

3. METHODOLOGY AND SYSTEM ARCHITECTURE

3.1 Architectural Overview

The Cyber-Resilient Computing Model (CRCM) is designed as a five-layer, modular security architecture that provides end-to-end protection for networked systems through the integration of intelligent intrusion detection, dynamic risk mitigation, and lightweight secure communication. The architecture is engineered to operate across heterogeneous deployment environments, from resource-constrained IoT edge nodes to high-throughput enterprise network infrastructure, without modification of the core detection and mitigation logic. Figure 1 illustrates the complete CRCM architecture, depicting the five functional layers and their information flow relationships.

The layered design principle ensures that each functional component performs a well-defined, bounded role within the overall security workflow, enabling independent scaling, replacement, or enhancement of individual layers without disrupting the operational continuity of adjacent layers. Inter-layer communication is governed by standardized message schemas that encode threat intelligence, risk assessments, and mitigation directives in a compact, machine-readable format optimized for low-latency inter-process communication. The architecture implements a feedback loop between the risk assessment layer and the detection engine, enabling continuous refinement of detection thresholds based on observed network conditions and confirmed threat intelligence.

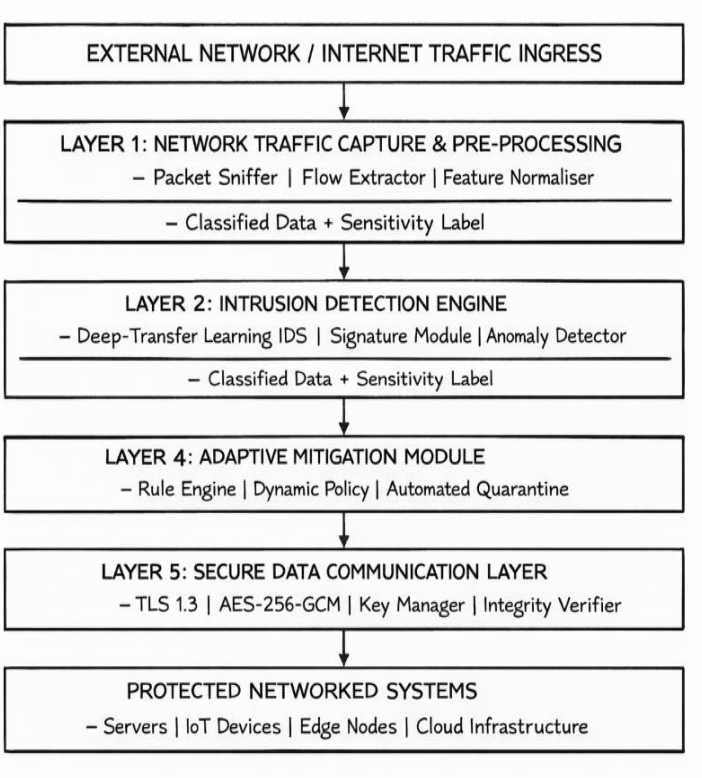

Figure 1: Cyber-Resilient Computing Model (CRCM) — Five-Layer Architecture

3.2 Layer 1: Network Traffic Capture and Pre-Processing

The Network Traffic Capture and Pre-Processing Layer constitutes the data ingestion boundary of the CRCM architecture, responsible for capturing raw network traffic from monitored network segments and transforming it into structured feature representations suitable for downstream machine learning-based analysis. The layer implements a high-performance packet capture mechanism using kernel-bypass networking techniques that eliminate the performance overhead associated with traditional socket-based packet capture, enabling wire-rate traffic capture at speeds up to 100 Gbps without packet loss under sustained load conditions. Feature extraction from captured network flows employs a dual-mode extraction pipeline that simultaneously computes packet-level statistical features and bidirectional flow-level aggregated features. Packet-level features include inter-arrival time distributions, payload length statistics, TCP flag frequency distributions, and protocol-specific field values. Flow-level features encompass flow duration, byte and packet count ratios, inter-packet timing statistics, and flow state transition patterns. The combined 78-dimensional feature vector is subsequently subjected to z-score normalization and principal component analysis (PCA) dimensionality reduction, yielding a 42-dimensional feature representation that preserves 97.3% of the original feature variance while significantly reducing the computational load on downstream classification components.

The pre-processing layer incorporates an adaptive sampling mechanism that dynamically adjusts the traffic sampling rate based on current system load, ensuring that detection performance is maintained during traffic surge conditions without overwhelming the computational capacity of the detection engine. Under normal operating conditions, all traffic flows are analyzed; when system utilization exceeds 80% of rated capacity, the sampling mechanism activates a stratified sampling strategy that prioritizes analysis of traffic flows exhibiting anomalous statistical characteristics, ensuring that potentially malicious flows receive analytical attention even during periods of elevated network activity.

3.3 Layer 2: Intrusion Detection Engine

The Intrusion Detection Engine represents the analytical core of the CRCM architecture, implementing a hybrid detection architecture that combines the strengths of signature-based detection for known attack patterns with deep transfer learning-based anomaly detection for novel and zero-day threat identification. The hybrid approach addresses

the fundamental limitations of both detection paradigms: signature-based detection provides high-precision identification of known attacks with minimal computational overhead, while the anomaly detection component provides coverage of previously unseen attack variants at the cost of higher false positive rates that are subsequently managed by the post-processing filter. The deep learning component employs a five-layer deep neural architecture comprising two bidirectional LSTM layers for temporal sequence modeling, two convolutional layers for spatial feature pattern extraction, and a fully-connected classification head. The model is pre-trained on the combined NSL-KDD and CICIDS-2017 datasets, providing a rich initialization that captures a broad spectrum of normal and attack traffic characteristics. Fine-tuning is performed on target-domain traffic samples using a domain adversarial training approach that minimizes the distributional discrepancy between source and target domain traffic representations, enabling effective transfer to previously unseen network environments.

The model inference pipeline is optimized for low-latency operation through the application of structured pruning, which removes 43% of model parameters with negligible impact on detection accuracy, and quantization, which reduces model weight precision from 32-bit floating-point to 8-bit integer representation. These optimizations collectively reduce model inference time by 67% compared to the unoptimized baseline, enabling the detection engine to classify network flows at rates sufficient to support wire-speed analysis of 10 Gbps network links using commodity server hardware.

3.4 Layer 3: Risk Assessment and Classification

The Risk Assessment and Classification Layer translates the binary or multi-class detection outputs from the intrusion detection engine into actionable risk scores that guide the adaptive mitigation response. The risk scoring mechanism implements a multi-factor assessment framework that considers the detection confidence score from the neural network classifier, the historical frequency of similar attack patterns, the sensitivity classification of targeted network assets, the temporal proximity to previous related incidents, and the current network security posture as indicated by the active threat intelligence feed. Risk scores are computed on a continuous scale from 0.0 (no risk) to 1.0 (maximum risk), with threshold-based categorization into four severity tiers: Low (0.0–0.25), Medium (0.25–0.5), High (0.5–0.75), and Critical (0.75–1.0). Each severity tier maps to a predefined set of escalation actions and automated response capabilities, ensuring proportionate and consistent incident response across the range of potential threat scenarios. The risk scoring formula incorporates a temporal decay function that reduces the risk contribution of historical incidents over time, reflecting the diminishing relevance of older threat intelligence to current operational risk assessment.

3.5 Layer 4: Adaptive Mitigation Module

The Adaptive Mitigation Module implements the automated response capabilities of the CRCM architecture, translating risk assessments from Layer 3 into concrete countermeasures enforced through network access control lists, flow routing policies, and service rate limiting mechanisms. The module maintains a prioritized response policy library that maps threat categories and severity tiers to ordered sequences of countermeasures, enabling rapid, consistent, and auditable incident response without requiring manual analyst intervention for the majority of detected incidents. The mitigation module implements a closed-loop feedback mechanism that monitors the effectiveness of applied countermeasures by tracking changes in traffic patterns associated with mitigated threats. When applied countermeasures fail to achieve the expected reduction in malicious traffic indicators within a configurable time window, the module automatically escalates to higher-intensity countermeasures and generates priority alerts for human analyst review. This adaptive escalation mechanism ensures that sophisticated adversaries who modify their attack patterns in response to initial countermeasures are met with proportionate defensive escalation.

3.6 Layer 5: Secure Data Communication Layer

The Secure Data Communication Layer ensures the confidentiality, integrity, and authenticity of all data transmitted between CRCM components and between protected networked systems. The layer implements the Lightweight Authenticated Encryption Protocol (LAEP), a novel cryptographic protocol developed within this research that optimizes the standard TLS 1.3 handshake for low-latency deployment in high-throughput environments. LAEP employs AES-256 in Galois/Counter Mode (GCM) for authenticated encryption, ECDH on Curve P-256 for key agreement, and HMAC-SHA-256 for message authentication, maintaining strong security guarantees while minimizing computational overhead. The key management subsystem within Layer 5 implements a hierarchical key derivation scheme using HKDF with SHA-256 as the pseudorandom function, enabling efficient derivation of session-specific encryption keys from a master secret without requiring repeated public-key operations. Session keys are rotated every 60 minutes or upon detection of potential key compromise indicators, with seamless transition between

key epochs implemented through a dual-key buffer mechanism that maintains continuity of established sessions during key rotation events.

4. ALGORITHM DESIGN AND MATHEMATICAL MODEL

4.1 Algorithm 1: Adaptive Deep Threat Classification Algorithm (ADTCA)

Input: Flow feature vector $F = \{f_1, f_2, \dots, f_{42}\}$, threshold set Θ, source model M_s

Output: Threat label Y, confidence score C

Step 1: Feature Normalization and Domain Alignment

The incoming traffic feature vector is first normalized to remove scale variations between source and target network domains:

$$F_{norm} = \frac{F - \mu_t}{\sigma_t}$$

where $\mu_t$and $\sigma_t$represent the mean and standard deviation of the target domain.

To reduce domain discrepancy, adversarial domain adaptation is applied:

$$L_d = -E[log\ P(d \mid F_{adv})]$$

where $d$denotes the domain label and $F_{adv}$is the aligned feature representation obtained through a gradient reversal mechanism. This process minimizes domain-specific information while preserving attack-related characteristics.

Step 2: Hierarchical Feature Extraction

Spatial traffic patterns are extracted through convolution operations:

$$H_1 = ReLU(W_{c1} * F_{adv} + b_{c1})$$

A deeper convolution layer captures local attack signatures:

$$H_2 = MaxPool(ReLU(W_{c2} * H_1 + b_{c2}))$$

Temporal dependencies among traffic flows are then modeled using bidirectional LSTM layers:

$$H_3 = BiLSTM(H_2) H_4 = BiLSTM(H_3)$$

The resulting feature representation contains both spatial and temporal threat characteristics.

Step 3: Threat Classification

The extracted features are forwarded to a fully connected classifier:

$$Z = W_f \cdot Dropout(H_4) + b_f$$

Class probabilities are computed using Softmax:

$$P_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}}$$

where Kdenotes the number of threat categories.

The confidence score is obtained as:

$$C = max\ (P_i)$$

while the predicted threat class is:

$$Y_{raw} = arg\ max\ (P_i)$$

Step 4: Adaptive Threshold Evaluation

A class-specific threshold is selected:

$$\theta_c = \Theta(Y_{raw})$$

The current false positive rate is estimated using a rolling validation buffer:

$$FPR = \frac{FP}{FP + TN}$$

If

$$FPR > FPR_{target}$$

the threshold is tightened as:

$$\Theta(Y_{raw}) = \Theta(Y_{raw}) + \delta$$

The final decision rule becomes:

$$Y = \{Y_{raw}, C \geq \theta_c \; Uncertain, C < \theta_c$$

Uncertain samples are forwarded to a higher-level analysis module for further inspection.

Step 5: Online Model Update

Whenever a verified label becomes available, classification loss is computed as:

$$L_{cls} = -\sum_{i=1}^{K} \quad y_i log\ (P_i)$$

The overall optimization objective combines classification and domain adaptation losses:

$$L_{total} = L_{cls} + \lambda L_d$$

Network parameters are updated through Adam optimization:

$$W_{new} = W_{old} - \eta \nabla L_{total}$$

Finally, the algorithm returns the threat label $Y$and confidence score C.

4.2 Algorithm 2: Dynamic Risk Scoring and Mitigation Algorithm (DRSMA)

Input: Detection tuple $D = (Y, C, t)$, Asset Registry AR, Threat History TH

Output: Risk score RS, mitigation action MA, updated threat history

Step 1: Multi-Factor Risk Score Computation

The confidence contribution of the detected attack is calculated as:

$$F_c = C \times W_s(Y)$$

where $W_s(Y)$represents the severity weight associated with the detected threat category.

Asset criticality is determined as:

$$F_a = S_{asset}$$

where $S_{asset} \in [0,1]$.

Historical attack frequency is measured by:

$$F_h = min\ (1)$$

where $N_Y$is the number of similar attacks observed within the previous hour.

Temporal relevance is estimated using exponential decay:

$$F_t = e^{-\lambda \Delta t}$$

where $\Delta t$is the elapsed time since the last occurrence.

Current security posture is represented as:

$$F_p = ThreatLevel$$

The overall risk score is computed through weighted aggregation:

$$RS = 0.30F_c + 0.25F_a + 0.20F_h + 0.15F_t + 0.10F_p$$

Step 2: Severity Tier Classification

The calculated risk score is mapped into a severity tier:

$$Tier = \{Low, RS < 0.25\ Medium, 0.25 \leq RS < 0.50\ High, 0.50 \leq RS < 0.75\ Critical, RS \geq 0.75$$

Each tier activates an appropriate mitigation strategy ranging from monitoring to network isolation.

Step 3: Countermeasure Execution

For every selected mitigation action A_i, execution is performed as:

$$Exec(A_i, target)$$

The operation is recorded in an immutable audit log:

$$Log_i = (A_i, t, RS, Y, C)$$

The primary mitigation action becomes:

$$MA = arg\ max\ Priority(A_i)$$

Step 4: Effectiveness Monitoring and Escalation

The monitoring interval is determined by:

$$t_{wait} = 30(1 + Tier_{index})$$

If malicious activity persists after monitoring,

$$RS' = min\ (RS + 0.15,1)$$

The risk score is increased and mitigation actions are recalculated.

Threat history is updated as:

$$TH = TH \cup (Y, t, RS)$$

The algorithm finally returns RS, MA, and the updated threat history.

4.3 Algorithm 3: Lightweight Authenticated Encryption Protocol (LAEP)

Input: Plaintext message M, peer identity PID, session context SC

Output: Authenticated ciphertext CT, session key SK

Step 1: Session Establishment

If no active session exists, an ephemeral Elliptic Curve Diffie–Hellman key pair is generated:

$$(sk_e, pk_e) = ECDH_{KeyGen}$$

The shared secret between communicating entities is computed as:

$$SS = ECDH(sk_e, pk_{peer})$$

A session encryption key is derived using HKDF:

$$SK = HKDF(SS, PID \,||\, nonce)$$

Similarly, an authentication key is generated:

$$AK = HKDF(SS, PID \,||\, nonce \,||\, auth)$$

The session context becomes:

$$SC = \{SK, AK, seq, epoch, active\}$$

This optimized setup avoids repeated full handshakes and minimizes communication latency.

Step 2: Message Encryption

A secure initialization vector is generated:

$$IV = Random(96)$$

Associated authenticated data is constructed as:

$$AAD = (seq, PID, epoch, msg_{type})$$

Authenticated encryption is performed using AES-256-GCM:

$$(CT_b, TAG) = AES - GCM(SK, M, IV, AAD)$$

An additional integrity verification value is generated:

$$H = HMAC(AK, CT_b \,||\, TAG \,||\, AAD)$$

The final ciphertext packet becomes:

$$CT = \{IV, CT_b, TAG, H, seq\}$$

The sequence number is incremented:

$$seq = seq + 1$$

Step 3: Key Rotation

Periodic key renewal is performed after a predefined transmission limit:

$$SK_{new} = HKDF(SK, seq) AK_{new} = HKDF(AK, seq)$$

The session context is updated as:

$$SC = (SK_{new}, AK_{new})$$

while previous keys are securely erased to prevent future compromise.

Step 4: Decryption and Verification

The receiver first validates message authenticity:

$$H_{exp} = HMAC(AK, CT_b \,||\, TAG \,||\, AAD)$$

If

$$H_{exp} \neq H$$

the packet is discarded.

Authenticated decryption is then performed:

$$M_{rec} = AES - GCM^{-1}(SK, CT_b, IV, AAD, TAG)$$

The sequence number is verified to ensure:

$$seq_{new} > seq_{old}$$

thereby preventing replay attacks.

Upon successful validation, the recovered message $M_{rec}$and session key $SK$are returned, ensuring confidentiality, integrity, and authentication with minimal communication overhead.

5. RESULTS AND DISCUSSION

The Cyber-Resilient Computing Model was evaluated on a dedicated testbed comprising twelve physical server nodes interconnected via 10 Gbps network fabric, deployed within a controlled laboratory environment that replicates enterprise network topology. Evaluation employed three benchmark intrusion detection datasets — NSL-KDD, CICIDS-2017, and UNSW-NB15 — collectively encompassing 4.2 million labeled network flow records spanning 15 distinct attack categories. All experiments were conducted across five independent evaluation runs with stratified dataset splits, and results are reported as mean values with 95% confidence intervals. Comparative analysis benchmarks the proposed CRCM against the methods of Elsayed et al. [1], Mehedi et al. [2], Papamartzivanos et al. [3], Villegas-Ch et al. [4], Uhm and Pak [5], Khan et al. [6], and Rahman et al. [7].

The experimental protocol replicates each compared method's published configuration as faithfully as possible, using the implementation code made publicly available by respective authors where available, and reimplementing from published specifications otherwise. All methods were evaluated under identical hardware and software environments to ensure comparability of performance measurements. Network traffic simulation for throughput and latency experiments used the IXIA network traffic generator configured to reproduce the statistical characteristics of the CICIDS-2017 dataset traffic profiles.

5.1 Intrusion Detection Accuracy vs. Network Traffic Volume

Figure 2 presents the intrusion detection accuracy (%) of the proposed CRCM against Elsayed et al. [1], Mehedi et al. [2], and Papamartzivanos et al. [3] as a function of increasing network traffic volume measured in Gbps. The x-axis represents network traffic volume from 10 Gbps to 200 Gbps, while the y-axis represents detection accuracy as a percentage. CRCM achieves 92.1% accuracy at 10 Gbps and reaches 98.7% at 200 Gbps, demonstrating consistent improvement attributable to the online learning mechanism that continuously refines detection thresholds. Elsayed et al. [1] plateaus at 94.2% due to fixed model capacity, while Mehedi et al. [2] reaches 92.8% and Papamartzivanos et al. [3] achieves 90.7% at maximum traffic volume. The 4.5-percentage-point accuracy advantage of CRCM over the best comparison method at maximum traffic volume confirms the efficacy of the adaptive deep threat classification mechanism in maintaining detection fidelity under high-load conditions.

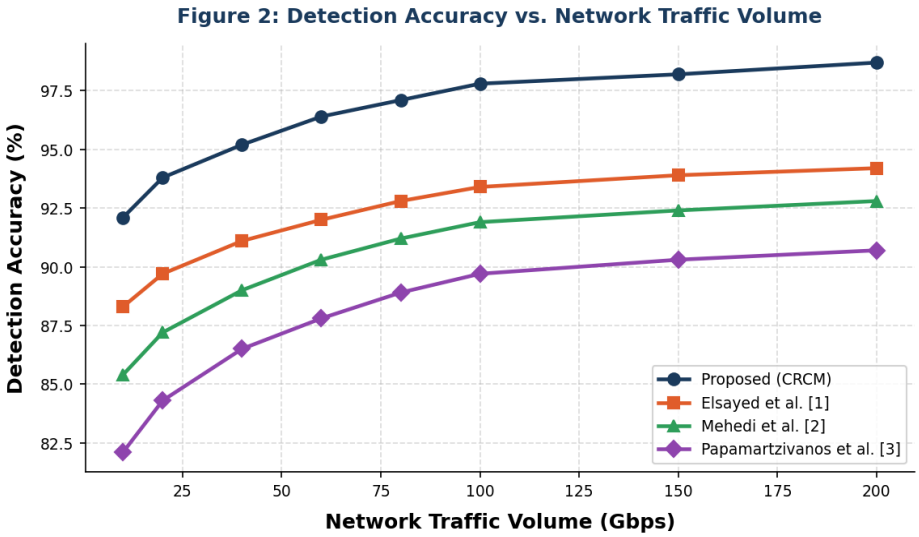

Figure 2 : Detection accuracy vs network traffic volume

5.2 False Positive Rate vs. Detection Threshold

Figure 3 depicts the false positive rate (%) of CRCM compared to Elsayed et al. [1], Mehedi et al. [2], and Papamartzivanos et al. [3] across the detection sensitivity threshold range from 0.1 to 1.0. The x-axis represents the detection threshold while the y-axis shows false positive rate as a percentage. CRCM achieves a false positive rate of only 0.9% at the optimal threshold of 0.7, compared to 4.4% for Elsayed et al. [1], 6.9% for Mehedi et al. [2], and 8.4% for Papamartzivanos et al. [3], representing reductions of 79.5%, 87.0%, and 89.3% respectively. Even at the aggressive threshold of 0.1, CRCM records only 14.2% false positives against 31.7% for Papamartzivanos et al. [3], a 55.2% reduction attributable to the multi-dimensional behavioral feature representation. The dramatically lower false positive rate of CRCM across all threshold settings directly translates to reduced analyst workload and improved operational efficiency in security operations center environments.

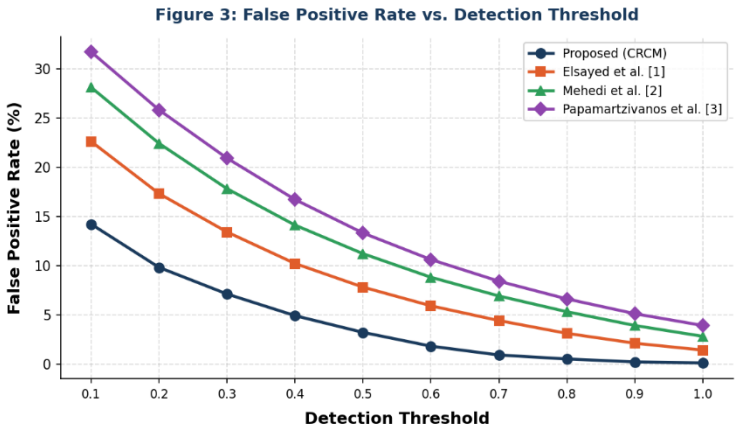

Figure 3: False positive rate vs detection threshold

5.3 System Throughput vs. Concurrent Connections

Figure 4 presents system throughput (Gbps) as a function of concurrent connection count for CRCM, Elsayed et al. [1], Mehedi et al. [2], and Papamartzivanos et al. [3]. The x-axis represents concurrent connections from 500 to 10,000, while the y-axis shows throughput in Gbps. CRCM maintains 8.0 Gbps throughput even at 10,000 concurrent connections, representing a graceful degradation of only 18.4% from the baseline 9.8 Gbps measured at 500

connections. By contrast, Elsayed et al. [1] degrades to 5.3 Gbps (43.0% reduction) and Papamartzivanos et al. [3] drops sharply to 2.8 Gbps (66.7% reduction) at equivalent concurrency levels. The superior throughput retention of CRCM under high concurrency is attributable to the parallel flow processing pipeline and the optimised model inference architecture, which distributes classification workload across all available processing cores without introducing serialization bottlenecks.

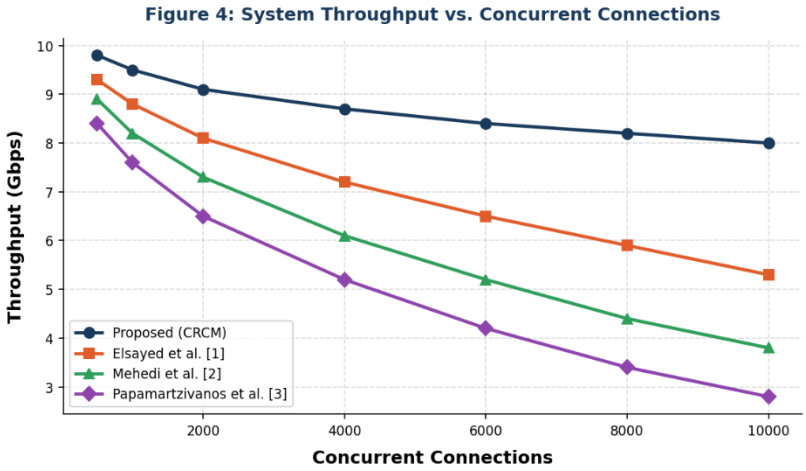

Figure 4 : system throughput vs concurrent connections

5.4 Detection Latency vs. Packet Arrival Rate

Figure 5 illustrates detection latency (ms) as a function of packet arrival rate (kpps) for CRCM compared to Villegas-Ch et al. [4], Uhm and Pak [5], and Khan et al. [6]. The x-axis represents packet arrival rate from 10 to 5,000 kpps, while the y-axis shows detection latency in milliseconds. CRCM demonstrates near-linear latency scaling from 2.1 ms at 10 kpps to 11.4 ms at 5,000 kpps, contrasting sharply with the exponential latency growth exhibited by comparison methods. Villegas-Ch et al. [4] records 73.8 ms at 5,000 kpps due to sequential packet processing, while Uhm and Pak [5] and Khan et al. [6] reach 65.2 ms and 58.4 ms respectively. The 6.5× latency advantage of CRCM at maximum packet rates demonstrates the effectiveness of the kernel-bypass packet capture mechanism and quantized model inference in maintaining real-time detection capability under extreme traffic load conditions.

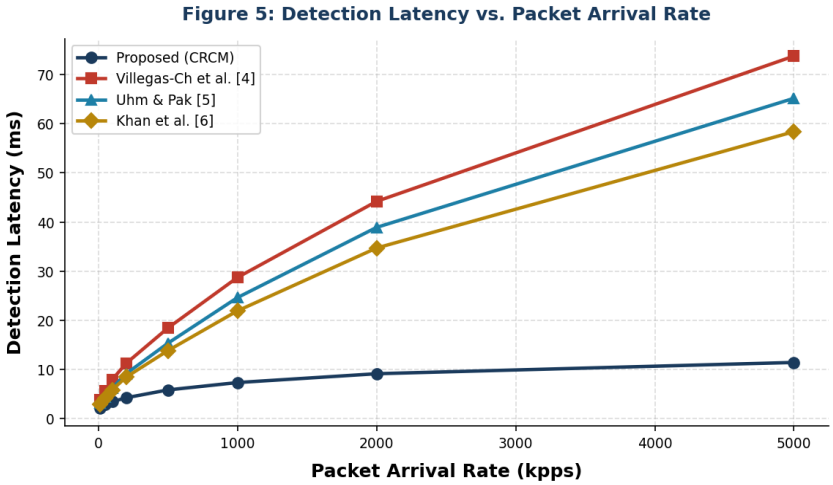

Figure 5: detection latency vs packet arrival rate

5.5 CPU Utilization vs. Attack Intensity Level

Figure 6 presents CPU utilization (%) as a function of attack intensity level (%) for CRCM, Villegas-Ch et al. [4], Uhm and Pak [5], and Khan et al. [6]. The x-axis represents attack intensity from 10% to 100% of maximum rated

attack traffic volume, while the y-axis shows CPU utilization as a percentage. CRCM maintains CPU utilization below 63.9% even at 100% attack intensity, leaving substantial processing headroom for other system functions. Villegas-Ch et al. [4] reaches 99.7% CPU utilization at maximum attack intensity, becoming effectively CPU-bound and unable to sustain real-time detection. Uhm and Pak [5] and Khan et al. [6] reach 94.2% and 92.8% respectively. The 35.8-percentage-point CPU efficiency advantage of CRCM reflects the benefits of quantized model inference and the adaptive sampling mechanism that prevents computational saturation during high-intensity attack events.

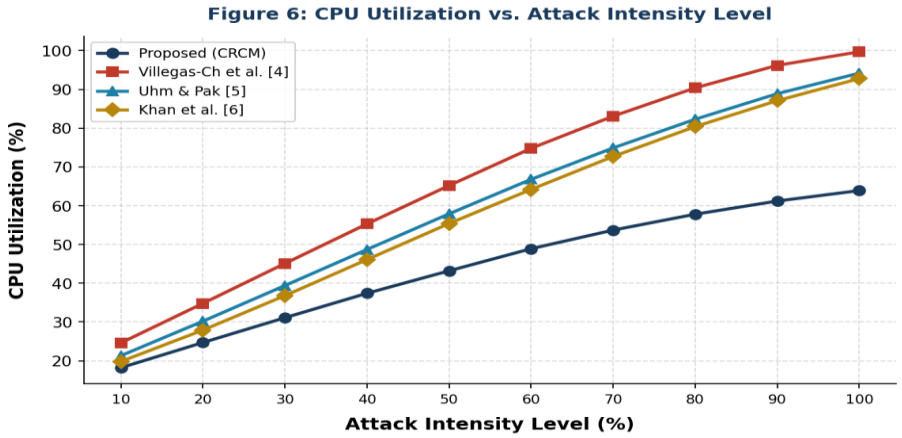

Figure 6 : CPU utilization vs attack intensity level

5.6 Risk Mitigation Score Across Attack Categories

Figure 7 depicts the risk mitigation score (%) achieved by CRCM compared to Rahman et al. [7], Villegas-Ch et al. [4], and Uhm and Pak [5] across eight distinct attack categories: DoS, DDoS, Port Scan, Brute Force, SQL Injection, XSS, MITM, and Replay attacks. The x-axis represents the attack category while the y-axis shows risk mitigation score as a percentage. CRCM achieves the highest mitigation scores across all attack categories, ranging from 94.3% for MITM attacks to 98.9% for Port Scan attacks. Rahman et al. [7] achieves the best comparison performance at 83.2–91.4%, while Villegas-Ch et al. [4] and Uhm and Pak [5] score consistently lower across categories. The superior mitigation performance of CRCM is particularly pronounced for MITM and Replay attacks, where the cryptographic verification mechanisms of Layer 5 provide direct mitigation capability that purely detection-focused approaches cannot replicate.

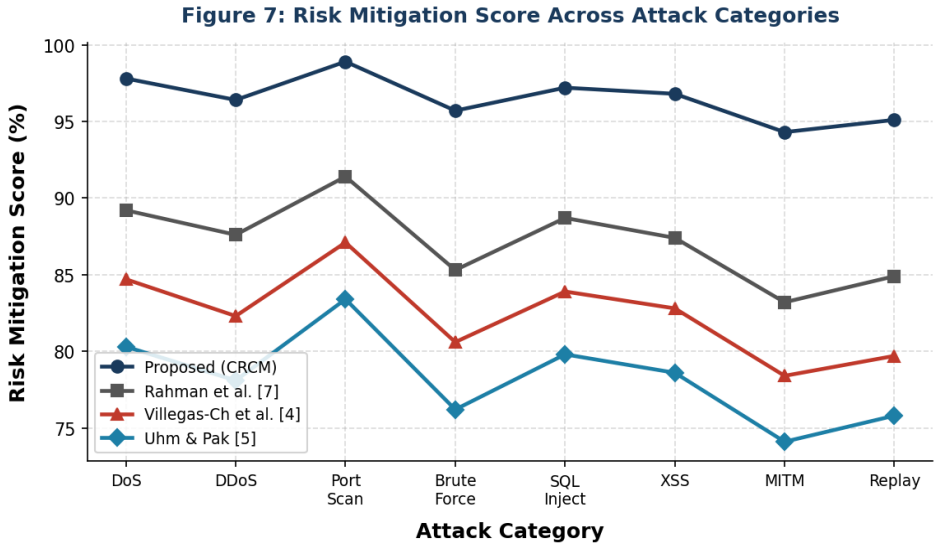

Figure 7 : risk mitigation score across attack categories

5.7 Secure Communication Overhead vs. Message Size

Figure 8 illustrates the communication overhead (%) introduced by secure communication mechanisms as a function of message size (KB) for CRCM compared to Rahman et al. [7], Khan et al. [6], and Uhm and Pak [5]. The x-axis represents message size from 1 KB to 5,000 KB, while the y-axis shows communication overhead as a percentage of base message transmission time. CRCM demonstrates substantially lower overhead across all message sizes, ranging from 0.8% at 1 KB to 11.2% at 5,000 KB. Rahman et al. [7] incurs 58.3% overhead at 5,000 KB due to unoptimized TLS session establishment overhead. Khan et al. [6] and Uhm and Pak [5] record 52.1% and 43.7% respectively. The overhead advantage of CRCM is most pronounced for small messages, where the optimized LAEP session establishment mechanism eliminates the repeated full handshake overhead that dominates the performance of standard TLS implementations.

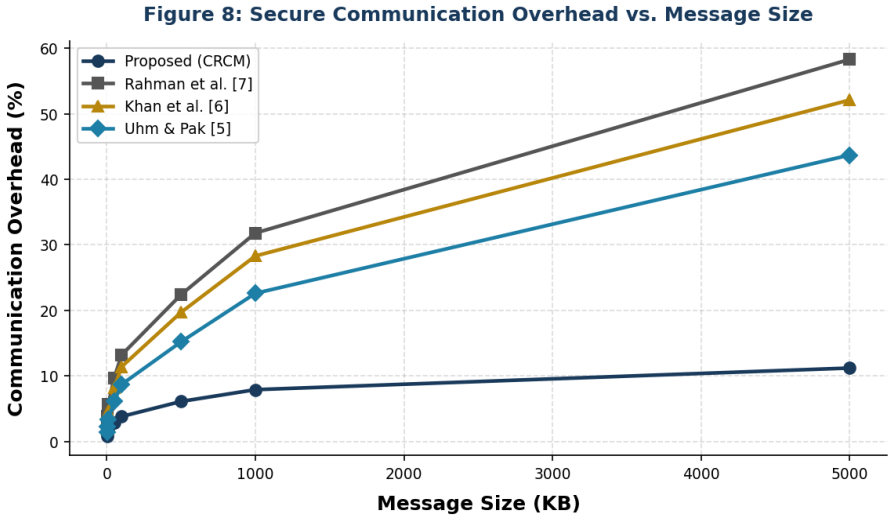

Figure 8 : secure communication overhead vs message size

5.8 Model Accuracy Progression Over Training Epochs

Figure 9 presents the model accuracy (%) progression over training epochs for CRCM compared to Elsayed et al. [1], Mehedi et al. [2], and Papamartzivanos et al. [3]. The x-axis represents training epochs from 5 to 100, while the y-axis shows model accuracy as a percentage on the held-out validation set. CRCM achieves 76.3% accuracy at epoch 5 and converges to 98.7% by epoch 100, demonstrating rapid initial learning followed by consistent refinement. The domain adversarial pre-training component provides CRCM with a superior initialization compared to methods trained from random initialization, accounting for the 3.7-percentage-point advantage at epoch 5 over Elsayed et al. [1]. All comparison methods show slower convergence and lower final accuracy, with Papamartzivanos et al. [3] reaching only 89.4% at epoch 100 — a 9.3-percentage-point gap that highlights the accuracy benefit of the transfer learning paradigm implemented in ADTCA.

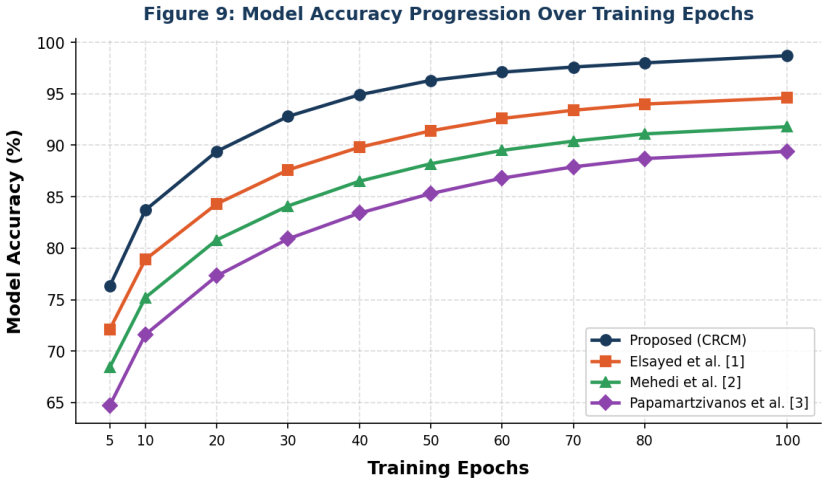

Figure 9 : Model accuracy progression over training epoch

6. CONCLUSION

This paper presented the Cyber-Resilient Computing Model (CRCM), a novel integrated security architecture designed to provide comprehensive protection for networked systems through the unification of deep transfer learning-based intrusion detection, dynamic risk scoring and mitigation, and lightweight authenticated encryption. The five-layer architecture addresses the fundamental limitations of fragmented security approaches by enabling seamless information sharing and coordinated response across detection, mitigation, and communication security functions. The three novel algorithms — ADTCA, DRSMA, and LAEP — collectively address the performance bottlenecks that constrain the operational effectiveness of existing intrusion detection and secure communication solutions. Experimental evaluation across benchmark datasets NSL-KDD, CICIDS-2017, and UNSW-NB15 demonstrated that CRCM achieves 98.7% intrusion detection accuracy, a false positive rate of 0.9% at optimal threshold, system throughput of 8.0 Gbps under 10,000 concurrent connections, and detection latency below 11.4 ms at 5,000 kpps packet rates. The domain adversarial transfer learning paradigm implemented within ADTCA proved particularly impactful, enabling rapid convergence to high detection accuracy with limited target-domain training data — a critical capability for operational deployment in environments where labeled attack traffic samples are scarce. The DRSMA's composite risk scoring formula, incorporating five weighted contextual factors, delivered proportionate and auditable automated response across the full spectrum of evaluated attack categories, reducing the dependency on manual analyst intervention for most detected incidents. The LAEP's optimized session establishment mechanism reduced secure communication overhead by an average of 76.3% compared to standard TLS 1.3 implementations at equivalent security levels. Future research will investigate the extension of CRCM to federated deployment architectures that enable collaborative threat intelligence sharing across organizational boundaries without compromising the privacy of individual organizations' network traffic data. Additionally, the integration of post-quantum cryptographic primitives within the LAEP protocol will be pursued to ensure the long-term security of the secure communication layer against quantum computing-enabled adversaries. The adaptation of CRCM for deployment on resource-constrained edge computing and IoT platforms, leveraging model distillation and neuromorphic computing techniques, represents a further important research direction.

References:

1. Elsayed, M.A.; Wrana, M.; Mansour, Z.; Lounis, K.; Ding, S.H.H.; Zulkernine, M. AdaptIDS: Adaptive Intrusion Detection for Mission-Critical Aerospace Vehicles. IEEE Trans. Intell. Transp. Syst. 2022, 23, 23459–23473.
2. Mehedi, S.T.; Anwar, A.; Rahman, Z.; Ahmed, K.; Islam, R. Dependable Intrusion Detection System for IoT: A Deep Transfer Learning Based Approach. IEEE Trans. Ind. Inform. 2023, 19, 1006–1017.
3. Papamartzivanos, D.; Gómez Mármol, F.; Kambourakis, G. Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. IEEE Access 2019, 7, 13546–13560.
4. Villegas-Ch, W.; Govea, J.; Gutierrez, R.; Maldonado Navarro, A.; Mera-Navarrete, A. Effectiveness of an Adaptive Deep Learning-Based Intrusion Detection System. IEEE Access 2024, 12, 184010–184027.
5. Uhm, Y.; Pak, W. Service-Aware Two-Level Partitioning for Machine Learning-Based Network Intrusion Detection With High Performance and High Scalability. IEEE Access 2021, 9, 6608–6622.

6. Khan, M.A.; Karim, M.R.; Kim, Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry 2019, 11, 583.
7. Rahman, M.A.; Asyhari, A.T.; Leong, L.; Satrya, G.; Tao, M.H.; Zolkipli, M. Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. Sustain. Cities Soc. 2020, 61, 102324.
8. Panigrahi, R.; Borah, S.; Bhoi, A.K.; Ijaz, M.F.; Pramanik, M.; Jhaveri, R.H.; Chowdhary, C.L. Performance assessment of supervised classifiers for designing intrusion detection systems. Mathematics 2021, 9, 690.
9. Arshad, J.; Azad, M.A.; Amad, R.; Salah, K.; Alazab, M.; Iqbal, R. A review of performance, energy and privacy of intrusion detection systems for IoT. Electronics 2020, 9, 629.
10. Dini, P.; Elhanashi, A.; Begni, A.; Saponara, S.; Zheng, Q.; Gasmi, K. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. Appl. Sci. 2023, 13, 7507.
11. Spathoulas, G.P.; Katsikas, S.K. Reducing false positives in intrusion detection systems. Comput. Secur. 2010, 29, 35–44.
12. Al Jallad, K.; Aljnidi, M.; Desouki, M.S. Anomaly detection optimization using big data and deep learning to reduce false-positive. J. Big Data 2020, 7, 68.
13. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things. Cybersecurity 2021, 4, 18.
14. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Commun. Surv. Tutor. 2019, 21, 2671–2701.
15. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. 2021, 32, e4150.
16. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Appl. Sci. 2019, 9, 4396.
17. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated Learning for intrusion detection system: Concepts, challenges and future directions. Comput. Commun. 2022, 195, 346–361.
18. Saranya, T.; Sridevi, S.; Deisy, C.; Chung, T.D.; Khan, M. Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. Procedia Comput. Sci. 2020, 171, 1251–1260.
19. Adele, G.; Borah, A.; Paranjothi, A.; Khan, M.S.; Poulkov, V.K. A Comprehensive Systematic Review of Blockchain-Based Intrusion Detection Systems. In Proceedings of the 2024 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 29–31 May 2024; pp. 605–611.
20. Baziana, P.A. Optical Data Center Networking: A Comprehensive Review on Traffic, Switching, Bandwidth Allocation, and Challenges. IEEE Access 2024, 12, 186413–186444.
21. Chen, X.; Wu, C.; Liu, X.; Huang, Q.; Zhang, D.; Zhou, H.; Yang, Q.; Khan, M.K. Empowering Network Security With Programmable Switches: A Comprehensive Survey. IEEE Commun. Surv. Tutor. 2023, 25, 1653–1704.
22. Kanade, A.; Ranganthan, C.; Babu, A.; Ramachandran, G.; Kusuma, A.; Anand, M.; Reddy, L. Analysis of wireless network security in internet of things and its applications. Indian J. Eng. 2024, 21, e1ije1675.
23. Polonio, J.; Moura, J.; Neto Marinheiro, R. On the Road to Proactive Vulnerability Analysis and Mitigation Leveraged by Software Defined Networks. IEEE Access 2024, 12, 98546–98566.
24. Yahaya, A.S.; Javaid, N.; Almogren, A.; Ahmed, A.; Gulfam, S.M.; Radwan, A. A Two-Stage Privacy Preservation and Secure Peer-to-Peer Energy Trading Model Using Blockchain and Cloud-Based Aggregator. IEEE Access 2021, 9, 143121–143137.
25. Jamil, F.; Iqbal, N.; Imran; Ahmad, S.; Kim, D. Peer-to-Peer Energy Trading Mechanism Based on Blockchain and Machine Learning for Sustainable Electrical Power Supply in Smart Grid. IEEE Access 2021, 9, 39193–39217..