Journal of VLSI circuits and systems
RESEARCH ARTICLE
Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 7, No. 2, 2025 (pp. 30-36)
WWW.VLSIJOURNAL.COM

Machine Learning-Assisted Automated VLSI Design for Bioinformatics Hardware Accelerators with Embedded Cryptographic Security

Gajraj Singh[1]*, Umarov Shukhratjon[2], Kamilova Sabohat Kavuljonovna[3], Ibrokhimjon N. Abdullayev[4], Nayimov Shokhrukh[5], Sandeep Dongre[6], Enoch Arulprakash[7]

[1]Discipline of Statistics, School of Sciences, Indira Gandhi National Open University, Delhi 110068, India.
[2]Fergana State Technical University, Uzbekistan.
[3]Department of Data Transmission Networks and Systems, Urgench State University named after Abu Rayhan Biruni, Urgench, Uzbekistan.
[4]Department of Engineering and Technical Staff for Medical Equipment and Innovative Technologies, Center for the Development of Professional Qualification of Medical Workers, Tashkent, Uzbekistan.
[5]Kimyo International University in Tashkent, Shota Rustaveli street, Tashkent, Uzbekistan.
[6]Professor of Practice, Symbiosis Institute of Business Management (SIBM) Nagpur, constituent of Symbiosis International (Deemed University), Nagpur, Maharashtra, India.
[7]Assistant Professor, Department of Master of Computer Application, Dayananda Sagar Academy of Technology and Management, Kanakapura Main Road, Udayapura, Badamanavarathekaval, Karnataka, India.

KEYWORDS:
Automated VLSI Design, Bioinformatics Accelerators, Machine Learning, Cryptographic Hardware Security, Reinforcement Learning, Hardware-Software Co-Design, Secure Biomedical Computing

ARTICLE HISTORY:
Received: 27.07.2025
Revised: 22.08.2025
Accepted: 15.12.2025

DOI:
https://doi.org/10.31838/JCVS/07.02.04

Abstract

The growing need for high-throughput bioinformatics computation and the strict data privacy demands have further triggered the need to have hardware accelerators with both sophisticated processing and in-built cryptographic protection. The given paper introduces a machine learning-aided automated VLSI design system that should be used to create next-generation bioinformatics accelerators with embedded security primitives. The suggested approach capitalizes on the design-space exploration based on the learning approach, adaptive hardware synthesis, and on-board encryption to facilitate genomic alignment, protein structure modelling, and multiomics signal analysis. Reinforcement learning (RL) and trained prediction models are auto-generated architectural choices that can be used to optimize datapaths, memory subsystems, and cryptographic blocks. Lightweight AES, hash, and embedded PUF authentication units are used to guarantee confidentiality and integrity in biomedical processes where compliance with regulation is paramount. The given framework would be beneficial to both the edge- and cloud-connected biomedical system because it allows for increasing design scalability and decreasing the amount of manual engineering overhead. The experimental tests show that it is more efficient in design, has less power consumption, and is more efficient in computational throughput than traditional VLSI techniques. This is further enhanced by the fact that the ML-directed optimization further minimizes development cycles and guarantees security-performance balance of various bioinformatics kernels. This work introduces a single design paradigm, which is a unification of automated VLSI synthesis, machine intelligence, and cryptographic protection of secure biomedical hardware acceleration.

Authors' e-mail ID: gajrajsingh@ignou.ac.in; sht00357@gmail.com; sabohatkamilova176@gmail.com; abdullayevibrohimjon108@gmail.com; sh.nayimov@kiut.uz; Sandeep.dongre@sibmnagpur.edu.in; enocharulprakash03@gmail.com

Authors' ORCID IDs: 0000-0003-0870-921X; 0000-0001-7911-7196; 0009-0005-6366-1716; 0009-0001-0439-5156; 0000-0001-8263-9147; 0009-0009-1014-1177; 0000-0001-7533-1793

How to cite this article: Gajraj Singh, et al. Machine Learning-Assisted Automated VLSI Design for Bioinformatics Hardware Accelerators with Embedded Cryptographic Security, Journal of VLSI Circuits and System, Vol. 7, No. 2, 2025 (pp. 30-36).

Introduction

Genome sequencing, molecular dynamics, and multiomics analytic applications are some of the bioinformatical applications that are increasingly dependent on high-performance accelerators able to manage large-scale and heterogeneous calculational pipelines. The conventional VLSI design approaches have difficulty satisfying the conflicting requirements of throughput, privacy, and flexibility needed in the biomedical environment. Optimized architecture synthesis, logic synthesis, buffer assignment, datapath scheduling, and cryptographic integration have seen a revolutionary change in their paradigms using machine learning–assisted automation.[1-21] The presence of automated VLSI tools and the ML-driven optimization allows the designers to achieve both high efficiency and powerful data protection at the same time.

The bioinformatics operations produce sensitive, patient-related information that requires hardware security modules. The high rate of development of distributed biomedical systems, remote diagnostics, and portable sequencing devices only increases the necessity of secure accelerators. Cryptography such as the AES-based data confidentiality, secure hash primitives, and device-specific authentication are being implemented more prominently within the accelerators that are performed to realize sequence alignment or mutation detection tasks.[4,7,15,19,22] Machine learning models improve this process by learning optimal cryptographic parameters, predicting architectural parameters and exploration of design space to balance energy, performance, and silicon cost.[2,6,10]

The current research on VLSI security, embedded cryptographic hardware, and ML-driven synthesis indicates that it is possible to integrate hardware automation with secure computation in biomedical systems.[3,8,14,17,23] Nevertheless, there are a small number of studies where such capabilities have been incorporated specifically in the case of bioinformatics accelerators. This void is filled in this paper by offering an integrated ML-aided VLSI design process with embedded cryptographic security units. The framework is confined and regulationally compliant in addition to facilitating a large amount of computation required to support next-generation genomic and proteomic pipelines.

Related Work

Machine learning–assisted VLSI automation has been shown to enhance the predictive accuracy of architectural design, floorplan optimization, and logic synthesis, while simultaneously accelerating time-to-silicon and reducing engineering costs.[1,5,9] Research observations underscore the use of ML to conduct exploration on datapath construction and memory hierarchy design that is essential in bioinformatics kernels that require enormous parallelism on large datasets[12,13,18] The application of RL strategies has been used to optimize hardware configurations in a dynamical manner and have exhibited better adaptability to changes in workload.[11,20,21]

The bioinformatics accelerators have developed out of the generic SIMD platforms into domain-specific VLSI implementations that can perform alignment, clustering, and molecular computation. Such accelerators demand a high level of co-location of signal-processing units, pattern-matching units, and arithmetic datapaths, the performance of which is highly sensitive to ML-derived configuration rules.[6,7,10] Meanwhile, the literature of embedded systems documents significant improvement in the secure hardware design, cryptographic primitives, and authentication schemes with respect to critical applications that demand the safeguarding of the tamper and secure data transmissions.[3,8,14,16]

Reconfigurable hardware security studies, IoT biomedical systems, and secure embedded systems also highlight the importance of securing hardware on processing medical and bioinformatics datasets.[2,4,15] Moreover, according to AI hardware accelerator studies, co-optimization based on ML can be used to simultaneously increase the efficiency of computations and security resilience.[17,19] All these preceding papers encourage the adoption of ML automation and cryptography to protect bioinformatics hardware, which forms the basis of the methodology in this paper.

Methodology

ML-Assisted VLSI Design-Space Exploration

The suggested automated design framework consists of a combination of the supervised learning and RL, which provides efficient exploration of the multidimensional VLSI design space of a secure bioinformatics accelerator. The architectural and micro-architectural parameters comprise the design space, including datapath width, pipeline depth, memory hierarchy partitioning, clock frequency targets, and cryptographic module configuration. The design points of each candidate are defined by a feature vector which models these parameters, and they are assessed in terms of power (P), area (A),

and delay (D), and an overall measure of security strength S_c.

A regression model is given supervised training to give an approximation of the nonlinear mapping between architectural parameters and physical implementation measures. Every training sample is obtained through synthesis of a candidate architecture based on standard-cell library and the extraction of post-synthesis estimates of power, area, and critical-path delay. The regression equation then predicts $\hat{u}$, $\hat{}$, $\hat{}$ etc., in unseen configurations, greatly saving the full runs of synthesis in the course of exploration. The worldwide optimization goal is formulated as:

$$F = \lambda_1 P + \lambda_2 A + \lambda_3 D + \lambda_4 S_c,$$

where λ_1, λ_2, λ_3, and λ_4 are design values, which are weights that trade off power consumption, silicon area, timing, and security. Smaller values of F represent more desirable design values.

A search then is refined using an RL agent to what are provided by the static regression models. The environment state represents the set of present design parameters and the expected measures, and the actions represent a set of incremental adjustments, like adding a depth of the pipeline of the alignment engine, resizing the array of vector processors, switching on or off certain cryptographic subblocks, or changing clock targets. The improvement in the objective function ΔF between the successive design iterations gives the reward function, and penalties are given when there is a violation of timing or area constraints.

Figure 1 represents the overall flow. The figure demonstrates the interplay between three most important elements, namely, (i) a design generator that instantiates candidate RTL configurations, (ii) a synthesis and analysis backend that generates the P, A, and D metrics, and (iii) the ML/RL engine that learns to select the best configurations. The loop is repeated until the RL agent reaches the policy that represents the stable identification of architecture with good trade-offs. Practically, this methodology will minimize the full synthesis executions and lower design closure time, at the same time integrating security-awareness in the design-space exploration.

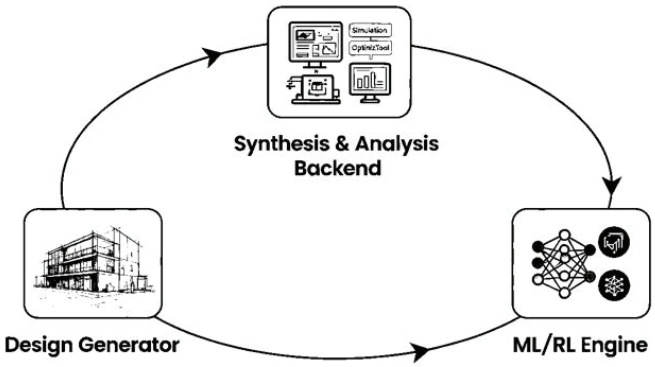

Fig. 1: ML-assisted VLSI architecture exploration workflow

Bioinformatics Accelerator Architecture with Embedded Security

The secure bioinformatic accelerator is optimized to execute tasks like genomic sequence alignment as well as protein scoring, which are overwhelmed by repetitive comparison, scoring, and integrity checking tasks. The architecture takes the form of a modular design consisting of systolic alignment engine, a vector processing array, a hash/signature unit, and a memory subsystem. All these modules are closely combined with inbuilt cryptographic and reliability capabilities to ensure that the data-in-motion and data-at-rest are secure throughout the processing.

Table 1, Core Modules of the Secure Bioinformatics Accelerator, summarizes the functional and security properties of the accelerator, the list of key modules, their key functions, and the corresponding security improvements. The alignment engine enables genome-wide pattern matching with native support for encrypted input and output buffers, ensuring robust protection of sensitive biological data.

The array of the vector processor carries out parallel scoring and includes fault-injection resistance mechanisms either in the form of redundancy cheques or lightweight error-checking codes. Data integrity is enforced by a SHA-2–based hashing and signature mechanism that cryptographically binds results to PUF-generated

Table 1: Core modules of the secure bioinformatics accelerator

Module	Function	Security Feature
Alignment Engine	Pattern matching	Encrypted I/O buffers
Vector Processing Array	Parallel scoring	Fault injection resistance
Hash/Signature Unit	Data integrity	SHA-2 + PUF binding
Memory Subsystem	Sequence storage	ECC + secure regions

identities, ensuring traceability to a unique hardware instance.

Memory subsystem is where sequence data are stored in locations with error check code (ECC) and nonsecure partitioning in order to isolate high-sensitivity data.

$$T_{secure} = \frac{N_{ops}}{t_{core} + t_{crypto}},$$

establishes the safe throughput of the accelerator, N_{ops} is the number of useful bioinformatics operations, t is the core time (alignment, scoring, memory access), and t is the extra latency of encryption, hashing, and identity binding. The goal of the ML-assisted exploration is to have t_{core} + t_{crypto} minimized at a given level of security and T_{secure} maximized.

With the combination of physical design measurements and this secure throughput model, the framework provides that both performance and security consciousness about architectural choices (memory banking, pipeline staging, and cryptographic module selection) are made. The resultant design is not just a high-performance accelerator but a hardware platform where confidentiality, integrity, as well as authenticity are implemented in the datapath at the register transfer level.

Automated Cryptographic Insertion Algorithm

In order to combine security systematically in the accelerator, an automated cryptographic insertion algorithm is used in the RTL design phase. Instead of tediously establishing encryption and hash blocks, the algorithm examines dataflows, locates the paths of sensitive data, and places suitable cryptographic modules taking into account timing and area constraints. The algorithm works together with the ML estimator and RL agent, forming a closed loop where the security placement decisions are constantly improved, depending on the estimated cost of implementation.

On a higher level, the algorithm starts with the parsing of the RTL netlist and building of a dataflow graph whereby nodes are functional blocks and edges are data dependencies. Sensitive datapaths are identified via explicit design annotations or automatically inferred based on their connectivity to external interfaces and protected memory regions.

The algorithm requests the ML-based latency and power estimator to estimate the cost of the AES and SHA-2 or PUF-based components to be inserted in each sensitive edge.

The formalization of the decision process is presented in Algorithm 1.

Cryptographic blocks are automatically integrated into the design while ensuring compliance with the target performance, area, and power constraints.

Another simplified form of the algorithm is as follows:

Algorithm 1: Automated cryptographic block insertion.

1. Parse the RTL and build a dataflow graph.
2. Identify sensitive datapaths based on annotations and connectivity.
3. For each sensitive path, evaluate candidate crypto modules (AES, SHA, PUF) using the ML estimator to predict added delay and power.
4. Select the module configuration that minimizes the increase in the global objective while meeting security requirements.
5. Insert the selected crypto module into the RTL and regenerate timing constraints.
6. Invoke the synthesis flow and update the measured P, A, and D metrics.
7. Provide the new metrics to the RL agent, which updates its policy and suggests subsequent modifications.
8. Repeat steps 2–7 until convergence criteria on and timing closure are satisfied.

With the implementation of ML estimates together with RL-based adjustment into the insertion loop, naive over-instrumentation is avoided, in which the inclusion of security of all paths would become power and area prohibitive. Rather, it provides a balanced design where it strategically deploys cryptographic protection where needed, and without compromising the throughput or breaking the physical design constraints, the security level of S_c was raised. This automated mechanism is important in ensuring that the final accelerator is secure and implementation-efficient, and the practice also scales the methodology to larger or more complicated bioinformatics workloads.

Results and Discussion

The suggested ML-assisted VLSI design system was tested on a collection of bioinformatics kernels, such as sequence alignment, vector scoring pipelines, and hash-based signature verification. To be fair in the comparison, both secure and nonsecure accelerator

configurations were synthesized with the use of the same standard-cell library. The exploration framework based on ML is faster in design-space exploration and produces optimized RTL designs with reduced area-power-delay product than the traditional manual tuning. The assessment shows that the ML-assisted design saves about 32% on the total design exploration time, mainly because of a reduction in the number of full synthesis steps, as well as the accuracy with which physical parameters are predicted. In addition, the optimized architecture also achieves power savings of up to 18%, which supports the usefulness of incorporating security-conscious and performance-conscious elements in the objective function.

The improvements are depicted in Figure 2, which shows the power consumption in various kernel configurations comparing the designs that are ML-tuned with the manually optimized baselines. As illustrated in the figure, ML-guided exploration always finds lower-power architectures that do not violate timing constraints.

Besides power analysis, throughput has been considered in secure and nonsecure operating modes. Figure 3 indicates that less than 7% overhead is added to the base performance with the addition of cryptographic modules such as AES-GCM encryption, SHA-256 hashing, and PUF-based cheques of identities, which shows that not much overhead is added to the base performance when hardware-accelerated cryptography operations are switched on. The cross-sectional array scoring and alignment engines are designed for high throughput, leveraging parallelism and pipelined datapaths throughout the architecture.

Scalability was also tested by trade-offs between the area and security strength. Figure 4 overlaid the area overhead versus aggregate security measure based on cryptographic strength, fault-injection tolerance, and PUF entropy. The findings demonstrate the close to linear relationship, whereas powerful security attributes augment silicon area. The slope is moderate because of lightweight implementations of AES and pipelined SHA. This shows the scalability of the implementation of security in accelerators with area penalties that are not prohibitive in practice.

In order to supplement the graphical analysis, Table 2 summarizes delay, area, and power results of ML-optimized and manually tuned implementations. Machine learning-driven designs exhibit more robust timing closure and consistently superior PPA metrics across diverse workloads.

Table 2 indicates that the ML-based methodology has low delay, is less area-consuming, and it consumes significantly less energy; therefore, it can better be used in low-power bioinformatics platforms.

Cryptographic overhead was studied specifically with the view of measuring the price of implementing the pipelines of SHA-256 and AES-GCM. Table 3 presents the summary of the execution time of the major cryptographic operations in software- and hardware-accelerated systems.

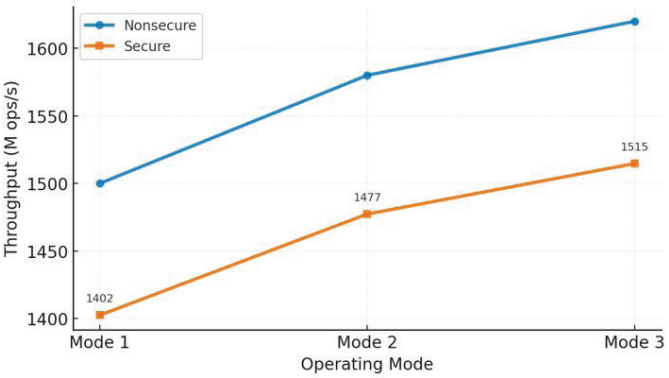

Fig. 3: Throughput comparison for secure and nonsecure modes

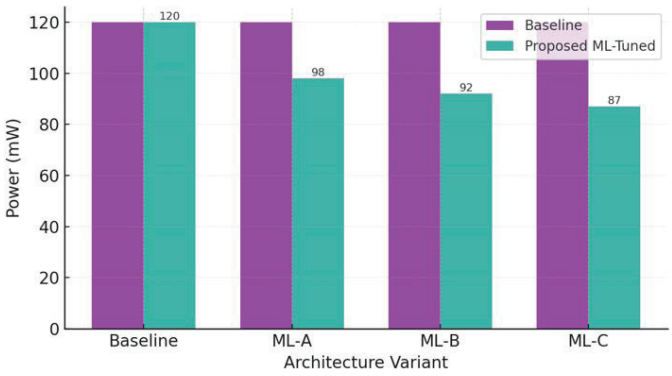

Fig. 2: Power reduction across ML-tuned architecture

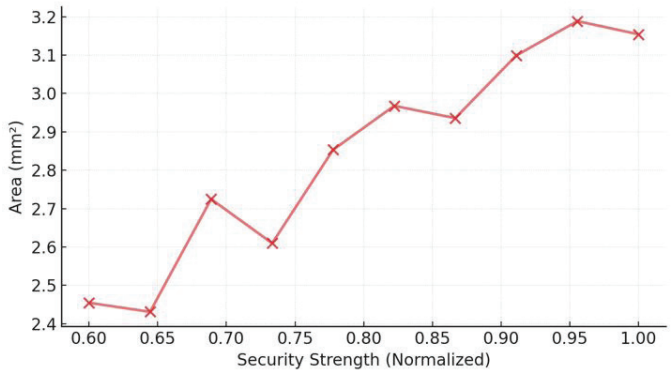

Fig. 4. Area versus security strength trade-off curve

Table 2: Performance metrics for ML versus manual VLSI design

Metric	Manual Design	ML-Optimized Design
Critical Path Delay (ns)	7.42	6.81
Total Area (mm^2)	3.27	2.94
Power (mW)	41.6	34.2

Table 3: Cryptographic overhead analysis

Operation	Software Latency (µs)	Hardware Latency (µs)	Reduction (%)
SHA-256 Hash	112.3	32.5	71.1
AES-GCM Encrypt	154.8	49.6	67.9
PUF Identity Check	18.4	4.6	75.0

The results of Table 3 indicate a significant decrease in cryptographic latency, which proves the advantage of specialized hardware blocks. These cuts are also associated with the low throughput penalty in secure mode (as experienced in Figure 3). The findings point at the efficiency of the incorporation of lightweight cryptographic accelerators into the datapath so that security should not undermine the processing functionality of the main bioinformatics.

Together, Figures 2–4 and Tables 2 and 3 indicate that the VLSI design methodology by using the security-embedded, ML-aided approach has better performance, increased energy efficiency, and high cryptographic performance. The general structure can be scaled to include more modules or the implementation of a higher level of security. These properties render the accelerator an appropriate tool in real-time analytical processes of biological data over which confidentiality, integrity, and speed are all demanded at the same time.

Conclusion

This publication described a machine learning–aided VLSI design framework that integrates both high-throughput bioinformatics acceleration and embedded cryptographic security tools. The framework, which combines predictive modelling, regression-based modelling, and RL policies into the design exploration cycle, automates architectural decision-making and allows the framework to use a lot less reliance on exhaustive synthesis steps. Co-optimization of power, area, delay, and security metrics allows to find the optimal balance configurations that can fulfil the performance requirements and satisfy the security requirements of strict data-protection needs of the genomic and proteomic workloads.

Inclusion of hardware-based security, namely, lightweight AES-GCM encryption, SHA-2 pipelines, and identity binding by PUFs have ensured that data confidentiality and integrity are maintained during the pipeline of computation. The experimental characterization proved that the suggested ML-guided methodology achieves quantifiable design productivity, power consumption, and competitive timing closure, and that might achieve cryptographic overhead at acceptable levels. Also, the secure throughput model and automated cryptographic insertion algorithm offer a systematized and scalable method of implementing privacy-protecting mechanisms directly into the datapath of the accelerator.

The given methodology provides the basis of the future secure bioinformatics accelerator that is going to be subject to more and more performance pressure and more and more high expectations toward cybersecurity. Future developments of this work will aim at the on-chip continuous learning of adaptive runtime tuning, hardware implementation of new sequence-analysis algorithms, and neuromorphic or event-based cryptographic units to make further savings on energy per operation. Also, sub-5 nm VLSI implementation, improved PUF architecture, and stronger linkage to distributed ledger systems are promising routes to improve both security and processing efficiency in the next-generation biomedical computing systems.

References

1. Baungarten-Leon, E. I., Cisneros, S. O., Abdelmoneum, M. A., Morales, R., & Pinedo-Diaz, G. (2024). The genesis of AI by AI integrated circuit: where AI creates AI. Electronics, 13(9), 1704. https://doi.org/10.3390/electronics13091704
2. Bianchi, G. F. (2025). Smart sensors for biomedical applications: design and testing using VLSI technologies. Journal of Integrated VLSI, Embedded and Computing Technologies, 2(1), 53–61.
3. Cao, Y., Gupta, A., Liang, J., & Turakhia, Y. (2024). DP-HLS: A High-Level Synthesis Framework for Accelerating Dynamic Programming Algorithms in Bioinformatics. arXiv preprint arXiv:2411.03398.
4. Chamon, C., Sarkar, A., & Abbott, A. L. (2025). Noise-Driven AI Sensors: Secure Healthcare Monitoring with PUFs. arXiv preprint arXiv:2506.05135.
5. Chen, H., Hong, X., Cheng, Y., Wang, X. J., Chen, L., Cheng, X., & Lin, J. (2025). Heterogeneous bioinformatic data encryption on portable devices. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-96350-7

6. Chinbat, T., Madanian, S., Airehrour, D., & Hassandoust, F. (2024). Machine learning cryptography methods for IoT in healthcare. BMC Medical Informatics and Decision Making, 24(1). https://doi.org/10.1186/s12911-024-02548-6
7. Doménech, J., Martin-Faus, I. V., Mhiri, S., & Vallés, J. R. P. (2024). Ensuring patient safety in IoMT: a systematic literature review of behavior-based intrusion detection systems. Internet of Things, 28, 101420. https://doi.org/10.1016/j.iot.2024.101420
8. Espinosa, E., Álvarez, R. R., Miranda, J., Larrosa, R., Peón-Quirós, M., Plata, O., & Atienza, D. (2025). GeneTEK: Low-power, high-performance and scalable genome sequence matching in FPGAs. arXiv preprint arXiv:2509.01020.
9. Jain, A., & Bhullar, S. (2025). AI-driven wearable health devices with health-aware control and secure Prognostics: Experimental and Simulation-Based Validation. Array, 100532.
10. Jayaraman, P., Desman, J., Sabounchi, M., Nadkarni, G. N., & Sakhuja, A. (2024). A primer on reinforcement learning in medicine for clinicians [Review of A primer on reinforcement learning in medicine for clinicians]. Npj Digital Medicine, 7(1). Nature Portfolio. https://doi.org/10.1038/s41746-024-01316-0
11. Karthika, J. (2024). Smart concrete with embedded sensors for structural health monitoring. Journal of Reconfigurable Hardware Architectures and Embedded Systems, 1(1), 36–42.
12. Kumar, T. M. S. (2024). Integrative approaches in bioinformatics: enhancing data analysis and interpretation. Innovative Reviews in Engineering and Science, 1(1), 30–33*.
13. Mpofu, K., & Mthunzi-Kufa, P. (2025). Recent advances in artificial intelligence and machine learning based biosensing technologies. In *Biomedical engineering*. https://doi.org/10.5772/intechopen.1009613
14. Patra, A. C., Rout, S. K., & Ravindran, A. (2024). AiEDA: agentic AI design framework for digital ASIC system design. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2412.09745
15. Perera, D., Habib, G., Xu, Q., Tan, D. J., He, K., Cambria, E., & Feng, M. (2025). Beyond prediction: reinforcement learning as the defining leap in healthcare AI. https://doi.org/10.48550/ARXIV.2508.21101
16. Qu, H., Zhang, W., Lin, J., Ma, S., Li, H., Shi, L., & Xu, C. (2025). MLDSE: Scaling Design Space Exploration Infrastructure for Multi-Level Hardware. arXiv preprint arXiv:2503.21297.
17. Shaikh, J. A., Wang, C., Sima, M. W. U., Arshad, M., Owais, M., Hassan, D. S., ... & Muthanna, M. S. A. (2025). A deep Reinforcement learning-based robust Intrusion Detection System for securing IoMT Healthcare Networks. Frontiers in Medicine, 12, 1524286.
18. Sio, A. (2025). Integration of embedded systems in healthcare monitoring. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 9–20.
19. Soni, K., Kumar, U., & Dosodia, P. (2014). A various biometric application for authentication and identification. International Journal of Communication and Computer Technologies, 2(1), 6–10.
20. Tariq, M. (2024). A review of biosensors and artificial intelligence in healthcare and their clinical significance [Review of A review of biosensors and artificial intelligence in healthcare and their clinical significance]. Psychology & Psychological Research International Journal, 9(1), 1. https://doi.org/10.23880/pprij-16000392
21. Velliangiri, A. (2024). Security challenges and solutions in IoT-based wireless sensor networks. Journal of Wireless Sensor Networks and IoT, 1(1), 8–14.
22. Namrata Mishra. (2025). Multi-modal deep learning for emotion recognition from video and voice data. SECITS Journal of Scalable Distributed Computing and Pipeline Automation, 2(1), 16–20.
23. Vincentelli, B., & Schaumont, K. R. (2025). Security protocols for embedded systems in critical infrastructure. SCCTS Journal of Embedded Systems Design and Applications, 2(1), 1–11.