ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 3 Issue 1 February 2025 / Page No: 126-131
Paper Id: IJACT-V3I1P111 / Doi: 10.56472/25838628/IJACT-V3I1P111

Original Article

Hybrid Edge AI and Centralized Processing for IoT: A Scalable, Secure Framework for Real-Time Manufacturing Analytics

Shankar Narayanan SGS
Principal Architect, Microsoft, USA.

Received Date: 02 December 2024 Revised Date: 12 January 2025 Accepted Date: 01 February 2025

Abstract: *The convergence of Industry 4.0 and IoT has fueled a need for low-latency decision-making at the edge of production systems, while also leveraging centralized resources for global analytics and security. This paper proposes a Hybrid Edge-Central (HEC) Architecture that partitions Graph Convolutional Networks (GCNs) across on-premises devices and cloud data centers, supplemented by Generative Adversarial Networks (GANs) for adversarial testing. A hardware security module (HSM) on each edge device protects API keys and ensures tamper-resistant cryptographic operations. We demonstrate how edge AI reduces latency and bandwidth usage (~60%), while cloud-side GCN layers capture cross-device correlations for predictive maintenance and anomaly detection. A case study in an automotive manufacturing plant showcases significant reductions in unplanned downtime (~30%) and improved security via GAN-driven stress testing. Our results highlight a robust, scalable solution for Industry 4.0 transformations requiring data security, real-time insights, and system-wide optimization.*

Keywords: *IoT, Artificial Intelligence, Framework, Manufacturing Analytics.*

I. INTRODUCTION AND MOTIVATION

A. The Need for Edge-Central Collaboration

In modern **IoT** environments—particularly in large-scale manufacturing—massive amounts of sensor data must be processed with minimal latency**.** Edge AI has emerged to handle tasks like anomaly detection or predictive maintenance directly on the factory floor [Shi et al., 2020]. However, broader insights (e.g., cross-line correlations, advanced machine learning pipelines) often demand the compute power and data integration capabilities of centralized or cloud-based systems. Balancing these two paradigms is essential for Industry 4.0 success, where real-time responsiveness and big-picture optimization must coexist.

B. Security Imperatives

As operations become increasingly digitized, data security and hardware-based key management become critical. Traditional software-based key storage is vulnerable to device tampering, especially in distributed environments. Coupling on-device HSM modules with secure communication channels (e.g., mutual TLS) helps protect API keys and sensitive credentials from unauthorized access, a critical concern in mission-critical production environments [NIST, 2018].

C. Paper Contributions

This paper presents a Hybrid Edge-Central (HEC) Architecture that unites low-latency edge inference with centralized GCN analytics and GAN-based adversarial testing. Key highlights include:

- GCN Partitioning**:** Splitting model layers between edge devices and a central node for both speed and global data correlation.
- Hardware Security Modules (HSMs)**:** Ensuring tamper-resistant API key storage on edge devices.
- GAN-Driven Security Testing**:** Synthetic anomalies and attacks to probe and enhance system resilience.
- Manufacturing Case Study**:** Demonstrating significant improvements in uptime, anomaly detection, and secure operations.

II. RELATED WORK AND BACKGROUND

A. Edge AI in Industry 4.0

Previous research indicates edge-based analytics can substantially reduce latency and bandwidth overhead [Lane et al., 2019]. However, many systems still rely on either fully edge-based or fully cloud-based approaches, missing the nuanced benefits of hybrid architectures.

B. Graph Convolutional Networks for IoT

GCNs model relationships in graph-structured data, making them suitable for industrial settings where machine connectivity, sensor correlation, and operational interdependencies form complex graphs [Wu et al., 2020]. Yet, partitioning GCN layers between edge and cloud is still an emerging concept in production pipelines.

C. GAN-Based Security and Synthetic Data Generation

Generative Adversarial Networks have proven effective for synthesizing realistic data (e.g., images, sensor patterns) to test AI models under adversarial conditions [Zhang et al., 2021]. Incorporating GAN-driven tests within the IoT pipeline is a novel approach to ensuring robust anomaly detection and proactive security.

D. Hardware Security Modules (HSMs) for Key Management

HSMs or secure enclaves (e.g., TPM) are specialized hardware that protect cryptographic keys from unauthorized access. In edge contexts, HSMs prevent attackers from extracting credentials, even with physical device access.

III. HYBRID EDGE-CENTRAL (HEC) ARCHITECTURE

A. Overview

The HEC Architecture consists of multiple layers:

- IoT Edge Devices: Equipped with lightweight AI models (GCNs or CNNs) and HSM modules for secure cryptographic operations.
- Central Data Center: Hosting advanced analytics (final GCN layers, GAN models) and a data lake for historical logs.
- Secure Communication: Mutual TLS ensures encrypted transmission, with API keys stored securely in each device's HSM.

B. System Diagram

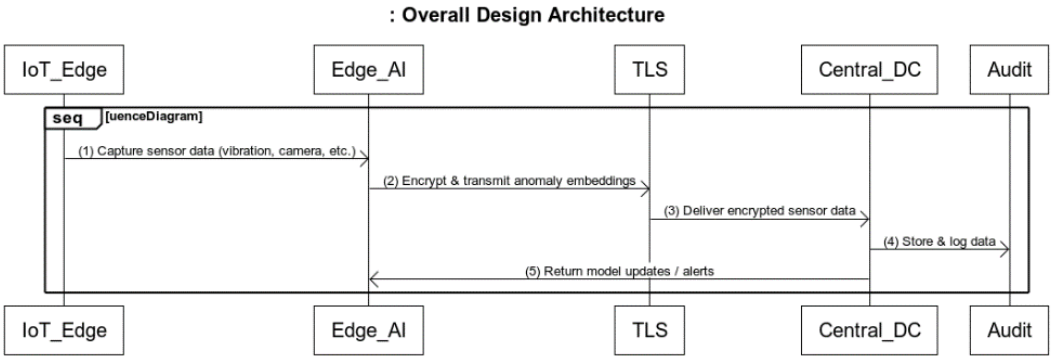

Figure 1: Overall Design Architecture

- IoT Edge Devices collect raw sensor data and pass it to Local Edge AI.
- Edge AI detects anomalies and encrypts critical data using HSM-managed keys.
- TLS channel ensures secure data delivery to the Central Data Center.
- Central Data Center logs the data in Secure Storage (audit logs).
- Feedback loop: The central system sends updated model weights or alerts back to the edge.

C. GCN Partition Diagram

To illustrate how Graph Convolutional Network layers are split between edge and central servers, consider the following simplified representation:

- Edge GCN (Layer 1): Performs initial local feature extraction (quick anomaly detection).
- Central GCN (Layers 2, 3): Performs deeper cross-device correlation and final decision-making.
- The arrow (A) represents sending partial embeddings from edge to central; arrow (B) represents the central's feedback to refine edge thresholds or weights.

a) Key Insight:

- Layer1 at the edge quickly extracts local features (e.g., vibration anomalies).
- Layer2 and Layer3 at the central server merge embeddings from all edges, capturing cross-line correlations (e.g., similar faults across multiple robots).

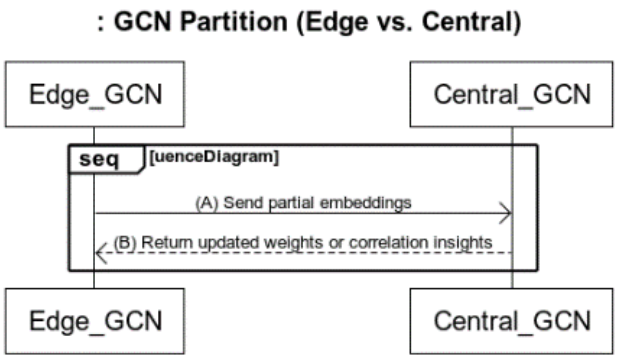

Figure 2: GCN Partition

IV. SECURITY CONSIDERATIONS

A. HSM Implementation at the Edge

Each edge device integrates an HSM that manages key generation, storage, and cryptographic functions. This approach protects API keys and ensures secure boot and firmware integrity:

- Secure Boot: Only signed and trusted firmware can execute, thwarting OS-level tampering.
- Encrypted Communications: TLS channels are established using keys that never leave the HSM.
- Credential Isolation: Even if an attacker gains physical access, extracting private keys is highly improbable.

B. Multi-Layer Encryption and Authentication

- Mutual TLS**:** Certificates uniquely identify each device, authenticated by a Certificate Authority within the central system.
- End-to-End Encryption**:** Sensor data or embeddings are encrypted before transmission, preventing eavesdropping on the factory floor or public networks.
- Role-Based Access Control (RBAC)**:** Operators and processes only have privileges aligned with their roles, mitigating insider threats.

C. Intrusion Detection

- Edge IDS**:** Lightweight intrusion detection runs on each device, detecting suspicious traffic or sensor manipulations.
- Central SIEM**:** A Security Information and Event Management system aggregates logs across devices, correlating potential threats for real-time alerting.

V. DETAILED WORKFLOW

A. Core Steps

a) Data Capture

- Edge devices continuously ingest vibration, temperature, and camera feeds from production machinery.

b) Edge Inference & Encryption

- A lightweight GCN/CNN inspects incoming data for immediate anomalies.
- If a threshold is exceeded, data or embeddings are encrypted using HSM-managed keys, then sent to the central system.

c) Central GCN Layers

- Partial embeddings from various edge nodes feed into final GCN layers in the central environment.
- Cross-device correlation identifies system-wide anomalies, reducing false positives and uncovering patterns missed at the edge.

d) GAN-Based Testing

- GANs generate synthetic anomalies or adversarial sensor data.
- This data tests both edge and central detection pipelines, ensuring robust performance against spoofing or novel failures.

e) Feedback Loop

- The central system returns updated model weights or configuration thresholds.
- Secure key management ensures only authenticated updates reach the edge devices.

f) Historical Logging & Compliance

- All events are stored in a Data Lake, with logs secured via ISO 27001–compliant processes.
- A parallel ledger or SIEM system maintains a tamper-evident record of security events and system alerts.

B. Data Flow Diagram

- Sensors feed continuous data (vibration, temperature, etc.) to Edge AI.
- Edge AI checks for anomalies locally using a lightweight GCN/CNN.
- If an anomaly is found, partial embeddings are encrypted (via HSM/TLS) and sent to the Central GCN for deeper analysis.
- GAN module generates synthetic or adversarial data, pushing it to an edge instance for stress-testing detection thresholds (ensuring robustness).
- Central GCN merges insights from multiple edges and returns updated model parameters or thresholds to the edge.

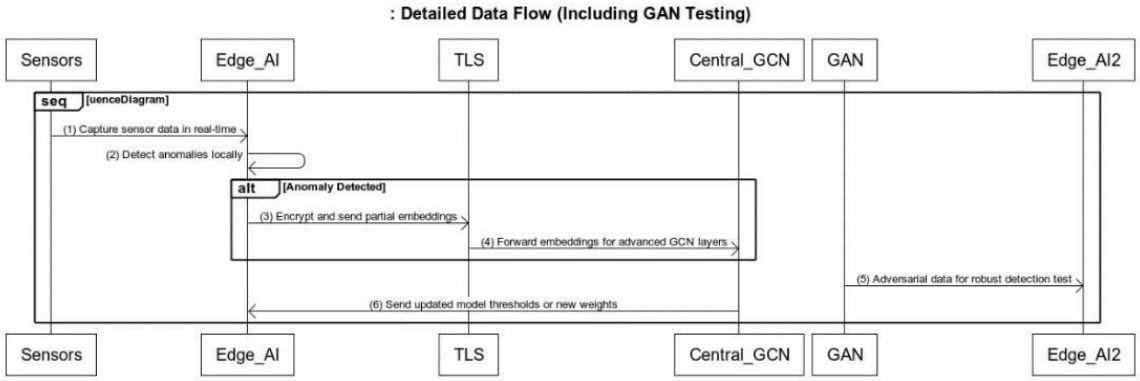

Figure 3: Data Flow Diagram

C. JSON Message Examples

Below are sample JSON structures that might be exchanged between edge devices and the central system. These examples illustrate typical data payloads, including encryption tags, sensor readings, and model updates.

a) Edge-to-Central (Sensor Embeddings)


```
{
"device_id": "robot-arm-23",
"timestamp": "2025-02-10T10:20:35Z",
"encrypted_payload": {
"ciphertext": "QWxpY2U6IFRoaXMgZGFoYSBpcyBlbmN...",
"encryption_algorithm": "AES-256-GCM",
"key_id": "hsm_key_robot_arm_23",
"iv": "aHRocDovL21va2VpLmZy",
"auth_tag": "ZG91YmxlIA=="
},
"model_data": {
"partial_embedding": [0.0123, 0.9584, 0.0032, 0.5579],
"local_anomaly_score": 0.82,
"confidence": 0.90
}
}
```


- device_id: Unique identifier for the edge device.
- encrypted_payload: Contains sensor data or additional metadata, encrypted using HSM-managed keys (for demonstration, ciphertext is truncated).
- model_data: Key inference results, including a partial embedding for the central GCN to process.

b) Central-to-Edge (Model Update)


```
{
 "device_id": "robot-arm-23",
 "timestamp": "2025-02-10T10:21:05Z",
 "new_weights": "base64_encoded_weights_here==",
 "version": "model_gcn_v2.1",
  "update_instructions": "Apply these weights to Layer 1. Retrain local thresholds to reduce false positives.",
 "signature": "RSA-SHA256:Kkj++DFabc123..."
}
```


- new_weights: Base64-encoded model parameters for the local GCN layer.
- version: Model version tracking for deployment and rollback.
- signature: An HSM-backed digital signature verifying the authenticity of the update.

c) GAN-Generated Test Event (Central-to-Edge)


```
{
 "device_id": "robot-arm-23",
 "timestamp": "2025-02-10T10:22:10Z",
 "test_scenario_id": "gan_adversarial_vibration_01",
                                        "synthetic_data": "base64_of_synthetic_vibration_pattern==",
  "description": "Testing anomaly detection with slightly elevated amplitude and random noise injection.",
 "priority": "high"
}
```


- synthetic_data: Generated by the central GAN to simulate a new type of fault or adversarial attack, pushing it to the edge device for stress-testing local detection algorithms.

VI. CASE STUDY: AUTOMOTIVE COMPONENT MANUFACTURING

To validate the architecture, we implemented HEC in a mid-sized automotive component plant.

A. Deployment Setup

a) Edge Devices:

- ~200 industrial robots, each fitted with an NVIDIA Jetson or similar AI accelerator.
- Integrated hardware security modules (HSMs) for API key management.

b) Sensors:

- 500+ streams of vibration, torque, thermal imaging, and camera data.

c) Central Infrastructure:

- On-premises GPU cluster running PyTorch for GCN/GAN training, orchestrated by a Kubernetes-based system.

B. Inputs/Outputs of Models

a) Edge GCN/CNN:

- Input: Local sensor data windows (vibration time-series, machine status).
- Output**:** Anomaly score (0–1) + confidence interval.

b) Central GCN:

- Input: Embeddings from multiple edge devices.
- Output**:** Global fault risk score + recommended maintenance actions.

c) GAN:

- Input: Historical sensor patterns and real-world fault logs.
- Output: Synthetic anomalies or adversarial signals to test detection boundaries.

C. Results

a) Reduced Downtime:

- 30% fewer unplanned failures due to early detection.

b) Bandwidth Optimization:

- ~60% lower data transfer by sending only embeddings or alerts vs. full raw sensor streams.

c) Security Hardening:

- Red-team exercises failed to extract credentials from the HSM.
- GAN testing exposed minor edge threshold settings that were subsequently fine-tuned, enhancing overall system resilience.

VII. RESULTS AND DISCUSSION

A. Key Observations:

- Balancing Edge & Cloud**:** By partitioning GCN layers, devices maintain real-time responsiveness, while global analytics benefit from aggregated data.
- Security Gains**:** HSM-based key storage significantly reduces credential theft risks, and **GAN** stress tests highlight vulnerabilities early.
- Scalability**:** The system seamlessly scales to hundreds of devices, supporting near real-time anomaly detection.

B. Challenges:

- Complex Deployment**:** Requires specialized hardware and robust DevSecOps practices.
- Adaptive Partitioning**:** Deciding how many GCN layers remain at the edge vs. central often requires iterative experimentation based on network constraints and device capabilities.

VIII. CONCLUSION AND FUTURE DIRECTIONS

We presented a Hybrid Edge-Central (HEC) Architecture integrating partitioned GCNs, GAN-based adversarial testing, and hardware security modules for key management. Deployed in an automotive manufacturing scenario, the architecture demonstrated significant reduction in unplanned downtime, optimization of bandwidth, and enhanced security. Key areas for future exploration include:

- Federated Learning: Further minimizing data movement by exchanging model updates instead of raw data.
- Adaptive Layer Partitioning: Dynamically shifting inference layers between edge and central based on real-time latency or bandwidth conditions.
- Extended Industry Use Cases: Applying this robust design to smart cities, telemedicine, or aerospace where real-time performance and security are paramount.

This paper offers both an academic perspective and a practical blueprint for Industry 4.0 stakeholders seeking a secure, scalable, and intelligent IoT infrastructure.

IX. REFERENCES

[1] ISO 27001 (2013). *Information technology — Security techniques — Information security management systems — Requirements.* International Organization for Standardization.

[2] Lane, N. D., Bhattacharya, S., Georgiev, P., et al. (2019). *DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices*. In *Proceedings of MobiSys*.

[3] NIST (2018). *Framework for Improving Critical Infrastructure Cybersecurity.* National Institute of Standards and Technology.

[4] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2020). *Edge Computing: Vision and Challenges*. IEEE Internet of Things Journal, 3(5), 637–646.

[5] Wu, J., Lee, J., & Zhao, W. (2020). *Fault Detection and Diagnosis in Smart Manufacturing Using GCN*. IEEE Transactions on Industrial Informatics, 16(6), 3705–3716.

[6] Zhang, J., Zhang, H., & Tao, C. (2021). *Adversarial Testing of Autonomous Driving via GAN-Generated Scenarios*. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4234–4243.