International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.1 (2026) pp. 1205-1210
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

AI-Driven Network Security Architecture for Edge Computing

Naveen Kumar*

Independent Researcher, USA
* **Corresponding Author Email:** tyaginaveenk@gmail.com - **ORCID:** 0000-0002-5247-7772

Article Info:

DOI: 10.22399/ijcesen.4994
Received : 19 December 2025
Revised : 25 February 2026
Accepted : 01 March 2026

Keywords

Edge Computing Security,
Distributed Intelligence,
Zero Trust Architecture,
Automated Threat Detection,
Resource-Optimized Machine Learning

Abstract:

Edges pose a fundamental change to network security architecture. Edge computing moves computation, and thus exposure, to many physical locations, making customary enforcement points ineffective. Security architecture must consider the dynamic topology and variable connectivity, as well as the huge attack surface of billions of devices generating large volumes of data on the edge of the network. Distributed intelligence architectures implement a tiered security processing model. Security processing occurs on both edge and core compute, enabling real-time threat detection with low-overhead machine learning inference on the edge and centralized correlation analysis. APIs support programmable, policy-based automation of security across multi-vendor infrastructure, including zero-touch device provisioning and intent-based security operations. More advanced threat detection relies on behavioral baselining, predictive modeling, and ensemble learning to detect threats across multiple threat classes with a high level of precision and recall. Zero-trust architectures replace binary authentication with continual risk assessment and dynamic policy application. Zero-trust networks enforce micro-segmentation to restrict lateral movement and reduce an attack's blast radius. To extend advanced security capabilities to resource-constrained edge devices, model compression and energy-efficient inference at the edge are popular approaches to hardware acceleration. It can also considerably reduce the computational cost while maintaining detection performance with this integrated framework to address the security problems in distributed computing environments through smart automation, adaptive controls, and resource optimization.

1. The Evolution of Network Security Models in Distributed Computing

Enterprise network security architecture customarily has been based on a model of well-defined edges and centralized controls. Peripheral security zones rely on layered defenses that may include stateful firewalls, intrusion prevention systems, and demilitarized zones (DMZ architectures). An enterprise architecture's security components are tightly coupled with its network architecture. Customary perimeter-based architectures also assume that computational resources and critical data stores are hosted within a secure environment and not in untrusted external environments [1].Instead, edge computing does not attempt to reduce latency but instead masks latency by pushing cloud computation and services from the core network all the way out to the logical extreme of the network edge at industrial and commercial facilities, transportation and logistics, and telecom access network nodes. The number of connected IoT devices is expected to exceed 50 billion by 2025. In 2019, 45% of the data traffic was processed close to, or at, the edge of the network [1][2]. In 2019, the global data center IP traffic is expected to reach 10.4 zettabytes. The data generated by humans, machines, and things is expected to reach 500 zettabytes. The emergence of edge computing in computational topology has resulted in security outcome asymmetries as well. Limitations in computation and memory within edge devices, threats to their physical access and mobility, and their high data output, such as the Boeing 787 aircraft generating 5 Gbps or autonomous vehicles generating 1 Gbps, create security concerns.Besides the expansion of this perimeter, security at the edge has additional issues. Devices can frequently come and go, and their movement continuously changes the edge topology. Static security policies are not a good fit for the edge, and customary network segmentation

technologies, like VLANs, cannot be easily used for fluid edge deployments. Due to intermittent connectivity, networking and security decisions must be made based on local conditions.Threat modeling and risk assessment modes need to evolve from perimeter- to proximity-based, with the controls transitioning from periphery protection to per-device, per-application, and per-transaction enforcement. Emerging edge security models utilize multi-tiered threat detection and mitigation, enabling local decisions to be made while maintaining a coordinated global security posture on distributed three-tier IoT systems consisting of mobile end-devices, edge cloudlets, and a central cloud [1][2].

2: Distributed Intelligence Frameworks for Edge Security

Distributed intelligence architectures implement hierarchical processing models where the security functions are partitioned between edge layer and core layer computing resources according to acceptable latency, data sensitivity, and computational capabilities. Lightweight machine learning inference on the edge layer with quantized neural networks running on resource-constrained processors can lead to real-time threat detection, while more wide-ranging computations run at the core layer. In addition, by 2020, as many as 50 billion IoT devices will be connected to the Internet, generating a massive amount of security-relevant data that must be processed at the edge promptly [3].On the core layer, multiple telemetry signals originating from hundreds or thousands of distributed edge devices are aggregated and correlated, analyzed for long-term trends, and the model progress is repeated iteratively. This has led to the application of machine learning technologies like support vector machines, decision trees, or ensemble learning in the field of IoT security. For example, 60% of healthcare organizations implemented the Internet of Medical Things (IoMT) in 2015. The attack surface increased, and smart threat detection was required. [3] The core layer includes deep packet inspection, graph-based modeling of lateral movements in networks, and adversarial machine learning model retraining, which require GPU and high-memory nodes for implementing these computationally intensive processes.Behaviors at the edge layer are detected by estimating probabilistic baselines such as the distributions of packet sizes, inter-arrival time, finite state machine transitions of a given protocol flow, and graphs of data flows to their endpoints. Machine learning algorithms such as k-means clustering and principal component analysis are examples of unsupervised learning to detect statistical outliers. Experiments show that detection across attack types is possible. PHI has a black market value of $50, compared to $1.50 for credit card data. Protecting healthcare data in IoT thus has far-reaching implications [3].Federated learning frameworks develop threat intelligence in distributed edge environments subject to tight data locality and privacy constraints. Edge nodes train local machine learning models on site-specific network telemetry and only send model parameters or gradient updates to centralized aggregation servers, rather than sending abundant network traffic for collection and processing. Prior research demonstrated that the 2016 Google method can be used for federated learning of models on Android devices. Improvement includes training models accurately within the δ-accuracy loss bound when $|V\_FED - V\_SUM| < \delta$ [4]. The achieved global model is constructed from threat intelligence observed in heterogeneous edge environments and forms a continuously adaptive distributed intelligence system.

3: API-Orchestrated Security Automation

API-driven orchestration frameworks provide a programmable approach to defining, deploying, and enforcing security policy across edge-based distributed infrastructure. Modern edge security architectures leverage RESTful APIs or gRPC APIs to expose security capabilities as consumable services, enabling infrastructure-as-code-based approaches to protect the network. The performance of today's software-defined networking architectures is that distributed control plane systems can scale to a million queries per second and have an event processing latency of between 10 and 100 milliseconds to provide the baseline for real-time security orchestration in large networks [6].Automated provisioning workflows utilize declarative application programming interfaces to onboard edge devices without human intervention. Security orchestration platforms leverage network access control, certificate authorities, and identity providers to automatically provision authentication credentials, encryption keys, and initial security policies for devices based on classification and risk analysis. Distributed control architectures have been deployed to provide enterprise-scale automated security provisioning and have been shown to work at a global network state up to 1 terabyte with 99.99% availability [6].Intent-based security frameworks abstract low-level network security configurations behind high-level policy statements defined using natural language or policy template files. Security

orchestrators convert the intent statements into device-specific security configurations, then push the configurations to heterogeneous infrastructure, including firewalls, switches, wireless access points, and software-defined WAN (SD-WAN) gateways. Evaluation of the approach shows that distributed SDN platforms can provide an average of 45.2 milliseconds and a 99th percentile latency of 75.8 milliseconds to network events for realizing dynamic security policies in a distributed infrastructure [6].Infrastructure interoperability is a key architectural requirement of API-orchestrated security solutions in heterogeneous edge enterprise environments. Security APIs that can interoperate with software-defined networking controllers, network function virtualization orchestrators, and cloud management platforms enable threat response and security operations across different infrastructure domains. Path installation throughput testing achieves 18,832 paths per second, with a median end-to-end path latency of 53.1 ms, allowing network security policies to be deployed almost instantly across networks containing hundreds of switches and links [6]. If security tools detect threat behavior, security response workflows will call through to network infrastructure APIs to take action, for instance, VLAN reassignment, access control list modification, or port shutdown.

4: AI-Powered Threat Detection and Response Mechanisms

Predictive behavioral baselining creates statistical profiles that describe expected normal operating behavior for a workload at the edge. Edge workloads can be profiled based on fingerprints of network communications, resource utilization, and activity cycles. Machine learning can analyze telemetry data over several operating cycles and can develop probabilistic models of expected behavior. For example, the UNSW-NB15 dataset was created from 2,218,761 records of normal traffic captured over a period of 31 hours of network surveillance [8].Predictive threat modeling uses supervised and semi-supervised machine learning to identify attack precursors and early indicators of compromise before an attack. Ensemble learning combines the predictions of multiple detection algorithms, including random forests, decision trees, and support vector machines, in order to identify attack patterns within the sequences of network traffic. Random Forests have been shown to reach an accuracy of 97% on DoS and 76% on Probe for the KDD 1999 dataset, while Naïve Bayes classifiers reach 96% and respectively [7].Real-time anomaly scoring algorithms continuously evaluate risks of the edge devices and the workloads they handle for detection. The data is represented as 49 multi-dimensional feature vectors, which represent network behavior, system characteristics, and other contextual information. The UNSW-NB15 dataset includes samples of network traffic classified into nine attack families, including Fuzzers (24,246), Analysis (2,677), Backdoors (2,329), DoS (16,353), Exploits (44,525), Generic (215,481), Reconnaissance (13,987), Shellcode (1,511), and Worms (174) [8]. These scoring systems are based on a sliding window analysis of sequences from 100 connections to determine the risk level based on malicious behavior.Autonomous response loops implement closed-loop security automation schemes based on observe-orient-decide-act cycles (inspired by military command and control concepts). The decisions are based on ensemble approaches that take advantage of different classifiers. For example, an accuracy rate of 99.71% for normal traffic, 99.85% for Probe attacks, 99.97% for DoS, and 100% for R2L attacks can be achieved with a combination of ANN and SVM [7]. Action phases invoke orchestration APIs to execute containment actions. The DISCLOSURE system has been demonstrated in practice to yield effective results by employing Random Forest classifiers on real-world NetFlow traffic data. Rates of 65% true positives with 1% false positives [7] have been achieved. DISCLOSURE enables fast response to threats on the distributed edge infrastructure.

5: Implement Zero Trust at Network Edges

Continuous trust evaluation is an essential zero-trust capability that replaces the binary decision of authenticating users or devices with continuous risk assessment based on contextual signals throughout the session lifecycle. Zero-trust architectures use algorithms for continuously assigning trust scores based on parameters such as device posture, user behavior, or location and time. Continuous authentication allows organizations to quickly detect stolen credentials through behavioral anomaly detection tools that look for meaningful differences from a baseline pattern, considered a more secure alternative than session-based authentication standards.Dynamic identity validation extends credential validation by adding multiple contextual factors (e.g., device integrity measurements, firmware versions, certificate chain validation, and hardware attestation). Trust brokers rely on telemetry from trusted platform modules (TPM), mobile device management systems, and endpoint detection and response platforms to assess the overall security posture of devices. This can be

useful to reduce issues with single-factor authentication with a password, which can be susceptible to credential stuffing, phishing, and brute force attacks against default credentials.Adaptive enforcement employs a graduated response to trust score declines by increasing the level of restriction associated with the access request whenever a trust score threshold is breached. Examples of improved restrictions include step-up authentication, bandwidth throttling, micro-segmentation, and session termination. These mechanisms can be used to defend against unauthorized lateral movement, without the performance penalty of inappropriately restricting access for legitimate users who are otherwise within their trust boundaries.Micro-segmentation strategies segment and isolate data center workloads into separate security zones, establishing granular security policies for each workload or device instead of the network subnet. Software-defined perimeters create encrypted overlay networks to ensure zero-trust, point-to-point communications between only trusted endpoints, while hiding resources behind a virtual perimeter from untrusted endpoints. Micro-segmentation also limits the blast radius of a successful attack to just the zone that was breached instead of the entire flat network. Assuming breach as the security model, micro-segmentation assumes breaches are inevitable and that every zone might already be compromised. Every time an attacker crosses the boundary of a zone, they must re-authenticate and re-authorize before accessing other network resources.

6: Resource-Optimized Security for Constrained Edge Devices

Security features on resource-constrained edge devices using hardware acceleration technologies include neural processing units, tensor processing units, and smart network interface cards that have programmable packet processing pipelines. Security accelerators can offload cryptography, pattern matching, and machine learning inference from general-purpose processors, thereby enabling line-rate security processing at multi-gigabit throughput rates. Deep learning architectures such as CNNs currently outperform all other approaches on image classification tasks, achieving much lower error rates on a dataset consisting of a million images across 1000 classes compared to customary approaches [11].Compression methods based on pruning, quantization, or knowledge distillation reduce parameters and computations to allow the model to run on very low-power devices while maintaining detection accuracy. In Deep Compression, it is shown that pruning reduces parameters by a factor ranging from 9 to 13 for state-of-the-art models. It is also demonstrated that AlexNet is compressed from 240MB to 6.9MB (35×) and VGG-16 from 552MB to 11.3MB (49×) with no accuracy loss. Their network can be compressed by a factor of 35 to 49, using a three-stage pipeline of pruning, trained quantization, and Huffman coding, without a loss in predictive accuracy [12].Quantization is applied to the model parameters, which are stored as floating-point numbers. Each convolutional layer is quantized to 8 bits (256 shared weights), and each fully-connected layer to 5 bits (32 shared weights), preserving accuracy [12]. This shrinks the model considerably, as pruning reduces the number of connections by a factor of 9 to 13, while quantization reduces the number of bits per connection from 32 to 5 [12]. These compressed models can be executed faster on a range of processors, providing a 3× to 4× speedup per layer and 3× to 7× better energy efficiency for real-time inference workloads [12].Low-energy inference focuses on optimizing the processing of neural networks on battery-constrained edge devices. Benchmarks for sparse models indicate that sparse networks are much more power-efficient than dense networks. On a CPU, the energy costs of sparse computing are 42W for sparse and 84W for dense matrix-matrix multiplication for the same problem size [12]. Line-rate packet inspection, which requires parallelized processing pipelines and zero-copy packet handling, as well as deep convolutional networks with 10-20 levels of representation (and hundreds of millions of weights), enables complex patterns for security threat detection [11]. The implementations that fit entirely in on-chip SRAM cache prevent the power consumption of energy-hungry DRAM: 640 picojoules per 32-bit operation for DRAM compared to just 5 picojoules for SRAM [12].

Table 1: *Healthcare IoT Adoption and Data Security Economics [3]*

Parameter	Value	Unit
Healthcare Organizations with IoMT	60%	Percentage (2015)
Personal Health Information (PHI) Black Market Value	$50	USD per record
Credit Card Information Black Market Value	$1.50	USD per record

Table 2: *Path Installation Performance Characteristics [6]*

Performance Indicator	Measurement
Path Installation Throughput	18,832 paths/second
Median End-to-End Path Latency	53.1 milliseconds
Network Scale	Hundreds of switches and links

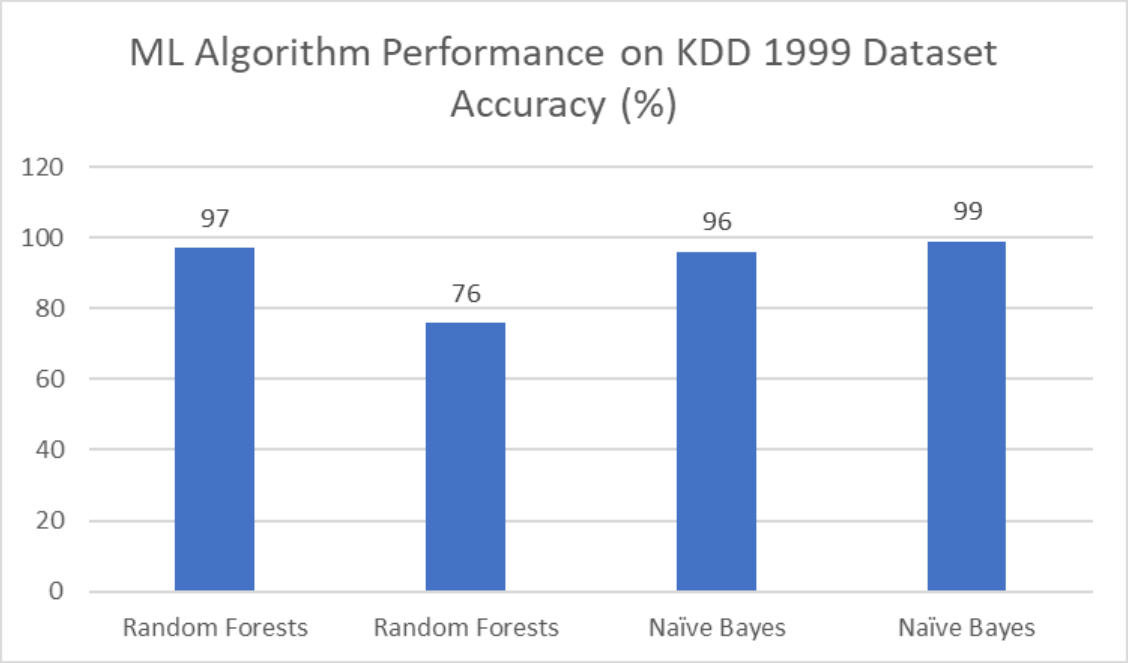

Figure 1: *Attack Detection Accuracy by Algorithm Type [7]*

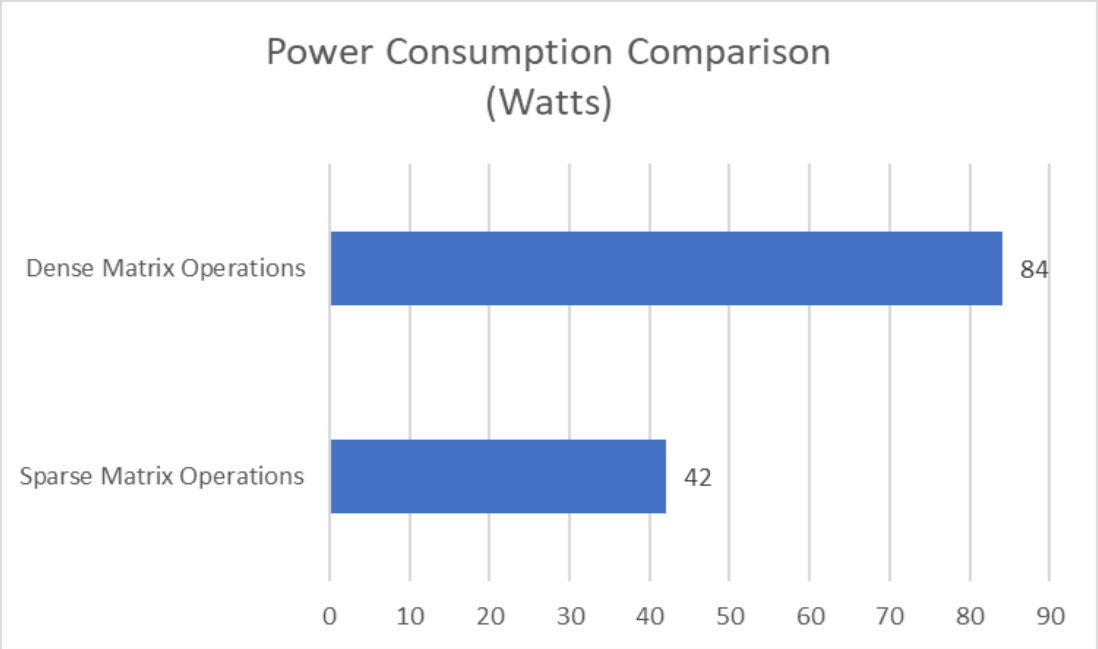

Figure 2: *Power Consumption Comparison [12]*

7. Conclusions

The emergence of edge computing architectures has required a shift in security models from the classical perimeter-based approach to a decentralized, responsive security model that can operate across large-scale distributed systems. The combination of artificial intelligence, automated orchestration, and zero-trust techniques enables security systems to operate effectively across resource-constrained edge environments. With distributed intelligence, workload is cleverly distributed to device aggregators for lightweight threat detection, while correlation and model

training are processed in the central infrastructure. APIs eliminate configuration overhead and bottlenecks, allowing organizations to orchestrate faster policy application and a coordinated response to threats across infrastructure silos. More advanced machine learning techniques exist to look for behavioral patterns and predictions of attacks, which are accurate enough for a self-defending system. Zero-trust architectures are also a way of dealing with these issues, as they use continuous validation and micro-segmentation to reduce the attack surface and limit horizontal movement. XDR and NDR technology developments, such as model compression and hardware acceleration, enable advanced security applications to run efficiently on resource-constrained devices. This allows them to provide the strong protection of customary tools but with the performance and efficiency of distributed edge computing infrastructure to both defend against evolving threats and meet modern enterprise expectations.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Jianli Pan, James McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications," ResearchGate, 2017. [Online]. Available: https://www.researchgate.net/profile/Jianli-Pan/publication/320723809

[2] Weisong Shi, et al., "Edge computing: Vision and challenges," IEEE, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7488250

[3] Mohammed Ali Al-Garadi et al., "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," arxiv, 2020. [Online]. Available: https://arxiv.org/pdf/1807.11023

[4] Qiang Yang et al., "Federated machine learning: Concept and applications," ACM Digital Library, 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3298981

[5] Diego Kreutz et al., "Software-Defined Networking: A Comprehensive Survey," arxiv, 2014. Available: https://arxiv.org/pdf/1406.0440

[6] Pankaj Berde et al., "ONOS: towards an open, distributed SDN OS," ACM Digital Library, 2014. https://dl.acm.org/doi/pdf/10.1145/2620728.2620744

[7] Anna L. Buczak and Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 18, NO. 2, SECOND QUARTER 2016. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7307098

[8] Nour Moustafa and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," University of New South Wales at the Australian Defence Force Academy, [Online]. Available: https://www.researchgate.net/profile/Nour-Moustafa/publication/287330529

[9] Pacharee Phiayura and Songpon Teerakanok, "A Comprehensive Framework for Migrating to Zero Trust Architecture." IEEEXplore, 2023. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10052642

[10] Elisa Bertino, Nayeem Islam, "Botnets and Internet of Things Security," CyberTrust. [Online]. Available:https://www.researchgate.net/profile/Elisa-Bertino/publication/313464793

[11] Yann Lecun et al., "Deep learning," HAL Open Science, 2023. https://hal.science/hal-04206682/document

[12] Song Han, "Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding," ICLR, 2016. https://arxiv.org/pdf/1510.00149