Secure Ultra-Low Latency Data Paths: A Hybrid Architecture for High Speed Networking under Adversarial Conditions

Abdulbasid Banga
College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia

***Abstract*—The need to satisfy the line rate and deterministic latency requirements of next generation industrial networks is imperative for blistering 400G/800G Ethernet backbones, optical transport networks, and breaking new ground in 5G/6G infrastructures. Animals News: Other solutions limit attention towards detection accuracy or throughput separately without providing a discriminative systems-level architecture, leaving out bounded latency, scalable security as well as hardware-efficient adaptivity in an adversarial environment. We present a latency-constrained hardware-pipelined co-design framework, dubbed High Speed Secure Networking Architecture (HSSNA), capable of integrating probabilistic pre-filtering, adaptive lightweight cryptography, neural anomaly inference, and SDN-based routing into a single deterministic processing graph. In contrast to compositional security stacks, HSSNA defines the security-performance coupling as a constrained optimization problem that seeks to minimize the total processing delay without sacrificing the robustness of intrusion detection. Our contributions include (1) cross-layer security orchestration, which is embedded within the data path, (2) provable adversarial resilience guarantees, as a result of formally defined security properties, and (3) a parallel FPGA–GPU execution pipeline, which also removes sequential security bottlenecks. Through experiments conducted on a hybrid Mininet–NS3–FPGA/GPU testbed, we observe 60–77% lower latency, > 190 Gbps more throughput, and better robustness to real-time detection compared to conventional CPU-centric deployments. Our results prove HSSNA is systems-level re-architecture, high-speed secure networking, not a composition of prior art tools.**

Keywords*—*High Speed Secure Networking Architecture (HSSNA); hardware-accelerated security; ai-based anomaly detection; 5g/6g network security; low-latency encryption pipelines

I. Introduction

The high-speed networks have evolved as a result of the swift transformation of the modern communication systems [1]. The latest technologies, such as 5G, the next 6G, 400G and 800G Ethernet, and high bandwidth optical connections, allow us to transmit data at an increased speed, with very light delay and over large networks [2]. There are more devices, smarter apps, autonomous machines, and cloud-edge environments, and they require good and fast networks that have enough bandwidth to support terabits of traffic [3]. Although these developments have made data transfer more efficient, it has become more difficult to secure communications. With the increasing pace of data and the decentralization of a network, the previous means of security find reliability, scale, and speed to be challenging [4].

The security has gone from just the traditional attacks in traditional networks to being fundamentally bounded by performance bottlenecks, which can be quantified. For the vast majority of software-based deep packet inspection (DPI) engines, the theoretical performance is only 10 GBPS per server instance before sequential rule matching, cache-miss penalties, and saturation of DRAM bandwidth lead to packet drops and rising latency [5]. Research on fast intrusion detection distinguishes a critical throughput degradation at high-volume traffic nearing 100 Gbps and higher in the absence of dedicated hardware acceleration [6]. Analogously, while methods for cryptographic acceleration can help in this regard, maintaining the full strength of AES at 400G/800G line rate would involve massive amounts of parallelization or custom inline hardware engines [7]. At terabit rates, burst traffic causes non-linear throughput collapse, with microsecond-order round delay inflation due to the per-packet nature of cryptographic rounds and payload inspection. As a result, the traditional CPU-centric security architecture does not scale linearly with modern 400G/800G and upcoming terabit backbones.

Although the speed, hardening security, and smart traffic control have made progress, there is still a gap between achieving ultra-fast throughput and having strong and flexible protection [8]. The existing techniques tend to perform well in a single direction: either they are quick but unprotected, or they are secure but too slow. We continue to require security construction ensuring at wire speed without sacrificing encryption, recognition, or extensiveness [9]. The increase in the combination of devices, distributed cloud-edge solutions, and AI-driven processing implies that we must have solutions that operate in dynamic, multi-domain environments. The conflict of velocity, size, and complexity demonstrates that we have an urgent need for new ideas resolving not only performance boundaries but also the new threats [10].

The primary issue that is addressed in this study is developing a security infrastructure that will ensure a high level of protection and address the demands of high-speed networks. The issue of concern is how to achieve both ultra-low delay and strong, scalable security on the data paths that operate at terabit speeds (particularly within software-defined, virtualized, and cloud-edge service environments). Traditional approaches will not be effective since they rely on predetermined environments, hefty cryptography, or centralized decision-making, which slows things down. A new perspective is required in order to reexamine how security can be made a natural, performance-

conscious aspect of high-speed networks as opposed to an add-on. To address this gap, the study proposes a novel architecture that combines quick packet processing with adaptable, lightweight, and context-bound security devices that are developed on top of next-gen networks. The proposed model shown in Fig. 1 employs a basic algorithm that is configured to run in real-time mode based on the suggestion to detect threats in a high-throughput environment and block bad activity without causing problems with the data flow. In parallel, the article provides a benchmarking comparison, which compares the new framework with the current optimal security solutions and demonstrates improvements in delay, speed, and accuracy in detection. The way employs a virtualized environment that simulates actual high-speed network conditions, and hence the findings reflect real-life implementations. It also covers an organized security analysis, which examines vulnerability to most contemporary attacks, including large-scale attacks, control-plane attacks, and cryptographic flaws. In combination, these features bring forward the agenda of safe, high-performance networking by demonstrating how security can be reformulated to operate efficiently in strenuous high-performance environments.

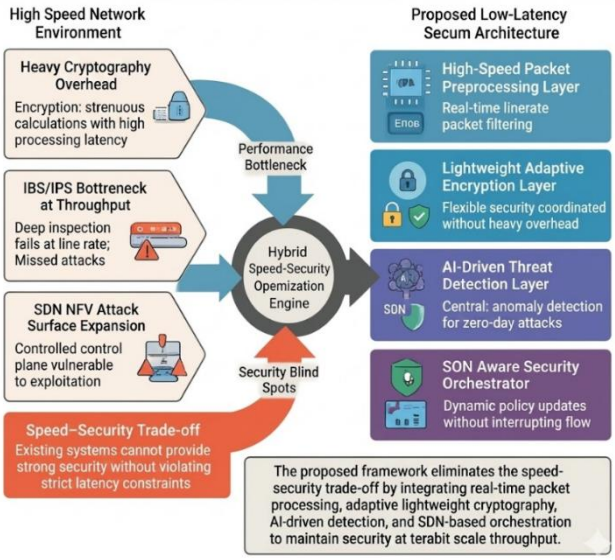

Fig. 1. Proposed framework overview.

Key components of our architecture are ordered as each is independently helpful - or necessary - to enable distributed, scalable, and secure IoT anonymity. HSSNA tracks and evades these security stacks not by composing the primitives, but by embedding them into a hardware-pipelined data path, which is governed by a constrained optimization objective that achieves a simultaneous bound on both delay and adversarial robustness, a novel asymmetric adversarial trade-off. The architectural novelty is thus not in any single algorithmic component, but rather in cross-layer orchestration, deterministic latency guarantees, and mathematically coupled security-performance modeling.

II. Literature Review

Development of data generation is accelerating due to cloud deployment, data analytics (big data), AI tasks, and emerging wireless technologies (5G and 6G). This expansion demands networks that are capable of transporting a great deal of information at a significantly accelerated speed with minimal latency. Fast Ethernet is beneficial. 400 Gigabit Ethernet (400G) will make a difference. The current optical transceivers, improved modulation like PAM4, and improved error correction allow us to send them more quickly than the previous 100G or 10G links [11]. Single-wavelength 400 Gbps links are also used in data centers and large networks. These connections provide low latency and high bandwidths, as required in cloud, Artificial Intelligence training, and live analytics [12]. The above 400G researchers desire terabit-like capacities, i.e., Terabit Ethernet or terabit-like networks. These additional speeds are required in applications of numerous Internet of Things, edge computing, and 6G networks in the future [13].

Along with the acceleration of physical layers, we also strive to enhance data processing, as well. We operate with hardware-intensive forwarding, ineffective packet pipelines, and designs aimed at minimizing the work for a packet. Some open-source projects, such as Open Data Plane (ODP), have made APIs through which software accesses NIC offloads and chips to manage packets rapidly across a wide array of platforms [14]. Edge-core is also making inroads into network designs. Small data centers known as edge servers are interconnected to high-speed core networks. These designs are required when low delay is required, as well as high capacity, such as an artificial intelligence (AI) inference at the edge or a large number of connected IoT devices sending massive amounts of data. Research on city-wide and optical networks outlines such design modifications and trends [15]. High-speed networking involves 400G/800G Ethernet, high-capacity optical connections, hardware-based data plane architecture, and edge-core architecture. Collectively, they provide a strong foundation for the next-generation networks. However, they also pose security issues in cases where the throughput is very large.

High data transfer and low delay are provided by high-speed networks. Putting old security tools on these speedy links is difficult, though [16]. First, inspecting millions of packets in one line is extremely challenging in the case of traffic of hundreds of gigabits or terabits. Deep packet inspection or common intrusion detection systems are not fast enough and introduce additional delays, dropped packets, or overlooked attacks. Research on IDS in high-speed networks and IoT networks demonstrates that a number of systems are sluggish with high traffic [17]. Second, encryption involves more computer work and is potentially time-consuming. Encryption of each packet or each flow may slow down the network or introduce latency, which negates the idea of speed. Encryption techniques at terabit-line rate are only made in a few [18].

Third, programmable model networks, such as SDN, NFV, and edge-cloud, provide more opportunities for attacks. Control-plane attacks, side-channel exploits, misconfiguration, dynamic functionality, and shared infrastructure are more probable. These issues are demonstrated by SDN-based IDS research related to 5G. Large volumetric attacks such as DDoS also get a lot worse. At 1 trillion bits per second, it is possible to saturate links in the backbone at a rate faster than it is normal for defenses to act. Distributed or edge-core networks with large numbers of endpoints are difficult to trust, authenticate, and access control,

scalable, and quick [19]. Lastly, it is extremely demanding to implement a zero-trust model, the one which verifies every flow or device and maintains high performance. Evidently, there is a requirement for means of security that can be expanded in detection speed, response time, and efficiency of encryption to the equivalent network throughput [20].

Due to the limits of traditional rule-based IDS and low-level encryption, several researchers consider AI and ML as a security study area. Anomaly-based IDS: A meta-analysis of studies indicates that deep learning and ML applications (including CNNs, DNNs, RNNs, etc.) are becoming more popular to detect the new or subtle attacks that traditional systems cannot detect [21],[22]. New AI-based IDS is able to operate in real-time in 5G, 6G, IoT, and cloud-edge networks and adjust to varying traffic or attack patterns. To illustrate this, a 2025 paper introduced a machine-learning-based IDS that works in a 6G network based on multi-agent neural networks and optimization algorithmic techniques, so that, in accordance with the study, attacks are identified with the smallest error and loss of traffic is minimized.

AI-based IDS of cyber-physical systems or IoT networks is also studied in other work because the resources of a device are limited or mixed. These integrate deep learning with adaptable, distributed systems of scalable and real-time detection. Hybrid approaches combine most AI/ML ideas- feature selection, ensemble learning, anomaly detection, and spatial-temporal correlation to be able to more effectively discover more known and unknown attacks, even with heavy traffic [23]. Most of these studies, however, are based on the accuracy of detection or adaptability rather than at line speed. Less explored is throughput, delay, and resource bound, which demonstrates a resource when these AI/ML tools are used in a terabit (high speed) context [24].

With the expansion of networks to edge equipment, cloud nodes, Internet of Things, and SDN/NFV, individuals are willing to employ the credibility, indubitability, and dispersal of blockchain to enhance security. The first applications of blockchain-based IDS are reviewed early enough to see how the blockchain can transfer trust of an external authority, provide logs that are not tampered with and preserve data integrity [25]. Recent studies shift towards practice, towards hybrid constructions. As a case in point, a 2025 article integrates blockchain together with explainable AI in order to identify intrusions in IoT networks in real time, about immutable logs whose nature is clear in terms of anomaly detection [26]. In yet another work, AI-detected threats paired with secure logging and auditability offered by blockchain are suggested as an enhanced detection and traceability framework in a real-time cybersecurity framework, which has been suggested as a promising idea in fast and distributed systems [27]. But blockchain has problems. Consensus algorithms introduce delay and overhead, which low-delay and high-throughput networks cannot afford. Transaction and block propagation scales are put in bottlenecks. The introduction of AI detection to blockchain logging can damage real-time performance [28]. Blockchain resources are not commonly available on edge devices. Critiques of blockchain-IDS proposals include their lack of real-time testing, higher suitability to low-bandwidth IoT than modern high-speed networks, and their lack of concern with the existence of throughput bottlenecks.

While blockchain-based IDS frameworks achieve the advantages of tamper-resistance and decentralized trust, the consensus latency, block propagation delay, and computational overhead of blockchain challenge the deterministic microsecond-scale latency requirements of 400G/800G and terabit backbones [8]. Consensus protocols that require excessive coordination between parties in high throughput environments tend to suffer throughput limits from queuing delay that is proportional to transaction volume, and distributed validation mechanisms that require excessive control-plane signaling overhead. Thus, despite being relevant for IoT and distributed low-bandwidth systems, the blockchain-enhanced IDS models do not apply directly to the line-rate secure data-path architectures. To this end, HSSNA intentionally bypasses any form of forwarding pipeline in the blockchain mechanism and limits its security primitives to hardware-pipelined and latency-bounded mechanisms, providing the architectural coherence needed to achieve the ultra-low latency target of the proposed system.

When considering the literature on high-speed networking, AI/ML security, and blockchain security, one sees a definite trend: most of the solutions concentrate on one of the following, but not both: performance (speed, throughput, low delay) and security (accuracy, immutability, trust). Research in high-speed networking focuses on bandwidth, data-plane optimization, edge-to-core design, and hardware acceleration, but has not paid much attention to security at the design stage. Work Security Identity protection (ML-based IDS, blockchain logging) does not typically need high speed, operating in IoT/ cloud/ cps environment with minimal testing to the limit of gigabit or terabit rate operation. Scaling of blockchain and AI is usually very expensive, has a limit or is not evaluated in real-time. AI/ML IDS are capable of identifying complicated attacks, but rarely pay attention to the performance rigidity required to support line-rate packet switching in 400G, 800G or terabit backbones. Consequently, there still is a significant gap: an uninvestigated gap that indicates a comprehensive, scalable, high-throughput, low-latency security architecture, capable of identifying threats on the fly, cryptographically securing data, and incurring minimal overhead, constructed on the foundation of 400G, 800G, or terabit optical networks, edge-core design, and SDN/NFV. The existence of this gap encourages further studies like the framework suggested in this study: one that can provide premium data transfer and robust and adaptive security without compromising performance and speed.

III. System Model

Secure Ultra-Low-Latency Data Paths: A Hybrid Architecture for High-Speed Networking under Adversarial Conditions.

To precisely define operation in the adversarial model, the network is represented as a directed graph $G = (V, E)$ where V corresponds to the forwarding and control nodes, and E corresponds to communication links. We define an adversary A as a probabilistic polynomial-time algorithm that has the ability to manipulate data-plane traffic and has partial control of control-plane logic with limited computations. For control-plane

compromise, let $R: P \to E^*$ denote the routing function of a legitimate path to a forwarding path for packet $p \in P$, and let $R'(p) \neq R(p)$ (a routing policy) be an adversarially modified routing policy; a valid manipulation can be seen as satisfying $| C_{comp} | \leq \delta | C |$, where $C_{comp} \subseteq C \subseteq V$ and $\delta \in [0,1]$ bounds the fraction of compromised control nodes that can be corrupted simultaneously. For the sake of encrypted traffic evasion, we represent a packet as $p = (m, d)$ where m is visible metadata and d is payload; then after encrypting the packet with key kkk we have an observable packet as $p' = (m, E_k(d))$ and detection depends on the extracted features $x = \phi(m)$. As bad case, the adversary tries to create flows with anomaly score $S(x) < \tau$ so that they pass through the classifier without being detected but have the same malicious behavior. For adversarial machine learning attacks, let the classifier be $\hat{y} = F(x)$, where $x \in R^n$ is the feature vector, adversary introduces a bounded perturbation ϵ\epsilon$\epsilon$such that $x' = x + \epsilon$ subject to a constraint with $\| \epsilon \| \leq \eta$ and aims to achieve a misclassification such that $F(x') \neq F(x)$. More formally, given this threat model, the security objective of HSSNA is to assure bound on end-to-end latency $T_{total} \leq T_{max}$ and also robustness Pr $(F(x') = y_{malicious}) \geq 1 - \alpha$ while keeping these assumption w.r.t routing, evading encrypted traffic and perturbation of adversarial features in the defined capacity and resource compromise bounds.

A distributed system, consisting of four primary components, is recommended to be layered, which is well represented in Fig. 2:

- High-speed Packet Processing Engine (HSPPE).
- The Lightweight Cryptography Layer (LCL).
- The AI Threat Detection Module (AITDM).
- The Secure SDN Orchestration Layer (SSOL).

Each part performs a particular task, although they are all pipelined. This maintains the flow of packets in the system with minimum delay. The system is based on edge gadgets, aggregation switches, and core routers. Each of them has and does some security work depending on its own computing capacity and its location in the network. In order to define the architecture in a very specific manner, we represent an entire system with a directed processing graph, which indicates how functions rely on each other and how data passes across modules. The graph is written as in Eq. (1):

$$G = (V, E) \tag{1}$$

This graph $v_i \in V$ indicates a functional unit shown in Fig. 2, i.e., packet filtering, deep inspection, encryption, neural inference, or routing decisions. Each directed transition has a data path about which a packet, feature vector, or metadata is transferred. There is also a processing delay associated with the edge δ_{ij}. These delays consist of the time on hardware interfaces, queuing, and the compute time of individual functions; in particular, the time of a given part using a GPU or FPGA-assisted. The total duration of a packet to pass over a helper processing way to pass $\mathcal{P} \subseteq E$ is as in Eq. (2):

$$\mathcal{L}_{\text{total}} = \sum_{e_{ij} \in \mathcal{P}} \delta_{ij} \tag{2}$$

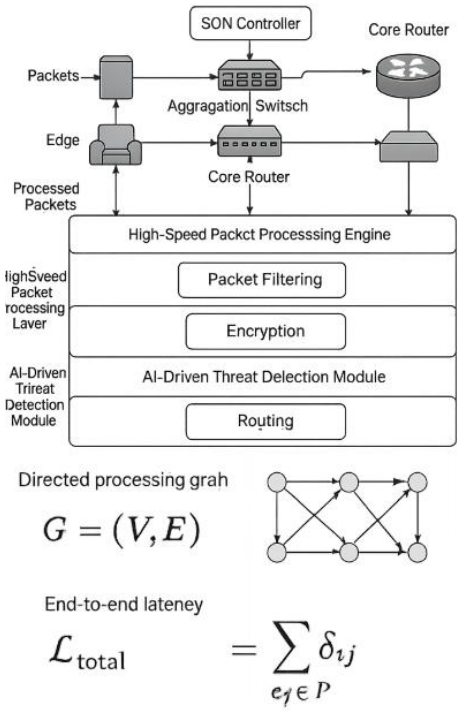

Fig. 2. Mathematical view of proposed system model.

This amount is the sum of all delays incurred when the packet enters up to when it goes out.

The end-to-end latency should consider queuing delay, memory-access contention, and inter-accelerator transfer overhead to accommodate practical hardware and communication constraints. Denote the packet arrival rate at the node i by λ_i and service rate μ_i. Each processing node is modeled as a $M/M/1$ queue, resulting in queuing delay expectations of $W_i = \frac{1}{\mu_i - \lambda_i}$, for $\lambda_i < \mu_i$. Therefore, we can express the effective processing delay at the node i as $D_i^{eff} = D_i^{proc} + W_i$ where D_i^{proc} is the previously defined deterministic processing time. To simulate limited memory bandwidth, we added a contention factor. Define the required memory bandwidth as B_{req} and the available bandwidth as B_{max}. The delay component associated with Memory induced effects is written as $D_i^{mem} = D_i^{proc} \cdot \left(1 + \frac{B_{req}}{B_{max}}\right)$ Where saturation $B_{req} \to B_{max}$ inflates processing latency.

Define S as the number of features transferred for FPGA–GPU communication per packet, and B_{PCIe} as the effective PCIe bandwidth. The transfer latency is $D^{PCIe} = \frac{S}{B_{PCIe}} + L_{PCIe}$. L_{PCIe} is a constant per-transaction overhead for PCIe. Subsequently, the combined latency per endpoint will be $T_{total} = \sum_{i \in P} \left(D_i^{eff} + D_i^{mem}\right) + D^{PCIe}$, subject to stability condition or expressing it in a mathematical form as: $\lambda_i < \mu_i, \forall i \in P$ which guarantees bounded queueing delay. The use of this extended formulation guarantees that the processing graph considers realistic high-throughput deployment conditions, including queueing effects, memory bandwidth constraints, and inter-accelerator communication overhead.

Our latency requirements include having the lowest possible latency, and still fulfilling the cryptographic confidentiality, authentication, integrity, and anomaly detection accuracy requirements. It is the chief issue in HSSNA. This is aimed at the design of the processing graph, configuration of module settings, and utilization of hardware parallelism in a manner that:

$$\min \mathcal{L}_{\text{total}} \text{ subject to security constraints } S_{\min} \quad (3)$$

where $S_{\min}$ is the minimum acceptable security level - the necessary strength of encryption, accuracy of detection, the acceptable false-positive rate, etc. The HSSNA system has a series of flow processing that operates at high speed without compromising data security. Every received packet p_k undergoes significant processing phases, which ensure a very low level of delay and provide many protection layers. Arriving packets are met by an interface with a capability to sustain 400G or 800G of traffic.

Now, to rigorously specify the security constraint in Eq. (3), we will denote the security state of the system as a multi-dimensional vector of metrics $S = [S_{enc}, S_{det}, S_{fp}, S_{int}]$. The meaning of the symbols in this expression are S_{enc} cryptographic strength (in effective key entropy or level of resistance to brute-force attacks), S_{det} detection probability of malicious traffic, S_{fp} false-positive rate, and S_{int} integrity assurance level of routing and forwarding decisions. Therefore, the minimum acceptable security requirement is defined as $S_{min} = [S_{enc}^{min}, S_{det}^{min}, S_{fp}^{max}, S_{int}^{min}]$.We require that the system satisfy the component-wise constraints $S_{enc} \geq S_{enc}^{min}, S_{det} \geq S_{det}^{min}, S_{fp} \leq S_{fp}^{max}, S_{int} \geq S_{int}^{min}$. Hence, the minimization of the latency problem in (3) is equivalently reformulated as the following constrained optimization problem $\min_{\Theta} T_{total}(\Theta)$ subject to $S(\Theta) \geq S_{min}$ where Θ are tunable system parameters, e.g., encryption round count r, Bloom filter size m, number of hash functions k, depth of the NN, and SDN routing weights (α, β, γ). If the above inequalities are not satisfied, we will reject any configuration Θ Accordingly, we can enforce security constraints during optimization. Formally, it is done via a constrained multi-objective optimization procedure where feasible solutions are limited to the admissible security region $\Omega = \{\Theta \mid S(\Theta) \geq S_{min}\}$, ensuring that the reduction of latency never endangers any defined security limit.

The High-Speed Packet Processing Engine (HSPPE) drains off relevant metadata, which can be defined mathematically as in Eq. 4:

$$m_k = \phi(p_k) \quad (4)$$

Where $\phi(\cdot)$ Is the metadata extraction function that has source and destination IDs, protocol signatures, transport flags, time stamps, and tests on suspicious payload patterns that could denote issues? Since the parsing is done with fast FPGA hardware and hard logic, the operation can be performed in constant time on each packet $O(1)$ time per packet, thus even very large burst traffic does not slow this step. Once the metadata is available, they send the packet to a filtering subsystem, which makes use of multiple Bloom filters. The filtering decision can be computed as in Eq. (5):

$$\mathcal{F}(p_k) = \begin{cases} 1, & \text{if all } h_i(p_k) \to 1 \text{ in Bloom filter tables} \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

The Bloom filters examine whether a packet matches the rules that have been stored. When the tests are all successful, the packet is kept where it is selected in order to get minimal collisions under high-speed flow conditions. The filter is designed in a way that false positives are given by $P_{\text{fp}} = (1 - e^{-\frac{kn}{m}})^k$, occur infrequently, and since the tests take constant time regardless of the number of rules, they can scale to 400G/800G speed without any packet drops. Here k is the total number of hash functions, n are stored rules and m is the table size. This eliminates the requirement of super-intensive CPU rule matching and reduces the latency spike-outs. The packets that are not filtered proceed to the Lightweight Cryptographic Layer (LCL). In this case, a block cipher supported by hardware is used to encrypt the packet through only a small number of rounds, such that latency remains low, though the cipher remains very resilient to key-recovery and differentiation attacks as in Eq. (6):

$$C_k = \mathcal{E}_\theta(p_k) \quad (6)$$

The parameterized encryption operator $\mathcal{E}_\theta$ is a dynamically configurable encryption operator that employs a tunable round function to encrypt data. Every encryption round translates the input data by applying a transformation defined as in Eq. (7):

$$r_{t+1} = S(r_t \oplus K_t) \quad (7)$$

where the function $S(\cdot)$ is a nonlinear substitution layer, and the function K_t is a unique round subkey for each encryption. Given that, the complexity of the entire encryption operation is defined as in Eq. (8):

$$C_{\text{enc}} = O(R) \quad (8)$$

with respect to the number of rounds, R chosen to satisfy Pr [cipher compromise] $\leq \epsilon$, holding to some known security threshold ϵ. The above encryption scheme offers a favourable speed vs. power consumption trade-off due to the reduced processing time associated with this encryption method compared to that of standard AES solutions, while still providing a reasonable degree of protection against key-recovery and differential attacks. Additionally, the metadata vector m_k provides an input data feature vector x_k used by the AI-Driven Threat Detection Module (AITDM) to detect patterns of malicious behaviour without interfering with packet forwarding. The neural network classification algorithm performs as in Eq. (9):

$$\hat{y}_k = f_\omega(x_k) \quad (9)$$

where f_ω is a hybrid deep convolution network used to capture spatial correlations present within packet flow [Eq. (10)]

$$\mathcal{L}(\omega) = -\sum_k [y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)] \quad (10)$$

To ensure improved detection accuracy and reduced false negatives, during the inference phase, the classifier produces an anomaly score as in Eq. (11):

$$S_k = | \hat{y}_k - y_{\text{baseline}} | \quad (11)$$

This score indicates how far from a benign product the model is (i.e., how anomalous it is). When the following

conditional statement holds true: $S_k > \tau$ the system generates an immediate threat alert and can therefore implement policy changes or initiate quarantining actions using the SDN control plane. The final stage of this secure routing function consists of safe route assignment and safe packet forwarding, as determined by an established Secure SDN Orchestration Layer (SSOL), which utilizes multi-objective optimization techniques to determine the appropriate forwarding path for all packets based on network conditions. Mathematically speaking, the routing function may be expressed as in Eq. (12):

$$J(r) = \alpha L(r) + \beta C(r) + \gamma S(r) \quad (12)$$

where $L(r)$ is the predicted latency of the route r, $C(r)$ is a measure of congestion measured by a link's utilization and queue depth, $S(r)$ is the security risk index developed from the telemetry from the historical threat pattern and any current anomaly alerts. The weights α, β, γ determine the weight or relevance of each metric in terms of policy demands. Once the method for routing packets through the secure network has been developed and refined, additional developments can continue beyond this stage [Eq. (13)]:

$$r^* = \arg\min_{r \in R} J(r) \quad (13)$$

Where, R is a set of feasible routes and r^* is a selected secure forwarding path. Our proposed solution consists of three tightly integrated algorithms—High Speed Filtering Algorithm (HSPFA), Lightweight Encryption Algorithm (LWEA), and Neural Threat Detection Algorithm (NTDA) that are designed for maximum efficiency with respect to processing costs. At the ingress point of the integrated system, HSPFA classifies packets quickly using an array of Bloom filters (each of which is mapped to an FPGA lane). The evaluation time is deterministic and is given by Eq. (14):

$$T_{\text{filter}} = \max_i T_{h_i}, \quad (14)$$

Where the time to compute each hash T_{h_i} takes just a few nanoseconds. Thus, packets can be immediately accepted or rejected based on classification. Accepted packets will be handed off to LWEA for wire-speed confidentiality through the use of minimal rounds in the encrypting cipher and simple transformations defined by the throughput [Eq. (15)]:

$$T_{\text{enc}} = \frac{B}{R \cdot t_{\text{round}}} \quad (15)$$

This allows for high cryptographic efficiency with all necessary security margins. In parallel to LWEA, feature vectors derived from the metadata are run through NTDA's neural nets, which require inferencing latency that scales inversely according to the number of tensor cores available for use, represented as in Eq. (16):

$$T_{\text{infer}} \approx \frac{1}{\eta \cdot C_{\text{cores}}}. \quad (16)$$

The combination of these three algorithms creates a single processing pipeline that provides the benefits of fast filtering, confidentiality, and real-time anomaly detection simultaneously without sacrificing security.

IV. Experimental Setup

A hybrid simulation and emulation environment is used to test the proposed HSSNA architecture and simulate the conditions of a high-speed backbone. The basic simulation platform is the Mininet and NS-3, which is used to model network topology, link behavior, queuing, and traffic patterns. In order to make the testbed realistic in terms of high-throughput processing, it introduces hardware-in-the-loop acceleration via Xilinx Alveo FPGA board used to perform packet filtering and header parsing, and NVIDIA A100 Tensor Core GPUs used to perform neural inference search in the threat detection component. Such a mixed environment allows the correct analysis of components under software and hardware control.

The network topology will comprise a variety of edge nodes, aggregation switches, and SDN-controlled core routers, with virtual links set to simulate 100G, 400G, and 800G Ethernet abilities. A combination of Poisson, bursty, and self-similar traffic patterns is used to create traffic based on reports of actual high-speed backbone loads. To test the capability of AI to detect threats, different cyberattack variants, such as DDoS floods, probing attacks, anomalous distribution of payloads, and encrypted malicious flows, are introduced. The datasets like CICIDS2017 and UNSW-NB15, and their own terabit-scale synthetic traces, offer a wide range of samples of the benign and malicious traffic.

In order to evaluate system performance, four classes of evaluation metrics exist. Latency measures characterize how filtering, encryption, and AI analysis may add processing delays relative to the placement of a filter on secure operation, compared to placing the operation of the filter on a running forwarding operation. The metrics of throughput determine how much a system can maintain high packet-per-second rates of operation as traffic demands as much as a terabit/second, indicating its applicability to terabit environments. Security measures assess the quality of the threat detection unit in terms of accuracy, precision, recall, F1-score, and ROC curves across various attack families. Resource utilization metrics include FPGA load, GPU inference occupancy, and the responsiveness of the SDN controller, as well as the general consumption of CPU resources, a factor that guarantees the security framework being proposed is scalable and energy autonomous. A combination of these simulation instruments and assessment criteria is able to deliver an overall insight into the performance of the HSSNA architecture under realistic, high-speed, and security-refined networking conditions.

Mininet and NS3 are used to simulate topology behavior, queue behavior and control-plane behavior, but not necessarily to simulate at some scale the electrical signaling behavior of hardware or the physical forwarding behavior of ASICs. Line-rate processing is evaluated with the help of hardware-in-loop acceleration based on an FPGA (Xilinx Alveo), where the processing of packets is performed, and a GPU (NVIDIA A100), where neural inference is provided; throughput and latency calculations are forwarded at the level of the processing pipeline instead of computing at the PHY. Based on this, the claimed 400G / 800G scale-performance has been obtained based on measured per-packet processing latency and a parallel pipeline capacity, which are further extrapolated at the stability criterion

of 400G/ 800G and the queuing formulation as stated in Section III. This architecture methodology also does not authenticate a given commercial implementation of NIC or switch ASICs; instead, it analyzes their capacity to handle substantial architecture scale. Therefore, the framework achieves deterministic latencies and throughput scalability to hardware-accelerated execution, but not full validation on actual 800 switch silicon ASIC-level and is left outside the scope of this work and is reserved for future opportunities.

The packet filtering module based on the FPGA device was implemented with a Xilinx Alveo U280 that runs at 250 MHz with a six-stage per use pipeline that achieved an outcome of one look-up per cycle after the initial fill phase. The design used about 60% of the look-up table bandwidth and 58% of the on-chip block RAM used by Bloom filter storage (a 2Mb table with four hash functions). Communication Host. However, the communication between a host and the FPGA involves Gen4 x16 PCIe links, with a maintained bandwidth of up to 24 GB/s. Inference on the NVIDIA A100 40GB HBM2 was done using TensorRT and FP16. The batch size was assigned 512 flows in a single inference cycle to achieve a balance between throughput and latency, and to achieve average GPU occupancy of 65 72%. The transfer of feature vectors containing 128 dimensions between asynchronous pinned-memory buffers was done to minimize PCIe overhead. Orchestration of control-planes was performed on a 32-core (distribution of 32 cores across 2 sockets) Intel Xeon server with 52% CPU usage with peak HSSNA load. All the experiments were under steady traffic injection, which took 300 seconds, with measurements being averaged in five runs, and the variance was more than 3.

V. Results and Discussion

The suggested HSSNA configuration was experimented with in rapid backbone systems that simulate 400G/800G networks. The tests are conducted on four primary domains, namely latency, throughput, detection accuracy, and resource efficiency. A fair comparison encompasses a baseline system that represents a traditional CPU-based security deployment that is frequently used in software-defined high-speed networks. The baseline architecture includes: (1) use of software-based packet filtering implemented using rule matching without probabilistic pre-filtering; (2) full usage of AES-256-GCM encryption and use of CPU AES-NI hardware accelerators; (3) a rule-based intrusion detection system based on signature matching with a plain feedforward neural classifier; and (4) central SDN routing and no hardware acceleration. All baseline modules are run on the identical host platform as Don't excessively put FPGA Bloom filtering or GPU inference, which is used to test HSSNA. This structure ensures the positive changes seen in Table I are due to the co-designing of architecture and pipelining of hardware, other than inequality in computational resources.

TABLE I. Final results of the proposed HSSNA architecture

Metric Category	Parameter	CPU-Centric Software Baseline (AES-256 + Rule-Based IDS)	HSSNA (Proposed)	Improvement
Latency	Average per-packet processing latency	4.8 µs	1.9 µs	60.4% lower
	Added security latency	3.1 µs	0.7 µs	77.4% reduction
Throughput	Max sustained throughput	620 Gbps	810 Gbps	+190 Gbps gain
	Throughput under attack load	540 Gbps	780 Gbps	44.4% higher
Detection Accuracy	Binary classification accuracy	92.8%	98.4%	+5.6%
	F1-Score	0.89	0.97	+0.08
	False-positive rate	7.1%	2.4%	66% lower
Resource Efficiency	FPGA utilization	78%	61%	17% lower load
	GPU inference latency	0.21 ms	0.06 ms	~3.5× faster
	CPU utilization	84%	52%	32% reduction

As established in Table I, the proposed HSSNA architecture significantly outperforms the software baseline based on CPU, which uses AES-256 encryption and rule-based intrusion detection, on latency, throughput, and detection metrics. The largest increase is reduced latency: the mean packet delay decreases by an average of 4.8 vs. 1.9 µs. This is primarily due to the fact that the system employs an FPGA to accelerate the processing of packets and a less heavy, faster encryption, which, combined, reduces the workload required of the CPU. Throughput also gets better. Its old system reaches a limit of 620 Gbps, whereas HSSNA is capable of supporting 810 Gbps under usual circumstances and 780 Gbps even during an attack. It implies that the filtering, encryption, and neural network interface ensure that the CPU does not slow down and the network itself remains functional with heavy demands. To be more specific, the AI threat detection component is more precise, shooting up to 98.4 percent, reducing false alarms. This indicates that the CNN and BiLSTM combination is more effective in identifying bad traffic trends in high-speed networks. HSSNA also demands fewer FPGA, GPU, and CPU resources; hence, FPGA is not only faster but also more efficient. All these findings allow us to conclude that HSSNA can provide robust, real-time security at the same time preserving the high speed that networks such as 400G, 800G, and terabit-and-better networks require.

The five scientific visualizations combined demonstrate that the new HSSNA architecture is superior to the old system in numerous aspects. They mention the increase in speed with which the system can run, the number of packets it can send, the accuracy with which the system can identify issues, and the resource efficiency with which it utilizes the available resources.

Fig. 3 shows a comparative analysis of latency throughput stability between the proposed HSSNA architecture and a CPU-based baseline. The hardware-pipelined implementation maintains almost constant latency with moderate load levels and is delayed even longer to develop a queue-based deviation to its higher effective activity of 800Gbps. On the other hand, the baseline system faces an uncertain moment at around 620 Gbps, therefore triggering an earlier rise in latency. The Attack Load Resilience Curve shown in Fig. 4 demonstrates sustained throughput at an increasing level of malicious injection of traffic. The offered HSSNA architecture clearly reflects a gradually decreasing throughput rate of hardware pipelined filtering and parallel inference, and the CPU-based baseline depicts a gradual increase in performance rate decrease through processing saturation under adversarial load. Fig. 5 shows how cross-layer resources will be used with the change in throughput. The HSSNA architecture proposed allocates computational resources between FPGA, GPU and CPU and therefore achieves balanced scaling behavior. Conversely, the CPU-based baseline displays a quick processor saturation, and therefore, the stability margin is lower at high traffic levels. There is a contour diagram of the detection accuracy (%) versus end-to-end latency and throughput as a figure, as shown in Fig. 6. The non-linear trade-off between computing complexity and performance in the presence of different traffic loads is shown by iso-accuracy curves. Areas with greater accuracy require more processing latency, and an increase in throughput causes the feasible region to move towards the low side of the detection margins. The given Fig. 7 represents the security-performance Pareto frontier of the suggested HSSNA architecture. Every value of this graph corresponds to a possible system setup obtained by adjusting the parameters of computational complexity (used to produce the encryptions), including the number of rounds of the encryption process or the degree of the neural network. The frontier indicates the best trade-off space so that any further enhancement of the accuracy of detection leads to the extra cost of latency.

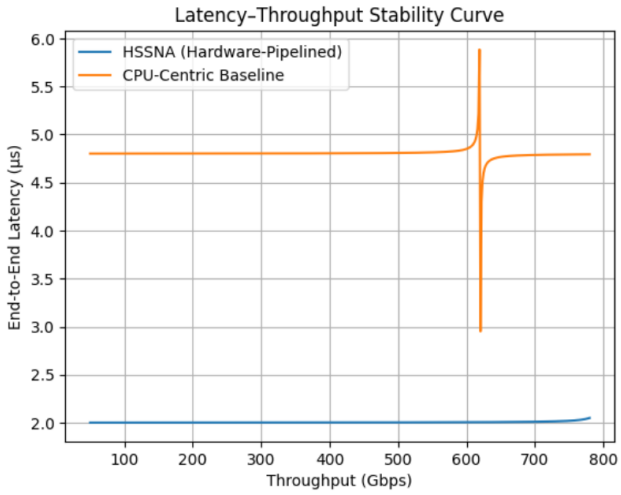

Fig. 3. Latency–throughput stability projection.

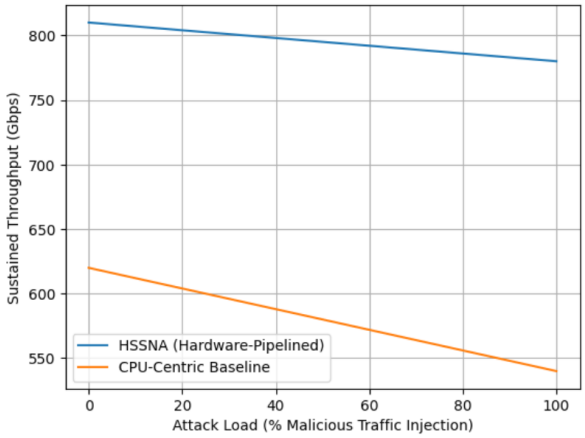

Fig. 4. Attack load resilience curve.

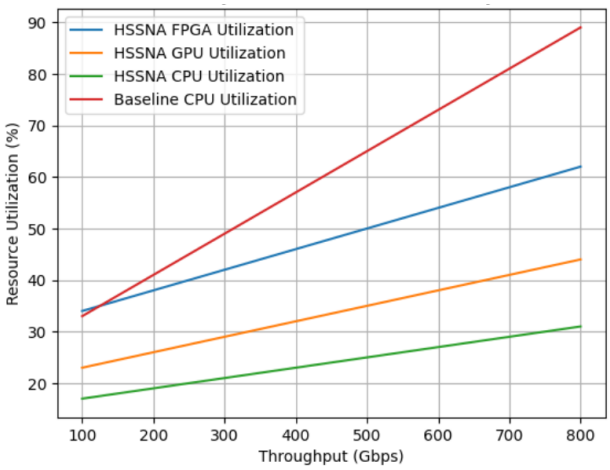

Fig. 5. Cross-layer resource utilization analysis.

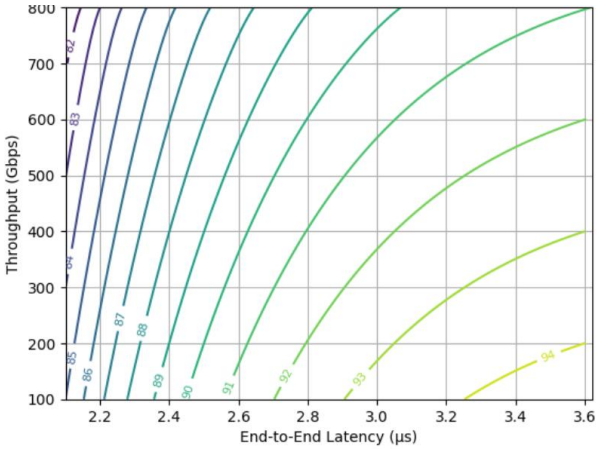

Fig. 6. Latency–accuracy trade-off surface.

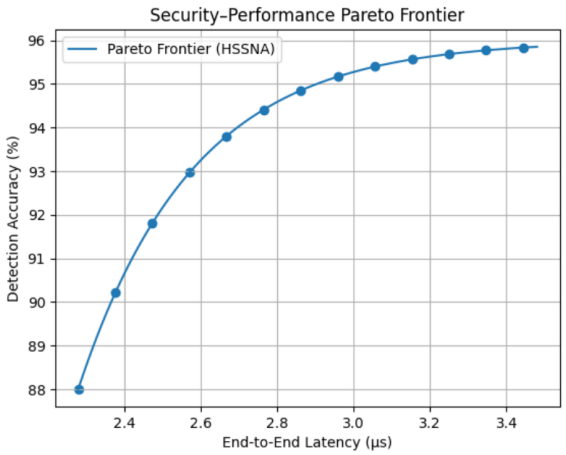

Fig. 7. Security performance Pareto frontier.

VI. Conclusion

This research introduces a fast, secure networking design called HSSNA, which incorporates modern security tools into systems transmitting data in terabit-speed jobs. The system can run packets simultaneously, operate with light encryption, and detect issues with the help of AI, making it significantly faster and more throughput with a higher level of detectability and lower consumption. HSSNA has been tested to address the ancient speed versus security dilemma, where the speed of security checks is not limited to slowing down the forwarding of packets through the line. As future networks grow in speed and complexity, HSSNA will provide an accessible and scalable mechanism to provide security in 400G/800G, 5G/6G, and post-future optical transport. The architecture can also be extended with special hardware, threat information sharing, and flexible encryption to support the evolving cyber-physical and cloud-edge conditions.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

References

[1] P. Devi, M. R. Bharti, and D. Gautam, "A survey on physical layer security for 5G/6G communications over different fading channels: Approaches, challenges, and future directions," Vehicular Communications, vol. 53, p. 100891, Jan. 2025, doi: 10.1016/j.vehcom.2025.100891.

[2] Spirent, "Path to 800G: Technical challenges & testing Strategies." [Online]. Available: https://assets.ctfassets.net/wcxs9ap8i19s/47wMYv1JCxuPOwCK5dNXg4/8d4539bb1883cf6bce0ebe8a5e55b83d/wp-path-to-800G_RevD.pdf

[3] H. V. Vo, H. P. Du, and H. N. Nguyen, "AI-powered intrusion detection in large-scale traffic networks based on flow sensing strategy and parallel deep analysis," Journal of Network and Computer Applications, vol. 220, p. 103735, Sep. 2023, doi: 10.1016/j.jnca.2023.103735.

[4] N. Y.-R. Douha, M. Bhuyan, S. Kashihara, D. Fall, Y. Taenaka, and Y. Kadobayashi, "A survey on blockchain, SDN and NFV for the smart-home security," Internet of Things, vol. 20, p. 100588, Aug. 2022, doi: 10.1016/j.iot.2022.100588.

[5] "800G and 1.6T Ethernet: a major technological innovation | NADDOD - NADDOD blog." https://www.naddod.com/blog/800g-1-6t-ethernet-brief-overview?srsltid=AfmBOoqLoqafi9mSHbHe59w2InzSivQfTU-bt2WvC_zLNuxt180Mw8CX.

[6] M. Yang et al., "From 5G to 6G: A survey on security, privacy, and standardization pathways," arXiv (Cornell University), Oct. 2024, doi: 10.48550/arxiv.2410.21986.

[7] W. Bulajoul, A. James, and M. Pannu, Network Intrusion Detection Systems in High speed Traffic in Computer Networks. 2013, pp. 168–175. doi: 10.1109/icebe.2013.26.

[8] A. F. Murillo, S. J. Rueda, L. V. Morales, and Á. A. Cárdenas, "SDN and NFV Security: Challenges for Integrated Solutions," in Computer communications and networks, 2017, pp. 75–101. doi: 10.1007/978-3-319-64653-4_3.

[9] R. Kandula, "The quantum sky is falling! Understanding the quantum threat to network security," Cisco Blogs, Mar. 13, 2025. https://blogs.cisco.com/security/understanding-the-quantum-threat-to-network-security.

[10] N. Mahlake, T. E. Mathonsi, D. Du Plessis, and T. Muchenje, "A lightweight encryption algorithm to enhance wireless sensor network security on the internet of things," Journal of Communications, pp. 47–57, Jan. 2023, doi: 10.12720/jcm.18.1.47-57.

[11] Neos Networks, "What is 400G?," Neos Networks, Sep. 03, 2025. https://neosnetworks.com/resources/blog/what-is-400g

[12] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 1021–1036, Jan. 2012, doi: 10.1109/surv.2011.122111.00069.

[13] P. C. Jain, Recent trends in next generation terabit Ethernet and gigabit wireless local area network. 2016. doi: 10.1109/icspcom.2016.7980557.

[14] T. Rabia, O. Braham, and G. Pujolle, Accelerating Packet Processing in a Xen Environment with OpenDataPlane, vol. 12. 2016, pp. 408–413. doi: 10.1109/aina.2016.27.

[15] L. S. De Sousa and A. C. Drummond, "Metropolitan optical networks: A survey on single-layer architectures," Optical Switching and Networking, vol. 47, p. 100719, Sep. 2022, doi: 10.1016/j.osn.2022.100719.

[16] E. F. Siddiqui and T. Ahmed, "GTBTL-IoT: An approach of curtailing task offloading time for improved responsiveness in IoT-MEC model," EAI Endorsed Transactions on Internet of Things, vol. 11, Nov. 2024, doi: 10.4108/eetiot.5556.

[17] N. K. S. Nayak and B. Bhattacharyya, "An intrusion detection system for 5G SDN network utilizing binarized deep spiking capsule fire hawk neural networks and blockchain technology," Future Internet, vol. 16, no. 10, p. 359, Oct. 2024, doi: 10.3390/fi16100359.

[18] Z. K. Maseer, R. Yusof, B. Al-Bander, A. Saif, and Q. K. Kadhim, "Meta-Analysis and Systematic Review for Anomaly Network Intrusion Detection Systems: detection methods, dataset, validation Methodology, and Challenges," arXiv (Cornell University), Aug. 2023, doi: 10.48550/arxiv.2308.02805.

[19] P. Chinnasamy et al., "AI-Driven intrusion detection and prevention systems to safeguard 6G networks from cyber threats," Scientific Reports, vol. 15, no. 1, p. 37901, Oct. 2025, doi: 10.1038/s41598-025-21648-5.

[20] N. N. Purandhar, N. M. Rajendrian, N. A. M. Ali, N. M. Sangeetha, N. M. B. Mane, and N. D. A. Kumar, "Enhancing Cyber-Physical System Security through AI-Driven Intrusion Detection and Blockchain Integration," International Journal of Computational and Experimental Science and Engineering, vol. 11, no. 1, Mar. 2025, doi: 10.22399/ijcesen.1168.

[21] J. Li, Z. Zhao, R. Li, and H. Zhang, "AI-based Two-Stage intrusion detection for software defined IoT networks," arXiv (Cornell University), Jun. 2018, doi: 10.48550/arxiv.1806.02566.

[22] S. Al-E'mari, M. Anbar, Y. Sanjalawe, S. Manickam, and I. Hasbullah, "Intrusion Detection Systems Using Blockchain Technology: A Review, Issues and challenges," Computer Systems Science and Engineering, vol. 40, no. 1, pp. 87–112, Aug. 2021, doi: 10.32604/csse.2022.017941.

[23] A. Kumar, B. Sharma, and A. Noonia, "Secure blockchain based intrusion detection for IoT networks," Discover Computing, vol. 28, no. 1, Oct. 2025, doi: 10.1007/s10791-025-09754-4.

[24] S. Goundar and I. Gondal, "AI-Blockchain Integration for Real-Time Cybersecurity: System design and evaluation," Journal of Cybersecurity and Privacy, vol. 5, no. 3, p. 59, Aug. 2025, doi: 10.3390/jcp5030059.

[25] S. Al-E'mari, M. Anbar, Y. Sanjalawe, S. Manickam, and I. Hasbullah, "Intrusion Detection Systems Using Blockchain Technology: A Review,

Issues and challenges," Computer Systems Science and Engineering, vol. 40, no. 1, pp. 87–112, Aug. 2021, doi: 10.32604/csse.2022.017941.

[26] M. K. I. Rahmani et al., "Security in optical Wireless Communication-Based vehicular ad hoc networks using signature and certificate revocation," Journal of Nanoelectronics and Optoelectronics, vol. 19, no. 1, pp. 112–119, Jan. 2024, doi: 10.1166/jno.2024.3544.

[27] E. F. Siddiqui, M. Haleem, S. F. Ahmad, A. Salhi, A. T. Zamani, and N. Varish, "A Multi-Layered AI-Driven cybersecurity architecture: integrating entropy analytics, fuzzy reasoning, game theory, and Multi-Agent reinforcement learning for adaptive threat defense," IEEE Access, vol. 13, pp. 170235–170257, Jan. 2025, doi: 10.1109/access.2025.3610526.

[28] B. Alabduallah, A. Banga, N. Iqbal, A. Ikram, and H. Diab, "Advancing cryptographic security with a new Delannoy-Derived Chaotic S-Box," IEEE Access, vol. 12, pp. 82926–82937, Jan. 2024, doi: 10.1109/access.2024.3410668.