STM JOURNALS

Journal of
VLSI Design Tools & Technology

ISSN: 2249-474X (Online)
ISSN: 2321-6492 (Print)
Volume 15, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoVDTT

https://journals.stmjournals.com/jovdtt

Review JoVDTT

VHDL Programming for Side-Channel Attack Countermeasures in IoT Security

Sanika Anil Bhosale[1,*], Kazi Kutubuddin[2], Sayyad Liyakat[2]

Abstract

The Internet of Things (IoT) landscape is expanding rapidly, connecting billions of devices across diverse domains. This interconnectedness, while offering unprecedented convenience and efficiency, also creates a fertile ground for security vulnerabilities. Among these threats, side-channel attacks (SCAs) pose a significant risk, particularly targeting the cryptographic implementations that underpin IoT security. SCAs exploit information leaked from the physical execution of cryptographic algorithms, such as power consumption, timing variations, and electromagnetic radiation, to reveal sensitive data like secret keys. Protecting IoT devices against SCAs is crucial, and one promising approach involves incorporating countermeasures directly into the hardware design using hardware description languages like Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL). This article explores the role of VHDL programming in implementing SCA countermeasures within IoT security. IoT devices often rely on cryptographic algorithms for authentication, encryption, and data integrity. However, even mathematically sound cryptographic algorithms can be vulnerable to SCAs if their implementation leaks information about their internal operations. For example, in a power analysis attack, an attacker can monitor the power consumption of a device while it executes a cryptographic algorithm. By analyzing the power consumption patterns, the attacker can deduce information about the key bits being processed, ultimately compromising the security of the device.

Keywords: IoT, security, side-channel attack, VHDL programming, VHSIC Hardware Description Language (VHDL)

INTRODUCTION

The Internet of Things (IoT) is revolutionizing various aspects of our lives, from smart homes to industrial automation. However, this interconnectedness presents a significant security challenge. IoT devices, which are often resource-constrained and deployed in vulnerable environments, are prime targets for malicious actors. Among various attack vectors, side-channel attacks (SCAs) pose a particularly subtle and potent threat. Owing to their ability to derive sensitive information by analyzing the physical characteristics of a device during operation, SCAs can bypass traditional cryptographic protections [1–10].

This section discusses the role of VHDL (VHSIC Hardware Description Language) programming in implementing effective countermeasures against SCAs in IoT device security. We will explore why hardware-level defenses are crucial, the types of SCAs that threaten IoT devices, and how VHDL can be leveraged to mitigate these vulnerabilities.

Software-based security solutions can be valuable but are often insufficient for sophisticated SCAs.

***Author for Correspondence**
Sanika Anil Bhosale
E-mail: sanikabhosale573@gmail.com

[1]Student, Department of Electronics & Telecommunication Engineering, Brahmdevdada Mane Institute of Technology (BMIT), Solapur, Maharashtra, India
[2]Professor& Head, Department of Electronics & Telecommunication Engineering, Brahmdevdada Mane Institute of Technology (BMIT), Solapur, Maharashtra, India

Received Date: April 17, 2025
Accepted Date: May 10, 2025
Published Date: August 20, 2025

Citation: Sanika Anil Bhosale, Kazi Kutubuddin, Sayyad Liyakat. VHDL Programming for Side-Channel Attack Countermeasures in IoT Security. Journal of VLSI Design Tools & Technology. 2025; 15(2): 20–33p.

Attackers exploiting SCAs monitor power consumption, electromagnetic radiation, timing variations, or acoustic emissions to deduce secret keys or algorithms. These signals are intrinsic to hardware implementation and can be exploited without directly accessing software. Therefore, embedding countermeasures directly into hardware design using VHDL is a vital security layer [11–20].

Several types of SCAs can compromise the security of IoT devices, as shown in Figure 1.

- *Power analysis attacks (PAAs):* These attacks monitor the fluctuating power consumption of the device to infer information about the operations being performed. Simple power analysis (SPA) can reveal the basic algorithm structure, whereas differential power analysis (DPA) uses statistical methods to correlate the power traces with secret keys.
- *Timing attacks:* By analyzing the time it takes for cryptographic operations to be completed, attackers can deduce information about the key being used. Variations in execution time based on key values can be exploited in timing analysis.
- *Electromagnetic (EM) analysis:* Similar to power analysis, EM attacks monitor the electromagnetic radiation emitted by the device during operation. EM emanations can reveal information about a device's internal state and cryptographic processes.
- *Acoustic attacks:* In cases, sound emissions from the device can also be used to deduce information about the calculations being performed.

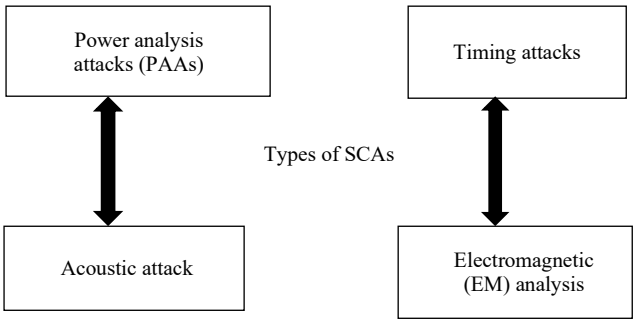

Figure 1. Types of side-channel attacks (SCAs).

VHDL allows designers to implement countermeasures at the hardware level, significantly increasing the resilience of IoT devices to SCAs. Here, the VHDL can be employed:

- *Masking:* This technique involves introducing random values (masks) to obscure the relationship between the device operations and the secret key. VHDL can be used to implement masked cryptographic algorithms, in which all sensitive variables are combined with random masks before processing. These masks are later removed, ensuring the correct computation outcome but obfuscating power consumption or other side-channel signals. Masking must be implemented carefully to avoid leakage of the secret key.

  Example of masking in VHDL

```
signal key_masked : std_logic_vector(127 downto 0);
signal random_mask : std_logic_vector(127 downto 0);

process (clk, rst)
begin
  if rst = '1' then
    key_masked <= (others => '0');
  elsif rising_edge(clk) then
    -- Generate a new random mask
    random_mask <= generate_random_mask();

    -- Mask the key
```



```
    key_masked <= key XOR random_mask;
    -- Perform operations on masked key
    -- ...
  end if;
end process;
```

- *Hiding:* This approach aims to make the device operations appear constant regardless of the actual calculations being performed. This involves the following techniques:
  - *Dual-rail logic:* This uses two wires to represent each bit, with one wire being the true value and the other being its complement. This ensures that the power consumption remains constant, regardless of the data being processed. VHDL can be used to implement logic gates and circuits.
  - *Constant power consumption circuits:* Circuits that consume a constant amount of power, regardless of the operations performed, can effectively hide information leakage.
  - *Dummy operations:* Dummy operations are inserted to obfuscate the timing and power consumption patterns.
- *Algorithm-level countermeasures:* Specific cryptographic algorithms are designed to be more resistant to SCAs. Implementing these algorithms in VHDL can provide a strong defense against attacks. Examples include:
  - *Threshold implementations:* A variant of masking where data are split into multiple shares. The number of shares required to reconstruct the original data was greater than the threshold, thereby preventing information leakage.
  - *Boolean masking:* Applying bitwise operations to secret data with random Booleans to obscure the relationship between the underlying secret data and the resulting side-channel leakage.
- *Randomization:* Introducing randomness into the execution flow of cryptographic algorithms can make it more difficult for attackers to correlate side-channel signals with the secret key. This can involve randomizing the order of the operations or using random delays. VHDL can be used to implement random number generators and control the timing of operations.

Implementing SCA countermeasures in VHDL presents several challenges:

- *Performance overhead:* Countermeasures can significantly affect the performance of IoT devices, which often have limited resources. Therefore, careful optimization is crucial.
- *Complexity:* Implementing sophisticated countermeasures can increase the complexity of the VHDL code, making it more difficult to verify and maintain.
- *Verification:* Thoroughly verifying the effectiveness of countermeasures against SCAs requires specialized tools and techniques, such as power analysis simulations and fault injection.
- *Cost:* The development and verification of SCA-resistant hardware can be expensive.

Side-channel attacks pose a serious threat to IoT device security. Although software defenses play a role, implementing countermeasures at the hardware level using VHDL is crucial for building robust and secure systems. By employing techniques such as masking, hiding, and algorithm-level countermeasures, developers can significantly reduce the vulnerability of IoT devices to SCAs. However, it is important to acknowledge that implementing such countermeasures introduces complexity and performance overhead and requires rigorous testing and validation. As the IoT landscape continues to expand, the demand for secure hardware will only increase, placing more emphasis on VHDL programming as a vital skill for ensuring the confidentiality and integrity of sensitive data [21–31].

VHDL: A KEY TOOL IN FORTIFYING IoT SECURITY AGAINST SIDE-CHANNEL ATTACKS

The IoT has exploded in recent years, connecting billions of devices, ranging from smart home appliances to critical infrastructure. However, such interconnectedness presents a significant security

challenge. One of the most insidious threats to IoT security is SCA, which exploits the physical characteristics of a device during cryptographic operations to reveal sensitive information. Fortunately, hardware description languages such as VHDL play a crucial role in developing and implementing robust countermeasures against such attacks [32–43].

Unlike traditional cryptographic attacks, which target algorithms directly, SCAs focus on extracting information leaked through measurable physical phenomena. These side channels can include the following:

- *Power consumption:* Variations in power consumption during cryptographic operations can reveal information about processed data and key bits.
- *Timing analysis:* Measuring the execution time of cryptographic operations can expose dependencies on key values.
- *Electromagnetic radiation:* Analyzing electromagnetic emanations can reveal correlations with the data being processed.
- *Acoustic emissions:* Sound generated during computation can, in some cases, reveal information about the operations being executed.

These attacks are particularly potent against embedded and IoT devices, which often have limited processing power and memory, thereby making software-based mitigation techniques inefficient.

VHDL is a powerful language used to describe and simulate digital electronic systems [44–54]. Its ability to model hardware at a low level makes it uniquely suited to implementing SCA countermeasures.

- *Precise control over hardware implementation:* VHDL allows developers to fine-tune the hardware implementation of cryptographic algorithms, thereby enabling the integration of countermeasures directly into the design. This contrasts with software-based approaches, which often lack the granularity needed to effectively mitigate SCAs.
- *Implementation of hardware-level countermeasures:* VHDL provides the means to implement various hardware-level countermeasures, including the following.
    - *Masking:* This technique involves introducing random values (masks) during computations to hide the correlation between the data and the side-channel signal. VHDL allows precise control over the implementation of masking schemes, thereby ensuring their effectiveness.
    - *Hiding:* Hiding techniques aim to make the side-channel signal constant and independent of data. This can be achieved through techniques such as constant power consumption circuits, which are effectively implemented using VHDL.
    - *Dual-rail logic:* This technique encodes each bit with two wires representing both the true and complement values. This helps to balance power consumption and reduce information leakage. VHDL allows precise design and verification of dual-rail logic implementations.
    - *Randomized clocking:* Introducing randomness in a clock signal can disrupt timing-based attacks. VHDL allows the implementation of randomized clock generators and their integration into a cryptographic core.
- *Simulation and verification:* The VHDL allows for thorough simulation and verification of the designed hardware. This is crucial for confirming the effectiveness of the countermeasures implemented before deployment. Simulations can identify potential vulnerabilities and optimize the design of SCA resistance.
- *Flexibility and portability:* VHDL is a standardized language that makes it possible to port designs across different hardware platforms and technologies. This flexibility is essential for IoT devices that often have diverse hardware requirements.

Several real-world applications have demonstrated the effectiveness of VHDL in implementing SCA countermeasures for IoT devices.

- *Secure bootloaders:* VHDL can be used to develop secure bootloaders that incorporate SCA-resistant cryptographic algorithms to authenticate firmware updates and prevent malicious code from being injected into the device.

- *Hardware security modules (HSMs):* VHDL can be used to design HSMs that protect sensitive keys and perform cryptographic operations in a secure environment shielded from SCAs. These HSMs are crucial for protecting communication and data at rest in the IoT devices.
- *Lightweight cryptography:* For resource-constrained IoT devices, VHDL can be used to implement lightweight cryptographic algorithms with built-in SCA countermeasures. These algorithms are designed to be efficient and secure even when implemented with low-power hardware.

Despite the advantages of VHDL, several challenges remain in the development of SCA-resistant IoT devices.

- *Complexity:* Implementing effective SCA countermeasures can significantly increase the complexity of hardware designs.
- *Verification:* Verifying the effectiveness of SCA countermeasures requires specialized tools and expertise.
- *Overhead:* Countermeasures can introduce overhead in terms of the area, power, and performance.

Future research should focus on developing automated tools and methodologies to design and verify SCA-resistant hardware. This includes exploring new countermeasures, formal verification techniques, and efficient hardware architecture [55–65].

Side-channel attacks pose a serious threat to IoT device security. The VHDL offers a powerful and versatile tool for developing and implementing robust countermeasures at the hardware level. By leveraging VHDL's capabilities of VHDLs, developers can create secure and resilient IoT devices that can withstand the ever-evolving landscape of security threats. As IoT continues to grow, the role of VHDL in ensuring the security of connected devices will become increasingly critical.

SHIELDING IoT: DESIGNING VHDL FOR SIDE-CHANNEL ATTACK RESISTANCE

The IoT promises a connected world, but this interconnectedness introduces significant security vulnerabilities. Among the most insidious are SCAs, which exploit unintended information leaked from the physical implementations of cryptographic algorithms. Measuring power consumption, electromagnetic radiation, or timing variations during cryptographic operations can reveal secret keys, even if the encryption algorithm itself is mathematically sound. This is particularly concerning resource-constrained IoT devices, where optimizing performance and power often overshadows the security considerations. This study delves into the design of VHDL programming for implementing side-channel attack countermeasures in IoT security. VHDL, a widely used hardware description language, offers control over the precise hardware implementation of cryptographic algorithms, making it a crucial tool for incorporating SCA protection [66–70].

Before diving into countermeasures, understanding the landscape of SCA threats in the IoT is paramount. Common SCA techniques include the following.

- *Simple power analysis:* SPA directly observes power traces to distinguish between different operations, such as multiplications and additions, revealing the algorithm's execution flow and potentially the key.
- *Differential power analysis:* Statistical analysis of power traces correlated with hypothetical key bits to extract the secret key.
- *Electromagnetic analysis (EMA):* Similar to power analysis, electromagnetic radiation emanating from the device was utilized.
- *Timing attacks:* The time taken for cryptographic operations to complete is measured, exploiting variations dependent on the key.
- *Fault injection attacks:* Faults are introduced into the system during cryptographic operations to force specific errors that reveal information about the key.

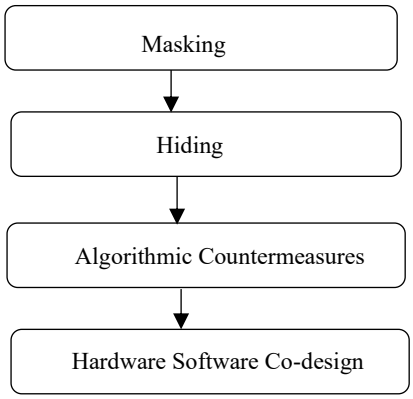

Figure 2. VHDL design for SCA.

IoT devices, which often operate in physically insecure environments, are prime targets for these attacks. Their limited processing power, low-power requirements, and poor physical security make them more vulnerable than traditional computing platforms.

The VHDL provides the granularity needed to implement SCA countermeasures at the hardware level. By carefully controlling the hardware implementation, the leakage can be minimized or rendered statistically insignificant. The key strategies for SCA resistance in VHDL design, as shown in Figure 2, include the following:

- *Masking:* Randomly masks the intermediate values during cryptographic operations, effectively decoupling power consumption from the data being processed. This requires the generation of random numbers and masking operations within the VHDL code.
  - *VHDL implementation:* VHDL's ability to define and manipulate signals to XOR data with a random mask before and after sensitive operations. Special care was required to ensure that the mask was randomly and uniformly distributed.

```
signal masked_data : std_logic_vector(DATA_WIDTH - 1 downto 0);
signal random_mask : std_logic_vector(DATA_WIDTH - 1 downto 0);

process(clk, rst)
begin
   if rst = '1' then
      -- Initialization
   elsif rising_edge(clk) then
      -- Generate a new random mask
      random_mask <= generate_random_mask(); -- Assuming a function generate_random_mask is defined
      -- Mask the data
      masked_data <= input_data XOR random_mask;
      -- Perform operations on the masked data.
      result <= perform_cryptographic_operation(masked_data);
      -- Unmasking results (if required)
      output_data <= result XOR random_mask;
   end if;
end process;
```

- *Hiding:* To ensure that power consumption is independent of the processed data, all operations take the same amount of time and consume the same amount of power, regardless of the input. This involves techniques such as balancing operations, dual-rail logic, and inserting dummy operations.
  - *VHDL implementation:* Careful consideration of the hardware implementation of the cryptographic algorithm. For example, use balanced logic gates, where the number of

transitions is consistent regardless of the data input. VHDL can be used to meticulously control gate-level implementations.

Example: Dummy operation for timing equalization

```
process(clk)
begin
   if rising_edge(clk) then
      if condition then
         -- Perform actual computation
         result <= operation_a(input_data);
      else
         Perform dummy operations with similar complexity.
         result <= dummy_operation(input_data);
      end if;
   end if;
end process;
```

- *Algorithmic countermeasures:* Modifying the cryptographic algorithm to improve SCA resistance. Examples include randomized execution and secure multiplication techniques.
  - *VHDL implementation:* Requires a deep understanding of the cryptographic algorithm and how to implement its SCA-resistant variant in hardware using VHDL. This might involve complex state machines, custom arithmetic units, and careful control of the data flow.
- *Hardware-software co-design:* Optimizing the overall system and distributing tasks between hardware and software to leverage the strengths of each platform. This could involve offloading computationally intensive but less sensitive operations into the software while implementing critical cryptographic functions with hardware-level SCA protections.

Implementing SCA countermeasures in VHDL is not without its challenges:

- *Increased complexity:* SCA-resistant designs are often more complex than their unprotected counterparts, requiring more hardware resources and potentially impacting the performance and power consumption.
- *Verification and validation:* Thorough testing and verification are crucial for ensuring the effectiveness of countermeasures. This requires specialized tools and expertise in SCA analysis. However, traditional functional verification alone is insufficient.
- *Performance overhead:* Countermeasures often introduce performance overhead, which must be carefully balanced against security gains. This is particularly important for resource-constrained IoT devices.
- *Implementation costs:* The design, implementation, and testing of SCA-resistant hardware can be costly and time-consuming.

Securing IoT devices against side-channel attacks is crucial to protect sensitive data and ensure the integrity of the entire ecosystem. VHDL provides a powerful platform for implementing SCA countermeasures at the hardware level. Through careful design and incorporating masking, hiding, and algorithmic techniques, developers can significantly reduce the susceptibility of IoT devices to SCA attacks. However, a holistic approach that considers the trade-offs between security, performance, and cost is essential for deploying effective and practical SCA protection in the real world. As the IoT landscape continues to evolve, investing in hardware security expertise and exploring advanced VHDL design techniques is vital for building truly secure and trustworthy connected devices.

DISCUSSION

IoT is rapidly expanding, connecting billions of devices from smart thermostats to industrial sensors. This interconnectedness brings immense convenience and efficiency but also introduces significant security vulnerabilities. Among these, *side-channel attacks* pose a serious threat by exploiting information leaked through a device's physical characteristics during operation. Fortunately, *VHDL*

programming offers a powerful tool for building robust countermeasures against such attacks at the hardware level.

Traditional security relies on cryptographic algorithms. However, side-channel attacks bypass these algorithms by analyzing the physical emanations from the device, such as

- *Power consumption:* Variations in power consumption during cryptographic operations can reveal secret keys.
- *Electromagnetic radiation*: Similar to power analysis, analyzing electromagnetic emanations can expose sensitive information.
- *Timing analysis:* Measuring the time taken for specific operations can reveal data dependencies and leak the key.
- *Acoustic emissions:* Sound waves emitted by a device can sometimes be correlated with internal computations.

These attacks are often implemented using sophisticated statistical techniques and require physical access to the device, making them particularly concerning in resource-constrained IoT environments, where physical security is often lax.

VHDL is a powerful hardware description language used to model and design digital circuits. By leveraging VHDL, developers can implement countermeasures directly within hardware, making them significantly more resilient to side-channel attacks.

How VHDL Enables Side-Channel Attack Countermeasures

1. Masking
   - *Concept:* Random values (masks) are introduced during cryptographic operations to obscure the correlation between the power consumption and actual data.
   - *VHDL implementation:* VHDL allows precise control and manipulation of data paths. Random number generators (RNGs) can be implemented in VHDL to generate masks. These masks are then applied to sensitive data using bitwise operations, ensuring that the power consumption is dependent on the mask rather than on the actual data.
2. Hiding
   - *Concept:* Ensuring that power consumption is constant regardless of the data being processed.
   - *VHDL implementation:* Techniques such as dual-rail logic and clock gating can be implemented in VHDL to equalize power consumption. The dual-rail logic encodes each bit using two wires, one representing the true value and the other representing the complement. This ensures constant switching activity and, therefore, constant power consumption. Clock gating can be used to selectively disable parts of the circuit that are not currently in use, reducing power consumption and making it less dependent on the processed data.
3. Balancing
   - *Concept:* Circuits are designed where the transitions between logic states are balanced, leading to a more uniform power consumption profile.
   - *VHDL implementation:* Careful VHDL coding practices can minimize data-dependent transitions. This involves optimizing the circuit design to reduce the correlation between data values and signal transitions.
4. Timing Randomization
   - *Concept:* Introducing random delays in the execution of cryptographic operations to disrupt timing analysis attacks.
   - *VHDL implementation:* VHDL allows the insertion of variable delays using techniques such as inserting random numbers of pipeline stages. This makes it difficult for attackers to precisely measure the execution times of specific operations.

5. Hardware Tamper Detection
    - *Concept:* Building mechanisms to detect physical tampering attempts on the device.
    - *VHDL implementation:* VHDL can be used to incorporate sensors and detectors into hardware design. These include sensors for voltage deviations, temperature changes, and physical intrusions. If tampering is detected, the device can be configured to erase sensitive data or even self-destruction.

Benefits of Hardware-Based Countermeasures

- *Increased security:* Hardware-based countermeasures are more difficult to bypass than software-based ones.
- *Performance efficiency:* Countermeasures integrated directly into hardware can often be implemented with minimal performance overhead.
- *Reduced complexity:* Offloading security tasks to hardware can simplify software development and reduce the attack surface.

Challenges and Considerations

- *Design complexity:* Implementing robust side-channel attack countermeasures in VHDL is a complex and challenging task. This requires a deep understanding of both the hardware design and the nuances of side-channel attacks.
- *Verification and validation:* Thorough verification and validation are crucial to ensure that the implemented countermeasures are effective and do not introduce new vulnerabilities.
- *Cost overhead:* Implementing side-channel attack countermeasures can increase the hardware design cost.
- *Resource constraints:* In resource-constrained IoT devices, balancing security with power consumption and performance is crucial.

As IoT devices have become increasingly pervasive, the need for robust security solutions has become more critical. VHDL programming offers a powerful approach for building secure hardware that can withstand sophisticated side-channel attacks. As researchers continue to develop new side-channel attack techniques, continuous innovation in VHDL-based countermeasures is essential for maintaining the security and integrity of the IoT ecosystem. The future lies in developing more sophisticated VHDL-based solutions that are adaptable, power-efficient, and capable of addressing the evolving threat landscape.

By embracing VHDL and its capabilities, IoT manufacturers can significantly enhance the security of their devices and build a more trustworthy and resilient infrastructure.

CONCLUSION

VHDL programming is a valuable tool for implementing countermeasures against side-channel attacks on IoT devices. By providing a means to design and verify hardware circuits at the digital level, VHDL programming allows the implementation of robust and effective countermeasures against various types of side-channel attacks. Moreover, VHDL programming can be used to develop and test hardware designs before implementation, thereby ensuring that they are secure and efficient. As the demand for secure IoT devices increases, the importance of utilizing VHDL programming to implement countermeasures against side-channel attacks cannot be overstated. By incorporating VHDL programming into hardware design and development processes, IoT device manufacturers can improve the security and reliability of their products, earn consumers' trust, and ensure their long-term success.

REFERENCES

1. Chandane ER, Patil A, Bokephode P, Shete A, Kulkarni A, Rajwani D. Landmine detecting robot. Int J Adv Res Sci Commun Technol. 2025;5(1):208-19. doi:10.48175/IJARSCT-22919.

2. Najjaran H, Goldenberg AA. Landmine detection using an autonomous terrain-scanning robot. Industrial Robot: an international journal. 2005 Jun 1;32(3):240–7.
3. Chounde A, Gopnarayan BB, Patil KB, Kamble SS, Khadake SB. Human health care system: A new approach towards life. Grenze Int J Eng Technol. 2024;10(2):5487-94.
4. Dudgikar AB, Ingalgi AA, Jamadar AG, Swami OR, Khadake SB, Moholkar SV. Intelligent battery swapping system for electric vehicles with charging stations locator on IoT and cloud platform. Int J Adv Res Sci Commun Technol. 2023;3:204-8.
5. Jadhav M, Nerkar PM. FPGA-based finger vein recognition system for personal verification. Int J Eng Res Gen Sci. 2015;3:382-8.
6. Kasat K, Shaikh N, Rayabharapu VK, Nayak M, Sayyad Liyakat KK. Implementation and recognition of waste management system with mobility solution in smart cities using Internet of Things. 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India. 2023. p. 1661-5. doi:10.1109/ICAISS58487.2023.10250690.
7. Kazi K. AI-driven IoT (AIIoT) in healthcare monitoring. In: Nguyen TVT, Vo NTM, editors. Using Traditional Design Methods to Enhance AI-Driven Decision Making. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 77-101. doi:10.4018/979-8-3693-0639-0.ch003.
8. Kazi K. Modelling and simulation of electric vehicle for performance analysis: BEV and HEV electrical vehicle implementation using Simulink for E-mobility ecosystems. In: Lakshmi D, Nagpal N, Kassarwani N, Varthanan GV, Siano P, editors. E-Mobility in Electrical Energy Systems for Sustainability. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 295-320. doi: 10.4018/979-8-3693-2611-4.ch014.
9. Kazi KSL, Mahant MA. Machine learning-driven Internet of Things (MLIoT)-based healthcare monitoring system. In: Wickramasinghe N, editor. Digitalization and the Transformation of the Healthcare Sector. Hershey (PA): IGI Global; 2024. p. 205-36. doi:10.4018/979-8-3693-9641-4.ch007.
10. Kazi KSL, Shinde SS, Nerkar PM, Kazi SS, Kazi VS. Machine learning for brand protection: A review of a proactive defense mechanism. In: Khan MI, Amin Ul Haq M, editors. Avoiding AD Fraud and Supporting Brand Safety: Programmatic Advertising Solutions. Hershey (PA): IGI Global; 2025. p. 175-220. doi:10.4018/979-8-3693-7041-4.ch007.
11. Kazi KSL. Advancing towards sustainable energy with hydrogen solutions: Adaptation and challenges. In: Özsungur F, Chaychi Semsari M, Küçük Bayraktar H, editors. Geopolitical Landscapes of Renewable Energy and Urban Growth. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 357-94. doi:10.4018/979-8-3693-8814-3.ch013.
12. Dincer I, Acar C. Smart energy solutions with hydrogen options. Int J Hydrogen Energy. 2018;43:8579-99. doi:10.1016/j.ijhydene.2018.03.120.
13. Kazi KSL. AI-driven IoT (AIIoT)-based decision-making system for high BP patient healthcare monitoring: KSK1 approach for BP patient healthcare monitoring. In: Mzili T, Arya AK, Pamucar D, Shaheen M, editors. Optimization, Machine Learning, and Fuzzy Logic: Theory, Algorithms, and Applications. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 71-102. doi:10.4018/979-8-3693-7352-1.ch003.
14. Kazi KSL. AI-driven-IoT (AIIoT) decision-making system for hepatitis disease patient healthcare monitoring: KSK1 approach for hepatitis patient monitoring. In: Agarwal S, Lakshmi D, Singh L, editors. Navigating Innovations and Challenges in Travel Medicine and Digital Health. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 433-52. doi:10.4018/979-8-3693-8774-0.ch022.
15. Kazi KS. AI-powered IoT (AI IoT) for decision-making in smart agriculture: KSK approach for smart agriculture. In: Hai-Jew S, editor. Enhancing Automated Decision-Making Through AI. Hershey (PA): IGI Global; 2024. p. 67-96. doi:10.4018/979-8-3693-6230-3.ch003.
16. Kazi KSL. AI-driven-IoT (AIIoT)-based Jawar leaf disease detection: KSK approach for Jawar disease detection. In: Bhatti UA, Aamir M, Gulzar Y, Bazai SU, editors. Modern intelligent techniques for image processing. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 34. doi:10.4018/979-8-3693-9045-0.ch019.

17. Kazi KSL. Artificial intelligence (AI)-driven IoT (AIIoT)-based agriculture automation. In: Satapathy S, Muduli K, editors. Advanced Computational Methods for Agri-Business Sustainability. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 72-94. doi:10.4018/979-8-3693-3583-3.ch005.
18. Kazi KSL. Braille-Lippi numbers and characters detection and announcement system for blind children using KSK approach: AI-driven decision-making approach. In: Murugan T, Abirami AM, editors. Driving Quality Education Through AI and Data Science. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 531-56. doi:10.4018/979-8-3693-8292-9.ch023.
19. Kazi KSL. ChatGPT: An automated teacher's guide to learning. In: Bansal R, Chakir A, Ngah AH, Rabby F, Jain A, editors. AI Algorithms and ChatGPT for Student Engagement in Online Learning. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 1-20. doi:10.4018/979-8-3693-4268-8.ch001.
20. Kazi KS. Computer-aided diagnosis in ophthalmology: a technical review of deep learning applications. In: Garcia MB, de Almeida RPP, editors. Transformative Approaches to Patient Literacy and Healthcare Innovation. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 112-135. doi:10.4018/979-8-3693-3661-8.ch006.
21. Kazi KSL. Healthcare monitoring system driven by machine learning and Internet of Medical Things (MLIoMT). In: Kumar VV, Katina PF, Zhao J, editors. Convergence of Internet of Medical Things (IoMT) and Generative AI. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 385-416. doi:10.4018/979-8-3693-6180-1.ch016.
22. Kazi KSL. IoT driven by machine learning (MLIoT) for the retail apparel sector. In: Tarnanidis TK, Papachristou E, Karypidis M, Ismyrlis V, editors. Driving Green Marketing in Fashion and Retail. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 63-81. doi:10.4018/979-8-3693-3049-4.ch004.
23. Kazi KSL. IoT technologies for the intelligent dairy industry: A new challenge. In: Thandekkattu SG, Vajjhala NR, editors. Designing Sustainable Internet of Things Solutions for Smart Industries. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 321-50. doi:10.4018/979-8-3693-5498-8.ch012.
24. Kazi KSL. KK approach for IoT security: T-cell concept. In: Kumar R, Peng SL, Jain P, Elngar AA, editors. Deep Learning Innovations for Securing Critical Infrastructures. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 369-90. doi:10.4018/979-8-3373-0563-9.ch022.
25. Kazi KSL. KK approach to increase resilience in Internet of Things: a T-cell security concept. In: Darwish D, Charan K, editors. Analyzing Privacy and Security Difficulties in Social Media: New Challenges and Solutions. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 87-120. doi:10.4018/979-8-3693-9491-5.ch005.
26. Kazi KSL. Machine learning (ML)-based Braille Lippi characters and numbers detection and announcement system for blind children in learning. In: Sart G, editor. Social Reflections of Human–Computer Interaction in Education, Management, and Economics. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 16-39. doi:10.4018/979-8-3693-3033-3.ch002.
27. Kazi KSL. Machine learning-based pomegranate disease detection and treatment. In: Zia Ul Haq M, Ali I, editors. Revolutionizing Pest Management for Sustainable Agriculture. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 469-498. doi:10.4018/979-8-3693-3061-6.ch019.
28. Kazi KSL. Machine learning-driven Internet of Medical Things (ML-IoMT)-based healthcare monitoring system. In: Soufiene BO, Chakraborty C, editors. Responsible AI for Digital Health and Medical Analytics. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 49-86. doi:10.4018/979-8-3693-6294-5.ch003.
29. Kazi KSL. Transformation of agriculture effectuated by artificial intelligence-driven Internet of Things (AIIoT). In: Garwi J, Dzingirai M, Masengu R, editors. Integrating Agriculture, Green Marketing Strategies, and Artificial Intelligence. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 449-484. doi:10.4018/979-8-3693-6468-0.ch015.
30. Kazi KSL. Machine learning-powered IoT (MLIoT) for retail apparel industry. In: Tarnanidis T, Papachristou E, Karypidis M, Manda VK, editors. Sustainable Practices in the Fashion and Retail

Industry. Hershey (PA): IGI Global Scientific Publishing; 2025. p. 345-372. doi:10.4018/979-8-3693-9959-0.ch015.
31. Keerthana R, Vinutha K, Bhagyalakshmi K, Papinaidu M, Venkatesh V, Liyakat KKS. Machine learning based risk assessment for financial management in big data IoT credit. Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024); 2024 Nov 15. doi:10.2139/ssrn.5086671.
32. Khadake S, Kawade S, Moholkar S, Pawar M. A review of 6G technologies and its advantages over 5G technology. In: Pawar PM, Ronge BP, Gidde RR, Pawar MM, Misal ND, Budhewar AS, More VV, Reddy PV, editors. Techno-Societal 2022: Proceedings of the 4th International Conference on Advanced Technologies for Societal Applications – Volume 1. Cham: Springer; 2024. p. 1043-51. doi:10.1007/978-3-031-34644-6_107.
33. Khadake SB, Dolli SP, Rathod KS, Waghmare MO, Deshpande MA. An overview of intelligent traffic control system using PLC and use of current data of vehicle travels. JNX J Multidiscip Peer Rev J. 2021:1-4.
34. Khadake SB, Kashid PJ, Kawade AM, Khedekar SV, Mallad HM. Electric vehicle technology battery management–review. Int J Adv Res Sci Commun Technol. 2023;3:319-25.
35. Khadake SB, Khedekar SV, Kawade AM, Vyavahare SS, Kashid PJ, Chounde AB, Mallad HM. Solar based electric vehicle charging system-review. Int J Adv Res Sci Commun Technol. 2024;4(2):42-57.
36. Khadake SB, Khedekar SV, Kawade AM, Vyavahare SS, Kashid PJ, Chounde AB, Mallad HM. Solar based electric vehicle charging system-review. Int J Adv Res Sci Commun Technol. 2024;4(2):42-57.
37. Khadake SB, Padavale PV, Dhere PM, Lingade BM. Automatic hand dispenser and temperature scanner for COVID-19 prevention. Int J Adv Res Sci Commun Technol. 2023;3:362-7.
38. Khadake SB, Patil VJ. Prototype design & development of solar based electric vehicle. 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India. 2023. p. 1-7. doi:10.1109/SMARTGENCON60755. 2023.10442455.
39. Khadake SB. Detecting salient objects of natural scene in a video's using spatio-temporal saliency & colour map. Int J Res Publ Eng Technol. 2016;2(8):30-5.
40. Kutubuddin KA, Nerkar PR, Sultanabanu KA. IoT-based skin health monitoring system. Int J Biol Pharm Allied Sci. 2024;13:5937-50.
41. Landage SS, Chavan SR, Kokate PA, Lohar SP, Pawar MK, Khadake SB. Solar outdoor air purifier with air quality monitoring system. Synergies of Innovation: Proceedings of NCSTEM 2023; 2023 Sep; Pandharpur, India. p. 260-266.
42. Rajabi E, Kafaie S. Knowledge graphs and explainable AI in healthcare. Information. 2022 Sep 28;13(10):459. doi:10.3390/info13100459.
43. Mahant MA, Vidyullatha P. Predictive data analytics framework based on child and pregnant women health care systems. Metall Mater Eng. 2025 May 7;31(5):1167–81.
44. Liyakat KKS. Heart health monitoring using IoT and machine learning methods. In: Shaik A, editor. AI-Powered Advances in Pharmacology. Hershey (PA): IGI Global Scientific Publishing; 2024. p. 257-82. doi:10.4018/979-8-3693-3212-2.ch010.
45. Sayyad Liyakat KK, Khadake SB, Tamboli DA, Sawant VA, Mallad HM, Sathe S. AI-driven-IoT (AIIoT) based decision-making–KSK approach in drones for climate change study. 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India. 2024. p. 1735-44. doi:10.1109/ICUIS64676.2024.10866450.
46. Liyakat KK. IoT based boiler health monitoring for sugar industries. Grenze Int J Eng Technol. 2024;10(2):5178-85. Available from: https://thegrenze.com/index.php?display=page&view=journalabstract&absid=3343&id=8
47. Liyakat KKS. Detecting malicious nodes in IoT networks using machine learning and artificial neural networks. 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), 2023 Mar 15-17; Pune, India. 2023. p. 1-5. doi: 10.1109/ESCI56872.2023.10099544.

48. Ramkumar G, Seetha J, Priyadarshini R, Gopila M, Saranya G. IoT-based patient monitoring system for predicting heart disease using deep learning. Measurement. 2023;218:113235. doi:10.1016/j.measurement.2023.113235.
49. Liyakat KKS. Machine learning approach using artificial neural networks to detect malicious nodes in IoT networks. In: Shukla PK, Mittal H, Engelbrecht A, editors. Computer Vision and Robotics. CVR 2023. Algorithms for Intelligent Systems. Singapore: Springer; 2023. p. 123-34. doi: 10.1007/978-981-99-4577-1_3.
50. Hodo E, Bellekens X, Hamilton A, Dubouilh PL, Iorkyase E, Tachtatzis C, et al. Threat analysis of IoT networks using artificial neural network intrusion detection system. 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia. 2016. p. 1–6. doi:10.1109/ISNCC.2016.7746067.
51. Magar SS, Sugandhi AS, Pawar SH, Khadake SB, Mallad HM. Harnessing wind vibration, a novel approach towards electric energy generation-review. Int J Adv Res Sci Commun Technol. 2024;4:73-82.
52. Mulani AO, Bang AV, Birajadar GB, Deshmukh AB, Jadhav HM, Liyakat KK. IoT based air, water, and soil monitoring system for pomegranate farming. Ann Agric Bio Res. 2024;29:71-86.
53. Mulani AO, Sardey MP, Kinage K, Salunkhe SS, Fegade T, Fegade PG. ML-powered Internet of medical things (ML-IoMT) structure for heart disease prediction. J Pharmacol Pharmacother. 2025;16:38–45. doi:10.1177/0976500X241281490.
54. Neeraja P, Kumar RG, Kumar MS, Liyakat KKS, Vani MS. DL-based somnolence detection for improved driver safety and alertness monitoring. 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India. 2024. vol. 5. p. 589-94. doi:10.1109/IC2PCT60090.2024.10486714.
55. Nerkar PM, Liyakat KKS, Dhaware BU, Kazi S. Predictive data analytics framework based on heart healthcare system (HHS) using machine learning. J Adv Zool. 2023;44(Spec Iss 2):3673-86.
56. Nerkar PM, Shinde SS, Liyakat KK, Desai S, Kazi SS. Monitoring fresh fruit and food using IoT and machine learning to improve food safety and quality. Tuijin Jishu/J Propuls Technol. 2023;44:2927-31.
57. Noor AK. Potential of cognitive computing and cognitive systems. Open Eng. 2014;5(1). doi:10.1515/eng-2015-0008.
58. Odnala S, Shanthy R, Bharathi B, Pandey C, Rachapalli A, Liyakat KKS. Artificial intelligence and cloud-enabled E-vehicle design with wireless sensor integration. SSRN Electron J. 2025. doi:10.2139/ssrn.5107242.
59. Parihar B, Kiran A, Valaboju S, Rashid SZ, Liyakat KKS, D R ASL. Enhancing data security in distributed systems using homomorphic encryption and secure computation techniques. ITM Web Conf. 2025;76. doi:10.1051/itmconf/20257602010.
60. Patil VJ, Khadake SB, Tamboli DA, Mallad HM, Takpere SM, Sawant VA. Review of AI in power electronics and drive systems. 2024 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC), Mathura, India. 2024. p. 94-9. doi:10.1109/PARC59193.2024.10486488.
61. Patil VJ, Khadake SB, Tamboli DA, Mallad HM, Takpere SM, Sawant VA. A comprehensive analysis of artificial intelligence integration in electrical engineering. 2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), Lalitpur, Nepal. 2024. p. 484-91. doi:10.1109/ICMCSI61536.2024.00076.
62. Khadake S, Patil VJ, Mallad HM, Gopnarayan BB, Patil KB. Maximize farming productivity through Agriculture 4.0 based intelligence, with use of Agri Tech sense advanced crop monitoring system. Grenze Int J Eng Technol. 2024;10(2):5127-34.
63. Bao R, Hutson A, Madabhushi A, Jonsson VD, Rosario SR, Barnholtz-Sloan JS, Fertig EJ, Marathe H, Harris L, Altreuter J, Chen Q, Dignam J, Gentles AJ, Gonzalez-Kozlova E, Gnjatic S, Kim E, Long M, Morgan M, Ruppin E, Valen DV, Zhang H, Vokes N, Meerzaman D, Liu S, Van Allen EM, Xing Y. Ten challenges and opportunities in computational immuno-oncology. J Immunother Cancer. 2024;12:e009721. doi:10.1136/jitc-2024-009721. PubMed PMID: 39461879.

64. Prasad KR, Karanam SR, Ganesh D, Liyakat KKS, Talasila V, Purushotham P. AI in public-private partnership for IT infrastructure development. J High Technol Manag Res. 2024;35:100496. doi:10.1016/j.hitech.2024.100496.
65. Kralova I, Sjöblom J. Biofuels–renewable energy sources: A review. J Dispersion Sci Technol. 2010;31:409-25. doi:10.1080/01932690903119674.
66. Randive AB, Gaikwad SK, Khadake S, Mallad H. Biodiesel: A renewable source of fuel. Int J Adv Res Sci Commun Technol. 2024 Dec;4(3):225-40. doi:10.48175/IJARSCT-22836.
67. Liyakat KKS, Konnur RG. Vehicle health monitoring system (VHMS) by employing IoT and sensors. Grenze Int J Eng Technol. 2024;10(2):5367-74.
68. Magadum PK. Machine learning for predicting wind turbine output power in wind energy conversion systems. Grenze Int J Eng Technol. 2024;10(1):2074-80.
69. Upadhyaya AN, Surekha C, Malathi P, Suresh G, Suriyan K, Liyakat KKS. Pioneering cognitive computing for transformative healthcare innovations. SSRN Electron J. 2025. doi:10.2139/ssrn.5086894.
70. Veena C, Sridevi M, Liyakat KKS, Saha B, Reddy SR, Shirisha N. HEECCNB: An efficient IoT-cloud architecture for secure patient data transmission and accurate disease prediction in healthcare systems. 2023 Seventh International Conference on Image Information Processing (ICIIP), Solan, India. 2023. p. 407-10. doi:10.1109/ICIIP61524.2023.10537627.

