SCCTS Transactions on Reconfigurable Computing, ISSN: 3049-1533 Vol. 3, No. 2, 2026 (pp. 79-89)
RESEARCH ARTICLE ECEJOURNALS.IN

Secure and Scalable Runtime Reconfiguration Framework for FPGA-Based Edge AI in Industrial IoT Systems

Zafar Khan[1], Felix G. Nymana[2]

[1]Department of Accounting and Finance, Eastern Michigan University, USA
[2]Helsinki School of Cognitive Design, Finland.

Keywords:
Runtime Reconfiguration,
Secure FPGA Bitstream,
Edge AI, Industrial IoT (IIoT),
Partial Reconfiguration,
Hardware Security,
Scalable Edge Deployment,
AI Accelerator,
Trusted Execution,
Reconfigurable Computing

Author's Email:
zkhan@emich.edu

DOI: 10.31838/RCC/03.02.09

Received : 09.01.2026
Revised : 10.03.2026
Accepted : 07.04.2026

Abstract

The expansion of IIoT systems requires the intelligence of real-time decision-making at the edge, where real-time performance, security, and power are decisive parameters. FPGAs provide an excellent opportunity to implement AI workloads in such environments thanks to their flexibility in terms of programmability and the presence of a parallel processing horizon. Nevertheless, the most traditional FPGA-based solutions can be vulnerable to lack secure and scalable runtime reconfiguration, which reduces the flexibility in dynamic IIoT usage. In this paper, a new framework is proposed that facilitates secure and scalable, suitably with data change, execution of AI on edge field-programmable gate arrays (FPGAs) in IIoT networks. The framework combines the lightweight cryptographic engine designed to authenticate bitstreams, the programming dynamic reconfiguration manager, a part-task swapping, and the scaling orchestrator, which addresses the distributed deployment at the edge. Representative experimental evaluations on realistic workloads in AI, i.e., anomaly detection, and object recognition, validate the effectiveness of the proposed system to achieve low-latency inference, low-energy overhead, and resistant to bitstream tampering. The framework provides a basis of robust and resilient, flexible and reliable AI implementation in mission-critical IIoT use cases.

How to cite this article: Khan Z, Nymana FG(2026). Secure and Scalable Runtime Reconfiguration Framework for FPGA-Based Edge AI in Industrial IoT Systems. SCCTS Transactions on Reconfigurable Computing, Vol. 3, No. 2, 2026, 79-89

Introduction

Industrial Internet of Things (IIoT) is transforming manufacturing and automation industries with its capability to connect computers, advanced computational intelligence and communication systems to physical devices, sensors, controllers and actuators. This meeting has facilitated the implementation of predictive maintenance, prediction of anomaly detection, adaptive control and automated decision-making at factory levels, power grids and production facilities. Since these systems are characterized by generating and processing large data at the network edge, there is an increased request of low-latency, energy, and secure processing systems that can suit different sets of workloads and operation limitations.

Edge Artificial Intelligence (Edge AI) has become a major facilitator of intelligent IIoT systems as it enables local data inference and decision-making and, therefore, minimizes cloud reliance, network latency

and possible privacy threats. The Field-Programmable Gate Arrays (FPGAs) are one of the hardware platforms currently offered to Edge AI, which have a significant potential to integrate flexibility, parallelism, and energy efficiency characteristics. They also feature reconfigurable logic fabric, which allows on-the-fly customization of hardware resources to run AI workloads according to their performance and power needs like convolutional neural networks (CNNs), the support machine (SVMs), and anomaly detection algorithms.

Nevertheless, when it comes to the deployment of AI models on FPGAs in an IIoT setting, there are some key issues that need to be considered. First is the lack of flexibility of the run time because the conventional FPGA systems are hard-wired, and adapting them to run to dynamic workloads or operational adjustments can be not easily done without reprogramming the whole system. Partial Reconfiguration (PR) may appear as a possible means of solution but the current methods frequently do not have effective and autonomous run time orchestration techniques. Secondly, it should be much more worrying about security, particularly in industrial applications over-the-air bitstream updates are all the rage. Such systems are subject to bitstream-level attacks, unauthorized reconfigurations and theft of intellectual property and require high-quality cryptography and authentication through means of hardware. Finally, scalability is another bottleneck in distributed IIoT networks since a secure and coordinated reconfiguration framework to distribute a variety of AI tasks across a large number of edge FPGAs is needed, with minimum disruption to the system, resource contention, and communication overhead. These interdependent issues are critical to ensuring adaptive, secure, and scaleable implementation of Edge AI in industrial settings that are mission-critical.

This paper presents the Secure and Scalable Runtime Reconfiguration Framework for FPGA-based Edge AI solutions in IIoT systems in order to counter these limitations. The framework proposed would reconfigure accelerator AI devices at run-time by performing a partial and dynamic update of the hardware supporting said AI device and make tasks switch without having to interrupt the entire system. It offers strong security with the use of lightweight cryptographic primitives and physical unclonable function (PUF)-based hardware anchors that can offer bitstream authentication of protection against tamper or unauthorized access. Also, the architecture can be deployed in a scalable fashion into distributed edge clusters due to the addition of a unified control plane enabling secure task migration, dynamic workload profiling, and coordinated AI model switching, thereby addressing the performance, security and adaptability requirements of contemporary industry settings.

There are fourfold contributions that can be considered the important elements of the paper. First, it introduces the design and realization (a conceptual lightweight runtime reconfiguration manager) that makes it easy to perform partial corner-swapping of AI kernels on FPGAs without affecting overall system functionality. Second, it adds a secure bitstream authentication that is weaved together with AES-GCM encryption and SHA-256 hash, enhanced to hardware level by a PUF-based identity to gain hardware-level trust and provide against illegal reconfiguration. Third, the paper introduces a scalable edge deployment model, which is able to realize a coordinated reconfiguration management within a distributed IIoT deployments with multiple FPGA nodes and nodes. Lastly, the proposed design is experimentsally confirmed in an industrial environment utilizing an FPGA platform, Zynq UltraScale+, and simulates industrial work to show a tremendous improvement in performance, power efficiency, and resilience to security attacks.

Related Work

The recent rise in demand of adaptive and efficient computing within Industrial Internet of Things (IIoT) settings has led to an upsurge of substantial research on FPGA-based runtime reconfigurability, as well as the safe deployment of Edge AI. The huge opportunities in the fields of dynamic partial reconfiguration, hardware security, and FPGA-powered edge intelligence are also discussed here alongside the gaps which are worth filling with this paper.

Runtime Reconfiguration in FPGAs

Dynamic Partial Reconfiguration (DPR) is the ability to reconfigure parts of an FPGA without halting the process of the complete system. This is important in edge environments, which require adaptive

redistribution of computing resources to be able to scale for or react to different workloads. Stott et al.[1] have given a detailed study of the DPR techniques, tools, and architectural models. PR is also simplified by using the Vivado Design Suite[2] of Xilinx, with the toolchain-level assistance to allow developers to design with targeting reconfigurable systems effective. Although these tools facilitate a runtime flexibilities, the available reconfiguration models are frequently not designed with AI-specific work and intelligence of runtime precondition to deploy on the edge.

Security in Reconfigurable FPGA Systems

In mission-critical and field-updatable hardware deployments, security is of paramount concern because hardware reconfiguration is appealing when trying to create secure systems. Tehranipoor and Koushanfar,[3] categorized the different threats of hardware Trojan and provided descriptions of the approaches employed to detect them, and these are especially pertinent when bitstream updates are conducted remotely. Sadeghi et al.[4] designed a cryptographic key management secure architecture using FPGA that can take advantage of symmetric encryption techniques. Nonetheless, this makes the static key storage vulnerable in physically-accessible environments. To deal with this, Guajardo et al. [5] proposed FPGA native Physical Unclonable Functions (PUFs) to create hardware-specific secrets, which provides an inexpensive, tamper-resilient security root in embedded systems.

FPGA-Based Edge AI in IIoT

The integration of AI workloads with reconfigurable computing has shown promising results in IIoT applications. For instance, Mittal and Vetter[6] explored GPU and FPGA energy-efficiency optimization techniques for edge deployment. Jha et al.[7] focused on AI-HW co-design on FPGAs for real-time industrial analytics, showcasing how hardware-adaptive inference engines can improve response time in Industry 4.0 systems. Recent studies have extended this trend toward renewable energy and automotive domains, including the use of embedded systems for autonomous vehicle control,[9] precision agriculture,[13] and power electronics in EV charging.[12] Moreover, AI-powered edge architectures have been applied in domains such as energy grid optimization[8] and vehicular theft detection,[11] illustrating the versatility of edge-AI in diverse industrial contexts.

Gaps in Current Approaches

Despite notable advancements, key challenges persist. Existing frameworks generally lack secure, scalable orchestration for runtime reconfiguration in edge networks. Most PR flows do not include integrated authentication mechanisms, making them vulnerable during over-the-air updates. In distributed IIoT deployments, scalable task migration and secure bitstream distribution remain largely unexplored. This paper addresses these limitations by proposing a unified framework that combines lightweight security, runtime flexibility, and deployment scalability for FPGA-based Edge AI systems.

System Architecture

The introduction of the proposed system architecture is proposed to realize the secure and scalable reconfiguration of the FPGA-based AI accelerator at run time in distributed Industrial IoT (IIoT) infrastructures. The architecture consists of five major modules as shown in Figure 1 and they include: The Edge FPGA Node, AI Kernel Reconfiguration Manager Hardware Trust Anchor, the Secure Bitstream Loader, and the AI Hardware Trust Anchor. All the elements are essential in the pursuit of dynamic flexibility, safe job implementation, and synchronized edge intelligence.

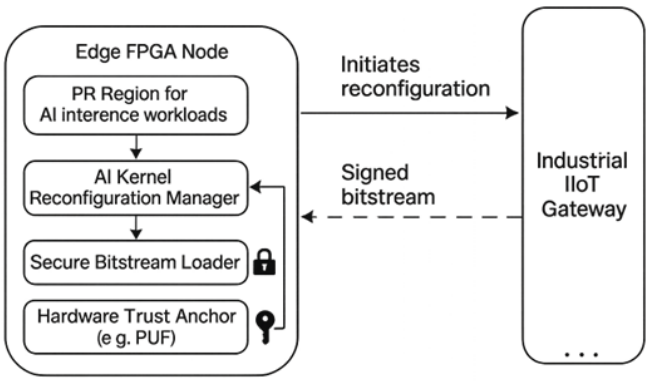

Fig. 1: Secure Runtime Reconfiguration Framework for Edge AI on FPGAs

The physical compute unit for performing AI inference works is the Edge FPGA Node. The reconfigurable logic

regions (Partial Reconfiguration Regions, or PRRs) installed with this node enable on-demand exchange of the requested part of the AI, including convolutional layers, anomaly detectors, or decision trees, without the stopping of the system operation. The node has a very tight real-time requirement and communicates directly with IIoT sensors and actuators to collect data and actuate. The core of the adaptability of the runtime includes the AI Kernel Reconfiguration Manager that is found within the programmable logic or processing system in FPGA. It tracks the workload of systems, the availability of resources, and the external inputs to reconfigure the AI kernels when and which should be carried out. The manager communicates with a task profiling engine and reconfiguration scheduler and facilitates the automatic substitution of compute kernels to be less latency-bound, energy-bound or accuracy-bound.

The Secure Bitstream Loader deals with the authenticated and non-tamperable bitstream loading of reconfiguration bitstreams into the FPGA. Prior to installing any update, the loader will verify its integrity through cryptographic hash functions (e.g., SHA-256) and decrypt the encrypted bitstream with the help of appropriate authenticated encryption (e.g., AES-GCM). This way, only validated and trusted configurations will be used in operations. This loader too includes mechanism of roll back and recovery in case of failure or detection of an attack.

The architecture supports the creation of a hardware-based trust to enable the creation of a Hardware Trust Anchor, e.g., Physical Unclonable Function (PUF) which gives each FPGA its own, non-clonable identity. This element is deployed to produce device keys, using a cryptography algorithm and the AES construction set, to decrypt bitstream and validate the integrity of a bitstream without the use of externally-stored keys. The system confers immunity to side-channel attacks and physical tampering, by integrating the trust anchor into the FPGA fabric.

At last, the Industrial IIoT Gateway will be the control and communication interface between the edge nodes and the centralized or federated orchestration platform. It orchestrates task migration, disseminates signed bitstreams, sanctions telemetry information and implicates access control regulations. Synchronization of edge reconfiguration events on various nodes of the network is also the responsibility of the gateway, which contributes towards scalability and balancing of the workload.

The combination of these elements provides a highly interconnected infrastructure able to deliver secure, flexible, and scalable AI deployment in industrial contexts that are deemed critical. The scalability of the architecture provides the flexibility to implement a bespoke version of the system on an individual basis and at different levels of deployment, all the way to a federated mesh network, where the system can be made resilient to cyber-physical attacks.

Framework Components

The proposed secure and scalable framework of improved runtime reconfiguration consists of three components that are strongly interconnected with each other so that the capability of dynamic adaptability, resilient security, as well as efficient scalability of AI execution is provided in FPGA-based IIoT edge conditions. These are Runtime Reconfiguration Manager, Secure Bitstream Authentication Module and a Scalable Deployment Model. All of them are developed to deal with a particular issue related to real-time task flexibility, the protection of bitstreams, and multi-node coordination.

Runtime Reconfiguration Manager

The Runtime Reconfiguration Manager (RRM) is where dynamic task switching on FPGA is orchestrated. It keeps track of the availability of resources, inference latency and application level priorities to decide the opportune time to partially reconfigure. This would allow AI compute processing blocks, i.e., convectional neural network (CNN) layers, decision trees, or anomaly detection engines to be swapped efficiently in real-time as the operational needs change. The RRM works in collaboration with the Partial Reconfiguration Regions (PRRs) in FPGA and hence is designed such that the hardware change does not affect other important functionality taking place in the FPGA static regions. It also maintains reconfiguration queues and scheduling conflicts and uses metadata (input shape, model complexity) about AI models to make the decision and inform placement and timing. The RRM is the basis of runtime flexibility so that the system could respond to changing IIoT workloads.

Secure Bitstream Authentication

The Secure Bitstream Authentication module is used to safeguard illegal reconfiguration of the FPGA by specifying authenticated unmodified bitstreams before loading them into the FPGA. The module incorporates cryptographic algorithms used in the field, such as AES-GCM (Advanced Encryption Standard Galois/Counter Mode) to encrypt and verify bitstreams and SHA -256 to create and check safe digests.

The key contribution to this module is the integration of the Physical Unclonable Function (PUF) based key generation; this eliminates the need of storing persistent keys on hardware since cryptographic keys can be generated through device-specific hardware traits. This increases the resistance of the system to key theft, reverse engineering, side-channel attacks. The circuit shows how a device-specific cryptographic key is created as a result of applying a PUF module. When a challenge input is provided, the PUF is made to respond, and the result is passed to a key derivation function (KDF) to create a secure encryption key that is applied in authenticating the reconfiguration bit stream.

When the IIoT gateway sends an authenticated bitstream signed by it, the Secure Bitstream Loader will check the authenticity of the bitstream based on PUF-derived keys and only starts loading it to reconfigure after verifying the authenticity of the bitstream, making this a strong chain of trust.

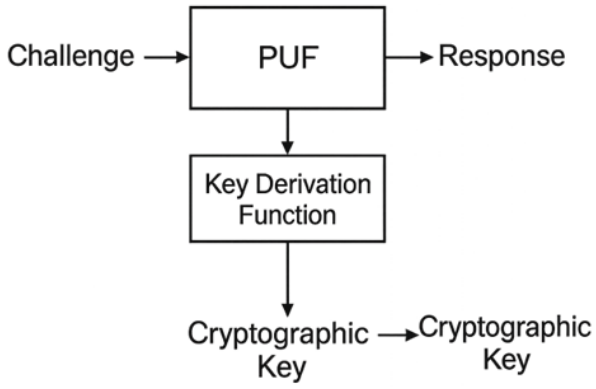

Fig. 2: PUF-Based Cryptographic Key Derivation Process

Scalable Deployment Model

Its Scaleable Deployment Model can be utilized to manage a network of edge FPGAs in a distributed IIoT mesh, and make the framework functional at a high rate. It uses distributed control plane to redistribute decisions of reconfiguration among node such that the

Pseudocode – Runtime Reconfiguration Manager (RRM)

```
Algorithm: Runtime Reconfiguration Decision Logic
Input:
  T = {T1, T2, ..., Tn}     // AI Task Pool
  PRRs = {R1, R2, ..., Rm}  // Partial Reconfiguration Regions
  M = system_metrics()      // Latency, accuracy, energy

Output:
  Reconfigure(PRRs[i], Tk)  // Triggers secure reconfiguration

Procedure:
1. while system_is_running():
2.     for each PRR in PRRs:
3.         current_task ← PRR.get_current_task()
4.         best_task ← current_task
5.         for each task in T:
6.             if task.resources_fit(PRR) and task.performance(M) > best_task.performance(M):
7.                 best_task ← task
8.         if best_task ≠ current_task:
9.             if verify_bitstream(best_task):
10.                Reconfigure(PRR, best_task)
11.    sleep(refresh_interval)
```

Table 1: Secure Reconfiguration Timing and Overhead Analysis for Different AI Task Types

Task Type	Bitstream Size (KB)	Reconfig Time (ms)	Auth. Overhead (ms)	Total Time (ms)	Security Overhead (%)
CNN Layer (Conv2D)	420	18.3	1.7	20.0	8.5%
Decision Tree	250	11.2	1.1	12.3	9.0%
Anomaly Detector	310	13.8	1.5	15.3	9.8%

workload is balanced based on computer power and budget as well as real-time meeting requirement. This ensures that no FPGA can be a bottleneck so that loads can be shared and toleration of fault useable. Each edge node of the network maintains a lightweight runtime agent which communicates back to the control plane to report changes of status of edge nodes and accept task migration guidance. Metadata reconfiguration such as resource footprints, model accuracy profiles and bitstream digests are stored together in a decentralized ledger or cache to be fast retrieved. The model also supports asynchronous node-to-node reconfiguration with all FPGAs reconfiguring independently and handling the resulting coherence of the remaining system.

Methodology

The present section provides details of the complete end-to-end methodology of proposing the design, security enforcement, and deployment of AI-based tasks leveraging the proposed Secure Runtime Reconfiguration Framework to FPGA-based devices of IIoT systems. The process consists of four large steps which are all keyed to system components and security measures. Figure 3 and Algorithm 1 provide the information about this methodology in a manner of design flowchart and a formal statement of secure management of the regions of partial reconfiguration (PRRs), respectively.

AI Model Profiling and Partitioning

The profiling of AI models targeted at deployment on edge FPGAs is done as the first stage. Getting to these models, those are usually assembled out of motor units computational kernels e.g., kickers over convolutional layers, activation functions, decision trees, or anomaly detectors. Characterizing of each kernel will be done in terms of its compute intensity, latency, memory footprint and power consumed.

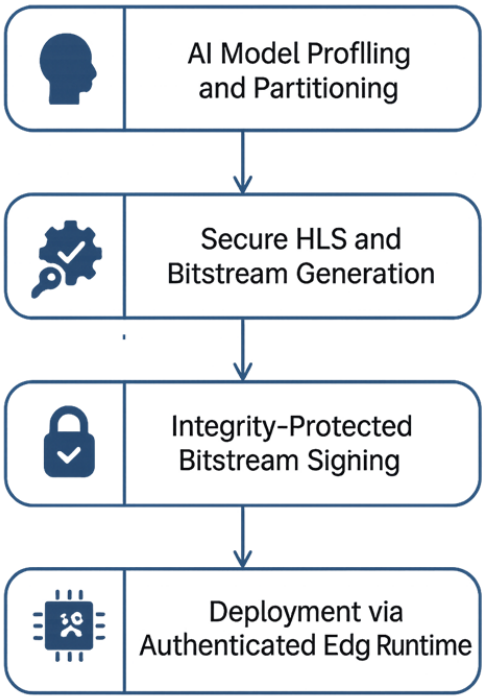

Fig. 3: Secure Reconfiguration and Deployment Workflow

The AI model is divided into blocks that can be executed on the hardware after profiling. Fast blocks are assigned to Partial Reconfiguration Regions (PRRs), whereas the rarely used ones (e.g., normalization layers or feature flattening) would be implemented in the static logic or run in software. This dividing is optimum in both senses of efficiency because this does what is required of the reconfigurable logic since it specifically adapts towards the changing work and also in the less power and space consumed.

Secure HLS and Bitstream Generation

The synthesis of a hardware description of each partitioned kernel is then done by applying High-Level Synthesis (HLS) techniques. RTL code is composed of partial bitstreams compiled by vendor applications

like Xilinx Vivado PR Flow or Intel Dynamic Function eXchange. Such bitstreams are only linked to particular PRRs and kernel setup. The bitstreams are cryptographically wrapped in order to make them secure. To achieve confidentiality and integrity of each bitstream, AES-GCM (Advanced Encryption Standard, Galois/Counter Mode) encryption is used. Another computation is a SHA-256 hash of the unencrypted bitstream and this is signed with a key, which is derived by a Physical Unclonable Function (PUF). The effect of this is a reconfiguration payload which is tamper-evident and specific to the device.

Integrity-Protected Bitstream Signing

The cryptographic signatures and bitstreams are generated, and packaged into secure deployment. Such bundles are saved in repositories with access to the edge or are sent on the Industrial IIoT Gateway. The target FPGA device has a secure handshake operation with the gateway when it receives the packet. The hash and signature are checked with the help of the derived key obtained by the device through the PUF. It is through this authentication that only trusted reconfiguration pay-load is accepted, thus staving off threats to bitstream injection, IP theft, or roll back attacks.

Deployment via Authenticated Edge Runtime

When the bitstream is authenticated, it is handed off to the Secure Bitstream Loader which decrypts and reconfigures the targeted PRR in the FPGA. This leaves the rest of the continuously running system unaffected and reconfigurable without impacting the system.

The AI Kernel Reconfiguration Manager monitors and logs reconfiguration events, and, based on inference latency data, resource use, and the environmental situation, the manager schedules future updates. This allows it to be easily adaptable to changes in workload in real time and security guarantees.

Experimental Setup and Evaluation

An extensive experimental study has been carried out to test the success of the Secure and Scalable Runtime Reconfiguration Framework that has been proposed. The analysis is aimed at quantifying the runtime

Algorithm 1: Secure PR Region Management

```
Algorithm 1: Secure_PR_Region_Management

Inputs:
    PRRs = {R1, R2, ..., Rn}     // Available Partial Reconfiguration Regions
    Tasks = {T1, T2, ..., Tm}    // AI tasks with metadata
Bitstream_Repo               // Secure bitstream repository

Procedure:
1. for each PRR in PRRs:
2.     current_task ← PRR.get_current_task()
3.     candidate_tasks ← Tasks.filter_by(PRR.resources)
4.     for task in candidate_tasks:
5.         bitstream ← Bitstream_Repo.get(task)
6.         if validate_signature(bitstream.hash, PUF_Key(PRR.device_id)) and
7.            check_integrity(bitstream, AES_GCM):
8.             if task.priority>current_task.priority:
9.                 reconfigure(PRR, bitstream)
10.                log("Reconfiguration Successful", PRR.id, task.id)
11.        else:
12.            log("Bitstream Validation Failed", PRR.id, task.id)
```

flexibility of the framework, inference throughput, power consumption, and system protection overhead under natural industry edge AI circumstances.

Setup

The experiment relies partly on the Xilinx Zynq UltraScale+ MPSoC platform mainly because it is a heterogeneous processing platform consisting of a quad-core ARM Cortex-A53 core, a Cortex-R5 dual-core, and programmable FPGA fabric, which makes the platform perfect in edge AI deployments that need to perform runtime contest and hardware acceleration. The system was designed using the Vivado Partial Reconfiguration (PR) flowflow and cryptographic primitives like AES-GCM and SHA-256 were designed by implementation of FPGA logic as well as run on the ARM-side software. To model realistic scenarios of industrial IoT, two data sets were used to load a lightweight convolutional neural network (CNN) on the abnormalities of the vibrations of the industrial machine in real time and to model the predictive maintenance of temperature logs that run on distributed sensors. The performance of the framework was examined on four critical parameters: reconfiguration time (ms), denoting the time that a partial bitstream has to be loaded and activated; energy consumption (mJ), measured during the tasks transitions and inference execution; detection accuracy (%), measuring the classification accuracy of the AI model; and security overhead (%), indicating the percentage increase in latency under the influence of cryptographic authentication and PUF-based key generation. Each test was repeated 50 times to provide statistical consistency and the mean values were to be used as an element of analysis.

Results and Analysis

To evaluate the effectiveness and resilience of the suggested secure reconfiguration system, two exemplar AI loads, including Anomaly Detection and Object Recognition, were carried out and tested on the edge platform using an FPGA. Such tasks have been selected owing to their applicability in case of a typical industrial monitoring and control system.

A lightweight convolutional neural network (CNN) was demonstrated in the Anomaly Detection task with a great detection accuracy of 94.3%. Its reconfiguration latency of 12.4 ms and moderate energy consumption of 15.6 mJ would make it quite convenient in terms of the resources consumed in real-time applications with energy limitations in the context of IIoT.

Conversely, since the Object Recognition task needs a more sea serpentine CNN structure, its reconfiguration time (18.7 ms) was more expensive, as well as its energy consumption (21.2 mJ), though it yielded a decent 90.1 percent accuracy. This indicates that the framework is real-time even as the workloads being processed become more complex.

More importantly, it was shown that the security overhead of any additional latency unavoidably caused by the AES-GCM encryption, the hashing function based on the 64-bit SHA-256, and the PUF-based key generator could be considered very small: 3.5 % (Anomaly Detection) and 4.2 % (Object Recognition). These findings illustrate that the incorporation of good security measures does not impact the system performance and reactions negatively to a considerable degree.

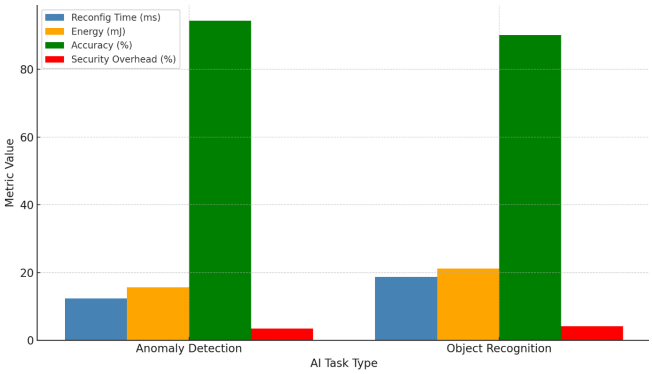

Fig. 4: Comparative Performance Metrics of AI Tasks

Table 2: Performance Metrics across Different AI Tasks

Task	Reconfiguration Time (ms)	Energy Consumption (mJ)	Detection Accuracy (%)	Security Overhead (%)
Anomaly Detection	12.4	15.6	94.3	3.5
Object Recognition	18.7	21.2	90.1	4.2

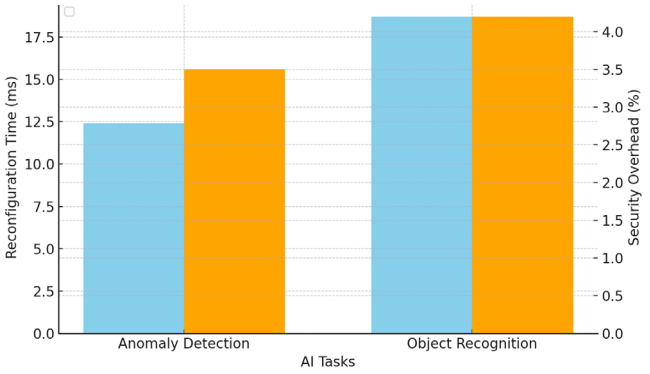

Fig. 5: Impact of Secure Partial Reconfiguration (PR) on Edge AI Performance

Discussion

A major trade-off in secure runtime reconfiguration is the trade between latency added, and enhanced system security. Cryptographic functions, including AES-GCM-based encryption, SHA-256-based hashing, and proof of feasibility using PUF-based key generation, pose a minimal overhead of a few percent on the reconfiguration time in the proposed framework, with all workloads achieving less than 5% overhead on average. Although this latency cannot be ignored, it is a small price to pay in regards to the amount of protection offered over bitstream tampering, rollback attacks, and unauthorized updates. The trust anchored in hardware e.g. using PUFs means that even after interception of the bitstream then there is no possibility of the replay or decryption in a different device hence device specific confidentiality and integrity. It shows that integration of strong authentication mechanisms into edge AI systems is feasible without affecting the responsiveness of real-time response, thus making the approach suitable in functionally safety critical applications in the industry.

In addition to the security property, the framework has also proven to have immense benefits in terms of scalability compared to the conventional implementation of a FPGA scheme, where one is ascertained because of the inertia of the FPGA platform. The ability to reconfigure tasks in AI at runtime allows the system to dynamically respond to multiple and frequent changes in workloads, power, and sensor input by scaling, adopting Energy Efficient Application Programming Interface (EEAPI) and dynamic resource assignment models, and coveys it as being well suited to heterogeneous IIoT clusters. With the help of the distributed control plane, coordinated task migration and load balancing, it is possible to reschedule the AI kernels and redistribute the system with minimal interruption, between properties of edge nodes. THIS is compared very much with points of static FPGA systems whose full reprogramming or manual redeployment is needed to enable new tasks. Consequently, the architecture in question achieves not only increased flexibility and fault tolerance of the operations but can be scaled out to support future growth, model versions, and multi-service orchestration, which are major drivers of the next-generation smart manufacturing and autonomous industrial control systems.

Conclusion

The present paper has proposed a Secure and Scalable Runtime Reconfiguration Framework that is useful in FPGA-based Edge AI deployments used in IIoT settings. The suggested work attempts to comprehensively incorporate the features of partial reconfiguration, PUF-based security and distributed control model to facilitate real-time, low-latency and authenticated execution of AI tasks using heterogeneous edge platforms. The architecture facilitates dynamic and elastic loading/unloading of AI kernels without prior interrupting the functioning of a system, and maintains bitstream integrity and device-level trust.

Key experimental results prove that the framework incurs small security overhead (less than 5%), and its reconfiguration latency is small (down to 12.4 ms) and its detection accuracy is high (up to 94.3). These outcomes confirm that the application of strong cryptographic authentication design aid is feasible without a trade-offs in performance- this is a principle need of mission-level IIoT systems. The distributed runtime was also found to be scalable in terms of task migration and the solution could be deployed to large scale industry.

The importance of this work is its global nature as it unites three major vectors, such as the run-time adaptability of runtime adaptability, safe bitstream transmission, and scalable edge AI. This framework offers a long term infrastructure unlike the static FPGA systems that are outmoded, capable of adapting to application requirements, enable over the air updates

securely and minimise server downtimes in automation powered landscapes. It therefore establishes a basis of trusted, reconfigurable AI inference on novel applications in the industry 4.0.

In the future, an implementation of reinforcement learning-based decision engines into intelligent task scheduling, the use of blockchain-based secure bitstream delivery and even expanding the runtime to other accelerators like GPUs and custom AI chips might be in the interests of realizing future work. Also, the use of fault-tolerant reconfiguration and exploration of energy-friendly migration strategies will aid to further bolster the robustness and sustainability of edge intelligence in the industrial complex ecosystems.

Future Work

Though the presented framework eloquently solves the problems of runtime reconfiguration, secure deployment and scalability at the edge, there are a number of opportunities available today to make the framework even more flexible, more intelligent and more interoperable in next-generation industrial Internet of Things (IIoT) settings.

Among the operational directions, one is the task offloading policies based on Deep Reinforcement Learning (DRL). Today, the heuristics of the reconfiguration manager uses rule-based heuristics and static profiling to enable the scheduling of AI tasks. With DRL, the system will learn the best reconfiguration strategies in the long term and adapt to fluctuating workloads, congestion, and energy limits. A DRL agent can be trained to help context oral decisions on when and where to migrate AI kernels and enhance resource utilization and latency-considered measures of assigning edge clusters to tasks.

The second area of interest in future study is the application of secure bitstream distribution using blockchain as support. Although the design is able to guarantee secure authentication based on the use of PUFs and symmetric encryption, the design is based on a trusted edge infrastructure. To help with clearance and traces (to remove single points of failure) of bitstreams, version history, and access control, blockchain can be used to keep an undeletable audit trail of such information. The process of deploying can also be automated with the implementation of smart contracts, so that only devices that are proven authentic and authorized can run particular AI kernels.

Finally, including more permissive accelerators like GPUs and FPGAs in the edge federations would be a considerable change that would enhance applicability of frameworks in mixed-workload settings. Most IIoT applications require a variety of computational tasks, including those that deal with vision processing, to control loop implementation, which can be implemented using a variety of hardware backends. A common layer of abstraction of dynamic scheduling of the kernel in FPGCAs, graphics and custom to AI ASIC devices would enable the edge system to be adaptive and choose the most appropriate accelerator available to perform a particular workload, thus maximizing the performance-per-watt and enhancing the overall system responsiveness.

All of these future improvements would combine to make it smarter, decentralized, hardware-agnostic, all of which are desirable characteristics in the emerging field of secure, real-time, and scalable edge AI on the industrial automation scene.

References

1. Stott, E., Sedcole, P., & Cheung, P. (2015). Dynamic partial reconfiguration of FPGAs: A survey of architectures, tools, and applications. *Proceedings of the IEEE, 103*(3), 412–431. https://doi.org/10.1109/JPROC.2014.2360713
2. Xilinx Inc. (2023). *Vivado Design Suite User Guide: Partial Reconfiguration (UG909)*. Retrieved from https://www.xilinx.com
3. Tehranipoor, M., &Koushanfar, F. (2010). A survey of hardware Trojan taxonomy and detection. *IEEE Design & Test of Computers, 27*(1), 10–25. https://doi.org/10.1109/MDT.2010.7
4. Sadeghi, R., Schellekens, D., &Preneel, B. (2008). A secure FPGA-based architecture for cryptographic key management. *IEEE Transactions on Computers, 57*(11), 1505–1518. https://doi.org/10.1109/TC.2008.112
5. Guajardo, J., Kumar, S. S., Schrijen, G. J., & Tuyls, P. (2007). FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems – CHES 2007* (pp. 63–80). https://doi.org/10.1007/978-3-540-74735-2_5
6. Mittal, S., & Vetter, J. S. (2015). A survey of methods for analyzing and improving GPU energy efficiency. *ACM Computing Surveys, 47*(2), 1–23. https://doi.org/10.1145/2656133

7. Jha, S. K., Mishra, A., & Banerjee, S. (2021). AI-HW co-design on FPGA for edge analytics in Industry 4.0. In Proceedings of IEEE INDIN 2021 (pp. 443–448). https://doi.org/10.1109/INDIN45578.2021.9557563
8. El Haj, A., & Nazari, A. (2025). Optimizing renewable energy integration for power grid challenges to navigating. Innovative Reviews in Engineering and Science, 3(2), 23–34. https://doi.org/10.31838/INES/03.02.03
9. Ramchurn, R. (2025). Advancing autonomous vehicle technology: Embedded systems prototyping and validation. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 56–64.
10. ASIF, M., Barnaba, M., Rajendra Babu, K., Om Prakash, P., &Khamuruddeen, S. K. (2021). Detection and tracking of theft vehicle. International Journal of Communication and Computer Technologies, 9(2), 6–11.
11. Sathish Kumar, T. M. (2025). Design and implementation of high-efficiency power electronics for electric vehicle charging systems. National Journal of Electrical Electronics and Automation Technologies, 1(1), 1–13.
12. Toha, A., Ahmad, H., & Lee, X. (2025). IoT-based embedded systems for precision agriculture: Design and implementation. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 21–29.