WIICT2024

Proceedings of the Workshop on Innovation on Information and Communication Technologies 2024

Institute
ITACA
Information and Communication Technologies

Editors:
Dr. Carlos Fernandez-Llatas
Dra. Maria Guillen

Committees

Organizing Committee

- **Chair:**Maria Guillem, Universitat Politècnica de València, Spain
- Alberto Bonastre, Universitat Politècnica de València, Spain
- Jose Manuel Catala, Universitat Politècnica de València, Spain
- Carlos Fernández-Llatas, Universitat Politècnica de València, Spain
- Jose Mariano Dahoui, Universitat Politècnica de València, Spain

Scientific Committee

- **Chair**: Carlos Fernandez-Llatas, Universitat Politècnica de València, Spain
- Paulo de Carvalho, University of Coimbra, Portugal
- Anna-Maria Bianchi, Politecnico di Milano, Italy
- Jorge Munoz-Gama, Pontificia Universidad Catolica de Chile
- Fernando Seoane, Karolinska Institutet
- Jorge Henriques, University of Coimbra, Portugal
- Cenk Demiroglu, Ozyegin University, Turkey
- Johan Gustav Bellika, National Center of Telemedicine, Norway
- Yigzaw Kassaye Yitbarek, University of Tromso, Norway
- Raymundo Barrales, Universidad Autónoma Metropolitana de México
- Onur Dogan, Istanbul Technical University,, Turkey
- Frank Y. Li, University of Agder, Norway
- Pilar Sala, MySphera., Spain
- Alvaro Martinez, MySphera, Spain
- Jose Carlos Campelo, Universitat Politècnica de València, Spain
- Juan Vicente Capella, Universitat Politècnica de València, Spain
- Antonio Mocholí, Universitat Politècnica de València, Spain
- Sara Blanc, Universitat Politècnica de València, Spain
- Juan-Carlos Baraza-Calvo, Universitat Politècnica de València, Spain
- Juan Carlos Ruiz, Universitat Politècnica de València, Spain
- Joaquin Gracia, Universitat Politècnica de València, Spain
- David De Andres, Universitat Politècnica de València, Spain
- Ricardo Mercado, Universitat Politècnica de València, Spain
- Lenin Lemus, Universitat Politècnica de València, Spain
- Vicente Traver, Universitat Politècnica de València, Spain
- Antonio Martinez-Millana, Universitat Politècnica de València, Spain
- Jose Luis Bayo-Monton, Universitat Politècnica de València, Spain
- Juan Miguel García-Gomez, Universitat Politècnica de València, Spain
- Elies Fuster, Universitat Politècnica de València, Spain
- Carlos Saez, Universitat Politècnica de València, Spain
- Ángel Perles, Universitat Politècnica de València, Spain
- Luis José Saiz Adalid, Universitat Politècnica de València, Spain
- Gema Ibañez, Universitat Politècnica de València, Spain
- Pedro Yuste, Universitat Politècnica de València, Spain
- Daniel Gil Tomàs, Universitat Politècnica de València, Spain
- Sabina Asensio, Universitat Politècnica de València, Spain
- Beatriz Garcia-Baños, Universitat Politècnica de València, Spain
- Francisco Castells, Universitat Politècnica de València, Spain
- Diogo Nunes, University of Coimbra, Portugal
- Adriana Leal, University of Coimbra, Portugal
- Zoe Valero Ramón, Universitat Politècnica de València, Spain
- Yolanda Vives, Universitat Politècnica de València, Spain
- Javier Urchueguia, Universitat Politècnica de València, Spain

- Jose Vicente Oliver. Universitat Politècnica de València, Spain
- Victoria Lerma, Universitat Politècnica de València, Spain
- Edgar Lorenzo, Universitat Politècnica de València, Spain

Table of Contents

Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT 2024). A Preface

Carlos Fernandez-Llatas[1][0000−0002−2819−5597] and Maria Guillen[1]

Instituto ITACA, Universitat Politecnica de Valencia

In recent years, the field of technology has witnessed an exponential growth in the development and application of advanced computational techniques, particularly in areas such as wireless sensor networks (WSNs), machine learning, and fault-tolerant embedded systems. These advancements have not only revolutionized traditional industries but have also introduced new challenges and opportunities for enhancing efficiency, security, and sustainability. This book presents a collection of papers that address some of the most pressing issues in these domains, providing a comprehensive overview of current trends, methodologies, and solutions.

The papers included in this volume were presented at the ITACA-WIICT'24, a workshop that serves as a meeting forum for scientists, technicians, and professionals dedicated to the study and research of Information and Communication Technologies (ICT). The primary goal of this workshop is to foster collaboration and the exchange of ideas among participants, promoting technological transfer and cooperation between academia and industry.

The first paper, "Enhancing attack detection in Wireless Sensor Networks: definition of a specialized dataset", delves into the security challenges faced by WSNs. This research highlights the vulnerabilities of these networks to Denial of Service (DoS) attacks and emphasizes the importance of anomaly detection techniques. By focusing on the LEACH protocol, a widely-used hierarchical routing protocol, the study develops a specialized dataset to enhance the detection and classification of various DoS attacks, including Blackhole, Grayhole, Flooding, and Sinkhole attacks. The paper also explores the role of machine learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) in improving the security of WSNs and mitigating these threats.

The second paper, "Improving the efficiency of Matrix Codes using Hsiao Codes", addresses the growing concern of memory system reliability in embedded systems, particularly in the context of Single and Multiple Cell Upsets (SCUs and MCUs). As the continuous scaling down of CMOS technology increases the fault rate in SRAM memories, the need for robust Error Correction Codes (ECCs) becomes paramount. This study proposes the use of Hsiao codes in Matrix code structures to reduce area, power, and delay overheads while maintaining error detection and correction capabilities. The evaluation of these ECCs, through fault injection and VHDL synthesis, demonstrates the potential of Hsiao-based

C.Fernandez-Llatas and M.Guillen (Ed.):
Workshop on Innovation on
Information and Communication Technologies
(ITACA-WIICT 2024)

ISBN 978-84-09-66630-0

Matrix codes in improving the efficiency and reliability of memory systems in embedded environments.

In the third paper, "Analysis of the impact of faults in a Convolutional Neural Network implemented in a Raspberry Pi", the focus shifts to the reliability of neural networks deployed on embedded systems, particularly in resource-constrained devices such as the Raspberry Pi. The paper investigates how faults in the memory, specifically bit-flips and stuck-at faults, impact the behavior of Convolutional Neural Networks (CNNs). Through a series of fault injection campaigns, the study analyzes the vulnerability of different layers of the network and identifies which bits are more susceptible to causing mispredictions. The results provide valuable insights into the design of fault-tolerant neural networks and highlight the importance of addressing reliability issues in embedded systems running memory-intensive applications.

The final paper, "In situ dielectric characterization as a tool towards more sustainable industrial processes", explores the application of microwave dielectric thermal analysis (MW-DETA) in improving the sustainability of energy-intensive industrial processes. By analyzing the dielectric properties of materials in real-time, this technique enables the optimization of processes such as the synthesis of ceramic pigments and the recycling of steel industry wastes. The study demonstrates how MW-DETA can reduce reaction temperatures, simplify raw material mixtures, and ultimately minimize resource consumption. This approach not only enhances process efficiency but also contributes to lowering the environmental impact of industrial operations.

Each of these papers represents a significant contribution to its respective field, reflecting the interdisciplinary nature of modern technological challenges. Collectively, they offer valuable insights into the development of innovative solutions for enhancing security, reliability, and sustainability in embedded systems, WSNs, and industrial processes.

We hope that this collection of papers will inspire further research and collaboration across multiple disciplines, ultimately contributing to the advancement of technology in ways that benefit both industry and society.

Enhancing attack detection in Wireless Sensor Networks: definition of a specialized dataset

Amal Chaffai, José Carlos Campelo, Alberto Miguel Bonastre

ITACA Institute, Universitat Politècnica de València
Edificio 8G Ciudad Politécnica de la Innovación, Camí de Vera, s/n, 46022 Valencia, Spain
amcha@doctor.upv.es; {jcampelo, bonastre}@itaca.upv.es

Abstract. Wireless Sensor Networks (WSNs) are integral to many modern applications, ranging from environmental monitoring to smart cities. However, they are susceptible to various security threats, particularly Denial of Service (DoS) attacks. This paper explores the landscape of DoS attacks on WSNs and proposes the development of a specialized dataset to better detect and classify four types of DoS attacks: Blackhole, Grayhole, Flooding, and Sinkhole. The definition of a specialized dataset will allow both the comparison between different intrusion detection systems and the application of learning techniques to enhance WSN security. By considering the LEACH protocol, one of the most popular hierarchical routing protocols in WSNs, we focus on identifying data anomalies. Additionally, we discuss various anomaly detection methods. The most relevant DoS attacks are analyzed and how the information of interest can be extracted for analysis, cataloging, and detection is described.

1 Introduction

Wireless Sensor Networks (WSNs) present unique security challenges due to their inherent characteristics, including resource constraints, dynamic topology, and susceptibility to physical attacks [1,2]. These challenges encompass various aspects of security. Including resource constraints since sensor nodes in WSNs are typically constrained in terms of processing power, memory, and energy. Securing communication and data processing while operating within these constraints presents significant challenges. Moreover dynamic topology, WSNs often operate in dynamic environments where nodes may join or leave the network unpredictably. This dynamic topology introduces vulnerabilities and complicates security management. Additionally vulnerability to Physical Attacks, the deployment of sensor nodes in unattended or hostile environments exposes WSNs to physical attacks, such as node capture, tampering, and compromise.

Denial of Service (DoS) attacks pose a significant threat to WSNs, aiming to disrupt network operations by overwhelming nodes or communication channels.

C.Fernandez-Llatas and M.Guillen (Ed.):
Workshop on Innovation on
Information and Communication Technologies
(ITACA-WIICT 2024)

ISBN 978-84-09-66630-0

Furthermore, addressing DoS attacks in WSNs requires robust detection and mitigation strategies to ensure network resilience and reliability. Machine learning techniques, particularly neural networks (NN), can play a crucial role in enhancing network security in WSNs. These techniques offer capabilities for anomaly detection, where Machine learning algorithms can analyze network traffic patterns to identify anomalies indicative of DoS attacks or other security breaches. Also, intrusion detection by learning from historical data, machine learning models can detect and classify suspicious activities or behaviors, enabling proactive responses to security threats. As well the mitigation strategies of Machine learning algorithms can aid in developing adaptive defense mechanisms that dynamically adjust to evolving threats and protect WSNs against DoS attacks and other security vulnerabilities. Integrating machine learning into WSN security frameworks enhances the ability to detect, respond to, and mitigate security threats effectively [15].

This combined overview provides a comprehensive understanding of the security challenges faced by WSNs, with a specific focus on DoS attacks and the role of machine learning in addressing these challenges. The LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol, one of the most popular hierarchical routing protocols in WSNs, is considered in this study. By examining the different types of anomalies that can occur within WSNs, particularly data anomalies, we aim to identify effective anomaly detection methods.

The rest of paper is organized as follows. Section 2 shows related works. Section 3 provides an overview of LEACH protocol. Section 4 details the essential features to build the dataset. Section 5 presents the attack models and Section 6 outlines the experimentation. Finally, conclusions are presented in Section 7.

2 Related Work

Denial of Service (DoS) attacks represent a significant threat to the security of Wireless Sensor Networks (WSNs), largely due to their relative ease of execution [12]. In recent years, researchers have increasingly turned to machine learning techniques for the detection and mitigation of such attacks. This section reviews some notable works in this domain, highlighting the efficacy of machine learning models in identifying DoS attacks in WSNs. Kim et al. [6] proposed a model based on a Convolutional Neural Network (CNN) for detecting DoS attacks using datasets such as KDD-99 and CICIDS2018. By treating input features as "images" and leveraging CNNs, their model achieved high accuracy rates exceeding 99% in both binary (normal vs. attack) and multiclass classification scenarios. Wu et al. [2] introduced LuNet, a hierarchical CNN+RNN neural network architecture, trained on datasets like NSL-KDD and UNSW-NB15. LuNet employs multiple levels of CNN and RNN to jointly learn from input data, achieving impressive accuracies of up to 99.36% and 99.05% in binary and multiclass classification tasks, respectively. Almomani et al. [3] investigated the effectiveness of eight different machine learning models, including Naive Bayes, Decision Trees, Random Forests, and Support Vector Machine, in detecting DoS attacks using a WSN-DS dataset. Notably, the Random Forest algorithm outperformed other models with a True Positive rate of 99.7%. Park et al. [7] proposed a Random Forest classifier to detect various types of DoS attacks using the WSN-DS dataset. Their model achieved high F1-scores and an overall accuracy of 97.8% across different attack types. Wazirali and Ahmad [8] evaluated machine learning classification algorithms for detecting flooding, gray hole, and black hole DoS attacks in WSNs. Their findings revealed that the J48 approach is the most accurate and efficient method for identifying gray hole and black hole attacks, while the Random Tree method excels in detecting flooding assaults. Despite the advancements in machine learning-based detection techniques, the absence of a public-domain specialized dataset for WSNs poses a significant challenge. Therefore, there is an urgent need to develop a labeled, specialized dataset that accurately characterizes WSN behaviors, both normal and anomalous.

3 Overview of LEACH protocol

The Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol is a popular hierarchical routing protocol designed specifically for Wireless Sensor Networks (WSNs). It aims to prolong network lifetime by minimizing energy consumption, a critical concern in resource-constrained sensor nodes. LEACH operates based on the principle of clustering, dividing the network into clusters and electing cluster heads (CHs) to facilitate data aggregation and transmission.

LEACH is different from conventual static clustering because the Cluster Head(CH) and Clusters are not fixed. The basic idea can be broken down in 3 steps and it is repeated every round. The procedure is: i) Election of CH, ii) Formation of Clusters and iii) Data transfer to CH, then from CH to Sink.

i) Election of CH:

Each node in network selects randomly a number between zero and one. If that number is less than a set threshold, then this node becomes a cluster head:

$$T(n) = \begin{cases} \dfrac{p}{1 - p * \left(r \bmod \dfrac{1}{P}\right)} & if\ n\ \in G \\ 0 & otherwise \end{cases}$$

where

- p = desired percentage of cluster head within network.
- r = current round.
- G = set of nodes that have not been CH in last 1/P rounds.
- n = is the node.

ii) Cluster Formation:

After each node decides whether to become a CH for a current round based on the previous probabilistic model, the remaining nodes join the nearest CH.

iii) Data Aggregation:

Cluster heads collect data from member nodes within their respective clusters and perform aggregation to reduce redundant transmissions and conserve energy. Aggregated data is then forwarded to the base station or sink node via multi-hop communication.

To distribute energy consumption evenly among nodes and prevent premature node failure, LEACH employs a rotational scheme where cluster heads are rotated periodically. This rotation helps mitigate the energy imbalance issue commonly observed in WSNs and prolongs network lifetime. Moreover, LEACH is designed to adapt to dynamic network conditions such as node failures, changes in topology, and varying energy levels. It employs mechanisms to handle cluster head failures by electing new CHs and redistributing cluster memberships as needed.

4 The essential features to build the dataset

We first need to define the types of features commonly used in Wireless Sensor Networks (WSNs) for detecting and classifying Denial of Service (DoS) attacks. Here's a general overview of the types of features that will be extracted from WSN data:

Traffic Features: these include various metrics related to network traffic, such as packet rate, packet size distribution, and packet inter-arrival times. Analysis of traffic features can reveal abnormalities indicative of DoS attacks, such as sudden spikes in traffic volume or irregular patterns. As well Network Topology Features describe the spatial arrangement and connectivity of sensor nodes within the network. Examples include node density, neighbor relationships, and network diameter. Changes in network topology may signal the presence of attack-induced node failures or compromised nodes. Additionally Communication Features capture communication patterns and protocols used within the network. For instance, metrics like packet loss rates, retransmission rates, and communication latency can provide insights into the integrity and efficiency of data transmission. Anomalies in communication features may indicate DoS attacks targeting network communication channels. And Energy Consumption Features given the resource-constrained nature of WSNs, monitoring energy consumption is crucial for detecting abnormal behaviors and potential attacks. Features related to energy usage, such as battery voltage levels, energy consumption rates, and power-saving mode activations, can help identify energy-draining DoS attacks or compromised nodes. Moreover Data Content Features pertain to the actual sensor data collected by nodes within the network. Depending on the application domain, data content features may include environmental parameters, event occurrences, or sensor readings. Deviations from expected data patterns or values may indicate data manipulation or injection attacks. As well as Temporal and Spatial Features capture variations in network behavior over time, while spatial features characterize spatial distributions and correlations within the network. Analysis of temporal and spatial features enables the detection of coordinated attacks or localized anomalies that may evade traditional detection methods. Accordingly Statistical and Machine Learning-based Features, from statistical analysis or machine learning algorithms can provide additional insights into network behavior and aid in anomaly detection. Examples include statistical moments (mean, variance, skewness), frequency domain features (Fourier transform coefficients), and feature representations learned from deep learning models.

By extracting and analyzing these diverse sets of features from the constructed dataset, the research can develop robust detection and classification models capable of accurately identifying various types of DoS attacks in WSNs. The selection and engineering of features play a crucial role in the effectiveness and generalization ability of machine learning-based security solutions for WSNs.

5 Attacks models

To address various types of attacks in Wireless Sensor Networks (WSNs), researchers have developed different models and algorithms tailored to detect and mitigate specific attack vectors [3,4,5]. Here are some common attack models in WSNs along with corresponding detection or mitigation techniques as illustrated in Fig.1.

Blackhole Attack:

- Attack Description: Malicious nodes advertise themselves as having the shortest paths to the sink, attracting traffic that they drop or manipulate, leading to data loss [9,10].
- Detection/Mitigation Techniques:
  Trust-based mechanisms: Nodes maintain trust levels for their neighbors and avoid routing data through untrusted nodes.
  Watchdog mechanisms: Nodes monitor their neighbors' behavior and detect anomalies such as data dropping.

Intrusion detection systems: Algorithms analyze network traffic patterns to identify deviations indicative of blackhole attacks.

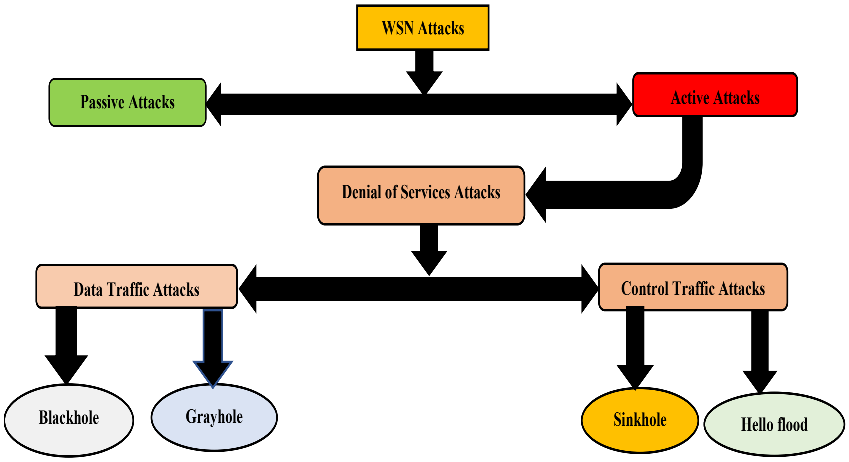

Figure 1. WSN attacks.

Grayhole Attack:

- Attack Description: Malicious nodes selectively drop or modify packets, causing disruptions without being immediately detected.
- Detection/Mitigation Techniques:
  Packet authentication and verification: Nodes use cryptographic techniques to verify packet integrity and authenticity.
  Secure routing protocols: Protocols employ secure routing mechanisms to ensure data delivery even in the presence of malicious nodes.
  Redundancy-based schemes: Data is replicated and sent through multiple paths to mitigate the impact of packet loss by grayhole nodes.

Sinkhole Attack:

- Attack Description: Malicious node attracts all the traffic in a network by advertising itself as the shortest or most efficient route. Once the traffic is redirected, the attacker can perform various malicious activities, such as dropping packets (blackholing), eavesdropping on communications, or selectively forwarding packets to disrupt the network's normal operation [14].
- Detection/Mitigation Techniques:

Traffic Pattern Analysis: Continuously monitor network traffic for abnormal routing patterns. Algorithms can detect if a single node is disproportionately attracting traffic, which might indicate a sinkhole attack.
Authentication and Integrity Checks:
Node Authentication: Implement strong authentication mechanisms to ensure that only legitimate nodes can participate in the network. Use cryptographic techniques to authenticate routing updates.

Flooding Attack:

- Attack Description: Attackers flood the network with excessive traffic, consuming network resources and rendering legitimate communication impossible [11].
- Detection/Mitigation Techniques:
  Traffic analysis: Algorithms monitor network traffic patterns and identify abnormal spikes in traffic volume.
  Rate limiting: Nodes enforce strict rate limits on incoming traffic to prevent network congestion caused by flooding attacks.
  Packet filtering: Nodes discard duplicate or suspicious packets to conserve bandwidth and prevent resource depletion.

6 Experimentation

Design the Data Collection Framework

- Tools and Platforms: Use of simulation tools like NS-3 [13] or real-world testbeds to generate data.
- Real-time Data Collection: Testbeds like IoT-LAB
- Data Labeling Tools: Labelbox, Snorkel
- Preprocessing and Analysis: Python (Pandas, NumPy), R

Dataset Construction:

We will construct a specialized dataset tailored to characterize various types of Denial of Service (DoS) attacks in Wireless Sensor Networks (WSNs). The dataset will comprise simulated network traffic and sensor data generated using a network simulator, incorporating realistic network topologies and attack scenarios. We will design the dataset to include representative instances of Blackhole, Grayhole, Flooding, and Scheduling attacks, as well as normal network behavior for comparison.

Feature Extraction:

From the constructed dataset, we will extract a comprehensive set of features to capture different aspects of network behavior and attack patterns. These features will include traffic metrics, network topology characteristics, communication patterns, energy consumption profiles, and data content attributes. Additionally, statistical and machine learning-based features will be derived from the raw data to enhance the discriminative power of our models.

Experimental Setup:

We will conduct our experiments in a controlled laboratory environment using standard WSN simulation tools and machine learning libraries. The experiments will be carried out on a computing platform equipped with adequate computational resources to handle the dataset size and model training requirements efficiently.

Model Evaluation:

We will evaluate the performance of various machine learning models for detecting and classifying DoS attacks in WSNs using the constructed dataset. The evaluated models include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs).

Performance Metrics:

To assess the effectiveness of the models, we will employ standard performance metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics provide insights into the models' ability to correctly classify different attack types while minimizing false positives and false negatives.

7 Conclusions

Our study contributes to the field of Wireless Sensor Networks (WSNs) security by addressing the detection and classification of Denial of Service (DoS) attacks.

This is a first step in the design of techniques to improve WSN attack detection capabilities. With artificial intelligence techniques deployed in all areas of our life, their application in WSN environments require a specialized and labeled dataset that will allow us to perform the training. In addition, the data set will allow comparisons to be made between different developments. This way, feature extraction and selection play a critical role in the performance of machine learning models for WSN security. Thus, in this proposal, we have shown the guidelines to extract the information of interest from the most representative attacks, starting from a real but simulated workload of the operation of a WSN to which the attack models are added. We found that incorporating diverse sets of features, including traffic metrics, network topology characteristics, and data content attributes, enhances the models' ability to discern between normal and anomalous behaviors.

In summary, our study lays the groundwork for advancing the state-of-the-art in WSN security through interdisciplinary collaboration and innovative research initiatives. By addressing the identified challenges, we can create more resilient and secure WSN systems capable of supporting a wide range of critical applications in the future.

References

1. N. Marriwala and P. Rathee, “An approach to increase the wireless sensor network lifetime,” in Proceedings of the World Congress on Information and Communication Technologies (WICT ’12), pp. 495–499, IEEE, Trivandrum, India, OctoberNovember 2012.
2. V. C. Gungor, B. Lu, and G. P. Hancke, “Opportunities and challenges of wireless sensor networks in smart grid,” IEEE Transactions on Industrial Electronics, vol. 57, no. 10, pp. 3557– 3564, 2010.
3. M. A. Rassam, M. A. Maarof, and A. Zainal, “A survey of intrusion detection schemes in wireless sensor networks,” American Journal of Applied Sciences, vol. 9, no. 10, pp. 1636–1652, 2012.
4. I. Butun, S. D. Morgera, and R. Sankar, “A survey of intrusión detection systems in wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 266–282, 2014.
5. H. Modares, R. Salleh, and A. Moravejosharieh, “Overview of security issues in wireless sensor networks,” in Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSim ’11), pp. 308–311, September 2011.
6. Kim J, Kim J, Kim H, Shim M, Choi E. Cnn-based network intrusion detection against denial-of-service attacks. Electronics. 2020;9(6):916
7. Park T, Cho D, Kim H, et al: An efective classifcation for dos attacks in wireless sensor networks. In: 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), 2018; pp. 689–692. IEEE.
8. Wazirali R, Ahmad R. Machine learning approaches to detect dos and their efect on wsns lifetime. CMC-Comput Mat Contin. 2021;70(3):4921–46.
9. M. Tripathi, M. S. Gaur, and V. Laxmi, “Comparing the impact of black hole and gray hole attack on LEACH in WSN,” Procedia Computer Science, vol. 19, pp. 1101–1107, 2013.
10. I. Kaushik, N. Sharma, and N. Singh, “Intrusion detection and security system for blackhole attack,” in Proceedings of the 2nd International Conference on Signal Processing and Communication (ICSPC), pp. 320–324, Coimbatore, India, March 2019.
11. Bilgili, S.; Demir, A.K.; Alam, S. IfNot: An approach towards mitigating interest flooding attacks in Named Data Networking of Things. Internet Things 2024, 25, 101076.
12. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of Things: Security and Solutions Survey. Sensors 2022, 22, 7433
13. The NS-3 Network Simulator. (Accessed on May 2024). Available online: http://www.nsnam.org
14. R. S. Raghav, S. Pothula, and D. Ponnurangam, “An enriched artificial bee colony (EABC) algorithm for detection of sinkhole attacks in Wireless Sensor Network,” Int. J. Mech. Eng. Technol., vol. 8, no. 8, pp. 193–202, 2017.
15. Zhang Q, Zhang W. 2019. Accurate detection of selective forwarding attack in wireless sensor networks. International Journal of Distributed Sensor Networks 15(1) DOI 10.1177/1550147718824008.

Improving the efficiency of Matrix Codes using Hsiao Codes

J. Gracia-Morán[1], L.J. Saiz-Adalid[1], J.C. Baraza-Calvo[1],
D. Gil-Tomás[1], P.J. Gil-Vicente[1]

[1] Instituto ITACA, Universitat Politècncia de València
Camino de Vera, s/n, 46022 Valencia, Spain
{jgracia, ljsaiz, jcbaraza, dgil, pgil}@itaca.upv.es
https://gstf.blogs.upv.es/

Abstract. With the continuous size reduction of CMOS technology, faults suffered by SRAM memory systems increase. In this way, the probability of occurrence of Multiple Cell Upsets (MCUs), in addition to Single Cell Upsets (SCUs), augments. Thus, Fault Tolerance Mechanisms (FTMs) are needed to protect memory systems. Traditionally, Error Correction Codes (ECCs) are a family of FTMs that have been used to protect memories. Nevertheless, an aspect that must be considered when an ECC is added to a computer system is the area, delay, and power consumption overheads that encoder and decoder circuits introduce. Matrix codes, used as a basis for different ECCs, are an example of common FTMs to cope with MCUs. These codes are based on Extended Hamming codes and parity checks. Nevertheless, they incur in substantial area, power, and delay overheads.

In this work, we present a series of Matrix codes based on Hsiao codes that reduce overhead by improving error coverage. Also, we evaluate the global goodness of the ECCs by using a metric that includes the most important factors.

1 Introduction

Nowadays, the continued physical feature size downscaling of CMOS technology provides RAM memory systems with a great storage capacity. Nevertheless, this size decrease has also caused an augment in the memory fault rate [1]. With the present aggressive scaling, the energy needed to provoke a Single Event Upset (SEU) in storage has been reduced. This energy reduction can provoke Multiple Cell Upsets (MCUs) in addition to traditional Single Cell Upsets (SCUs) [2][3]. This downscaling is especially problematic in spatial systems, as this is an aggressive environment subjected to the impact of high-energy cosmic particles [4][5].

Usually, when a cosmic particle hits a memory cell, it produces a flow of electron-hole pairs along the transport track [6]. In this way, *adjacent errors* can be generated, that is, multiple errors where all the erroneous bits are contiguous [7][8].

Thus, Fault Tolerance Mechanisms (FTMs) are needed to tolerate such faults. FTMs used in memories are frequently based on Error Correction Codes (ECCs). Common ECCs used to protect standard memories are Single Error Correction (SEC)

C.Fernandez-Llatas and M.Guillen (Ed.):
Workshop on Innovation on
Information and Communication Technologies
(ITACA-WIICT 2024)

ISBN 978-84-09-66630-0

codes or Single Error Correction-Double Error Detection (SEC-DED) codes. SEC codes can correct an error in one single memory cell. On the other hand, SEC-DED codes can correct an error in one single memory cell, as well as detect two errors in two independent cells [9][10][11]. In critical applications, more complex and sophisticated codes are used [12][13][14][15][16][24]. For instance, Matrix code [15] is a well-known ECC that has been used as a basis for different ECCs [16][17][18][19]. Matrix code combines Hamming codes with parity checks in a two-dimensional scheme, allowing the correction of two data bits in error.

Nevertheless, when ECCs are introduced in memory systems, a series of redundant bits are added. These extra bits are used to detect and/or correct the possible errors produced. In this way, the inclusion of these extra bits implies an overhead in the area, power, and delay consumed by the ECC circuitry.

It has been shown that it is feasible to reduce overheads by using SEC-DED Hsiao code [20] instead of SEC-DED Hamming code. This is possible by using a careful design of the parity check matrix that defines the code [21][22]. But, is this true when Hsiao code is used in a Matrix code? That is, can we reduce overhead in Matrix codes by using Hsiao codes? To what extent? Is this true for any data word length?

We answer these questions in this work. The idea is to use Hsiao codes to improve Matrix schemes. We have designed and implemented a series of Matrix codes that use SEC-DED Hamming codes or Hsiao codes plus parity checks. Then, we have injected faults to check the detection and correction properties of these codes. Also, we have synthesized them to measure area, power, and delay consumption. Finally, to do a global ECC evaluation, we have used a metric that considers area, power, and delay overheads, as well as error coverage and redundancy of each code [23].

This work is organized as follows. Section 2 summarizes the design of ECCs, including Matrix and Hsiao codes. Section 3 presents the evaluation of the different ECCs implemented, and Section 4 concludes the paper.

2 Introduction to the design of Error Correction Codes

2.1 Basics on coding theory

An (n, k) binary ECC encodes a k-bit input word in an n-bit output word [24]. The input word $\mathbf{u}=(u_0, u_1, ..., u_{k-1})$ is a k-bit vector which represents the original data. The codeword $\mathbf{b}=(b_0, b_1, ..., b_{n-1})$ is a vector of n bits, where the $(n - k)$ redundant bits added are called parity or code bits. **b** is transmitted across an unreliable channel which delivers the received word $\mathbf{r}=(r_0, r_1, ..., r_{n-1})$. The error vector $\mathbf{e}=(e_0, e_1, ..., e_{n-1})$ models the error induced by the channel. If no error has occurred in the ith bit, e_i=0; otherwise, e_i =1. In this way, **r** can be interpreted as $\mathbf{r} = \mathbf{b} \oplus \mathbf{e}$. Fig. 1 synthesizes this encoding, channel crossing and decoding process.

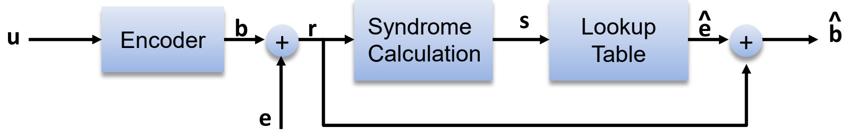

Fig. 1. Encoding, channel crossing and decoding process

The parity check matrix $\mathbf{H}_{(n-k)\times n}$ of a linear block code defines the code [9]. For the encoding process, **b** must accomplish the requirement $\mathbf{H}\cdot\mathbf{b}^{T}=\mathbf{0}$. For syndrome decoding, the syndrome is defined as $\mathbf{s}^{T}=\mathbf{H}\cdot\mathbf{r}^{T}$, and it exclusively depends on **e**:

$$\mathbf{s}^{T} = \mathbf{H}\cdot\mathbf{r}^{T} = \mathbf{H}\cdot(\mathbf{b} \oplus \mathbf{e})^{T} = \mathbf{H}\cdot\mathbf{b}^{T} \oplus \mathbf{H}\cdot\mathbf{e}^{T} = \mathbf{H}\cdot\mathbf{e}^{T} \tag{1}$$

There must be a different **s** for each correctable **e**. If **s**=**0**, we can assume that **e**=**0**. Therefore, **r** is correct. Otherwise, an error has occurred. Syndrome decoding is performed by addressing a lookup table that relates each **s** with the estimated error vector $\hat{\mathbf{e}}$. The decoded codeword $\hat{\mathbf{b}}$ is calculated as $\hat{\mathbf{b}} = \mathbf{r} \oplus \hat{\mathbf{e}}$. As the code is separable, from $\hat{\mathbf{b}}$ it is easy to obtain $\hat{\mathbf{u}}$ just discarding the parity bits. If the fault hypothesis employed to design the ECC is consistent with the channel behavior, $\hat{\mathbf{u}}$ and **u** must be equal with a very high probability.

2.2 Matrix codes

SEC-DED (Single Error Correction-Double Error Detection) Extended Hamming codes [10] can correct single bit errors and detect double random errors.

As just commented, a code can be defined by a parity check matrix. Specifically, in this paper we have used the parity check matrix shown in (2) to define the SEC-DED Extended Hamming code used in the Matrix code, where Xi are the data bits and Cj are the horizontal check bits. This parity check matrix has been extracted from [15].

$$\mathbf{H} = \begin{array}{c} \begin{array}{ccccccccccccc} C_0 & C_1 & \cdots & \cdots & C_4 & X_0 & X_1 & \cdots & \cdots & \cdots & \cdots & \cdots & X_7 \end{array} \\ \left|\begin{array}{ccccccccccccc} 1&0&0&0&0&1&1&0&1&1&0&1&0 \\ 0&1&0&0&0&1&0&1&1&0&1&1&0 \\ 0&0&1&0&0&0&1&1&1&0&0&0&1 \\ 0&0&0&1&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&1&1&1&1&1&1&1 \end{array}\right| \end{array} \tag{2}$$

From **H** matrix, it is possible to obtain the encoding and decoding formulas used by the ECC. Specifically, by using the formulas shown in (3), the calculus of C_j can be done as:

$$\begin{aligned} C_0 &= X_0 \oplus X_1 \oplus X_3 \oplus X_4 \oplus X_6 \\ C_1 &= X_0 \oplus X_2 \oplus X_3 \oplus X_5 \oplus X_6 \\ C_2 &= X_1 \oplus X_2 \oplus X_3 \oplus X_7 \\ C_3 &= X_4 \oplus X_5 \oplus X_6 \oplus X_7 \\ C_4 &= X_0 \oplus X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus X_6 \oplus X_7 \end{aligned} \tag{3}$$

On the other hand, to calculate the syndrome bits, (4) are used:

$$\begin{aligned} S_0 &= C_0 \oplus X_0 \oplus X_1 \oplus X_3 \oplus X_4 \oplus X_6 \\ S_1 &= C_1 \oplus X_0 \oplus X_2 \oplus X_3 \oplus X_5 \oplus X_6 \\ S_2 &= C_2 \oplus X_1 \oplus X_2 \oplus X_3 \oplus X_7 \\ S_3 &= C_3 \oplus X_4 \oplus X_5 \oplus X_6 \oplus X_7 \\ S_4 &= C_4 \oplus X_0 \oplus X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus X_6 \oplus X_7 \end{aligned} \tag{4}$$

To form the matrix code, we have used the logical bit layout shown in Fig. 2 (extracted from [15]), where X_i are the data bits, C_j are the horizontal check bits (calculated using the SEC-DED Hamming code obtained from (3)), and P_k are the column parity bits (calculated using even parity). In this way, by combining Hamming codes and parity checks [15], this Matrix code form a two-dimensional scheme for correcting and detecting some patterns of MCUs.

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	C_0	C_1	C_2	C_3	C_4
X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_5	C_6	C_7	C_8	C_9
X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}
P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7					

Fig. 2. Layout of a 32 data-bit word for the Matrix code (extracted from [15])

The logical layout shown in Fig. 2 can be changed to adapt it to different data word sizes. In this work, we have used also 16 and 64 data bits, with the logical layouts shown in Fig. 3 and Fig. 4, respectively.

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	C_0	C_1	C_2	C_3	C_4
X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_5	C_6	C_7	C_8	C_9
P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7					

Fig. 2. Layout of a 16 data-bit word for the Matrix code

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	C_0	C_1	C_2	C_3	C_4
X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_5	C_6	C_7	C_8	C_9
X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}
X_{32}	X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}
X_{40}	X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	X_{46}	X_{47}	C_{25}	C_{26}	C_{27}	C_{28}	C_{29}
X_{48}	X_{49}	X_{50}	X_{51}	X_{52}	X_{53}	X_{54}	X_{55}	C_{30}	C_{31}	C_{32}	C_{33}	C_{34}
X_{56}	X_{57}	X_{58}	X_{59}	X_{60}	X_{61}	X_{62}	X_{63}	C_{35}	C_{36}	C_{37}	C_{38}	C_{39}
P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7					

Fig. 4. Layout of a 64 data-bit word for the Matrix code

The basic behavior of the Matrix code is as follows. The primary data input (X_i) is divided into groups of several bits. In this work, this division is in groups of 8 bits. Each group is codified using a (13, 8) SEC-DED Hamming code (C_j). Lastly, a set of vertical parity bits (P_k) completes the matrix. In this way, these Matrix codes can correct all single errors, as well as they can correct and detect all 2-bit random errors affecting data bits X_i. When a single error is produced in a row, it can be corrected by using the syndrome bits generated by comparing the code bits stored in memory with the code bits calculated with the data bits also stored in memory. In the case of 2-bit errors in the same row, the parity check bits are also used.

Nevertheless, these ECCs present two main problems. Firstly, the redundant bits (C_j and P_k) add a great overhead in the area, power, and delay of the memory system.

Secondly, these Matrix codes are not able to correct all double errors produced in a row. As it can be seen in (3), C_4 bit calculates the even parity of the data bits only (X_i bits). That is, the Matrix code presented in [15] uses a pseudo Extended Hamming code. It behaves like a SEC code for all the codeword bits (X_i-C_j), but the double error detection (DED behavior) is only accomplished in the data bits (the X_i bits).

In this way, combining the value of C_4 with the value of the rest of C_j bits and with the values of the parity bits (P_k), it is possible to detect and correct all double errors produced in the data bits, that is, only in the X_i bits.

So, what happens when a double error affects a data bit and a code bit? Let's see an example of this event. Let's suppose there is a double error that affects a data bit (X_i bit) and a code bit (C_j bit). In this case, C_4 bit detects only a single bit error (in the X_i bit, as C_j bits are not included in this calculus). When the system tries to correct the error, syndrome bits C_0-C_3 points to a bit different to the erroneous X_i bit. Thus, the correction is done in a non-erroneous bit, introducing a new error.

2.3 Hsiao codes

When defining the parity check matrix **H** of a Hsiao code, three constraints must be satisfied [20]:

1. There are no all-0 columns.
2. Every column is distinct.
3. Every column contains an odd number of 1's.

The first two constraints give a code with a Hamming distance of 3. The third constraint guarantees an ECC with a Hamming distance of 4. Thus, the code generated is a SEC-DED code for all the codeword.

Hsiao codes constructed in [20] always showed fewer 1's in its parity check matrix **H** than the Hamming SEC-DED codes. This translates into less hardware area in the corresponding ECC circuitry, implying a lower static power consumption. Furthermore, by selecting the odd weight columns in a way that balances the number of 1's in each row of the **H**-matrix, the dynamic power also reduces. This reduction is provoked because the number of transitions in encoder/decoder circuits is limited. In addition, the delay of the checker can be minimized, as the delay is constrained by the maximum weight row. If we assume the use of typical 2-input XOR gates, the delay is proportional to the maximum number of logic levels in the row equations. So:

$$delay \sim \log_2[maximum\ weigth\ row] \quad (5)$$

Particularly, the parity check matrix used in this paper is shown in (6), where X_i are the data bits and C_j are the horizontal check bits.

$$\mathbf{H} = \begin{array}{c} C_0C_1\cdots\cdots C_4\,X_0X_1\cdots\cdots\cdots\cdots X_7 \\ \left|\begin{array}{ccccccccccccc} 1&0&0&0&0&1&1&1&0&1&1&0&0 \\ 0&1&0&0&0&1&1&0&1&1&0&1&0 \\ 0&0&1&0&0&1&0&1&1&0&1&0&1 \\ 0&0&0&1&0&0&1&1&1&0&0&1&1 \\ 0&0&0&0&1&0&0&0&0&1&1&1&1 \end{array}\right| \end{array} \qquad (6)$$

As it can be seen, this parity matrix satisfies the three constraints commented before to define a Hsiao code. In fact, we have been able to design a Hsiao code with a maximum of only three 1's bits per column. As commented previously, it is very easy to obtain the encoding and decoding formulas from **H**, as shown in (7) and (8) respectively:

$$\begin{aligned} C_0 &= X_0 \oplus X_1 \oplus X_2 \oplus X_4 \oplus X_5 \\ C_1 &= X_0 \oplus X_1 \oplus X_3 \oplus X_4 \oplus X_6 \\ C_2 &= X_0 \oplus X_2 \oplus X_3 \oplus X_5 \oplus X_7 \\ C_3 &= X_1 \oplus X_2 \oplus X_3 \oplus X_6 \oplus X_7 \\ C_4 &= X_4 \oplus X_5 \oplus X_6 \oplus X_7 \end{aligned} \qquad (7)$$

$$\begin{aligned} S_0 &= C_0 \oplus X_0 \oplus X_1 \oplus X_2 \oplus X_4 \oplus X_5 \\ S_1 &= C_1 \oplus X_0 \oplus X_1 \oplus X_3 \oplus X_4 \oplus X_6 \\ S_2 &= C_2 \oplus X_0 \oplus X_2 \oplus X_3 \oplus X_5 \oplus X_7 \\ S_3 &= C_3 \oplus X_1 \oplus X_2 \oplus X_3 \oplus X_6 \oplus X_7 \\ S_4 &= C_4 \oplus X_4 \oplus X_5 \oplus X_6 \oplus X_7 \end{aligned} \qquad (8)$$

Matrix **H** of Hsiao code (6) has some advantages over matrix **H** of Extended Hamming code (2):

- A lower total number of 1's.
- The number of 1's in the rows is more balanced, with a lower maximum weight row.

Thus, it is expected an improvement of the area, power, and delay overhead. In this way, we have built a Matrix code with the same layout shown in Fig. 2, Fig. 3 and Fig. 4 (for 16, 32 and 64 data bits respectively). In this new Matrix code, we have used the Hsiao code shown in (7) instead the pseudo Extended Hamming SEC-DED code shown in (3).

Notice that the redundancy of Matrix-Hsiao code is the same as the Matrix-Extended Hamming code, because the number of check bits C_j and parity bits P_k is the same.

3 Error Correction Codes Evaluation

As commented before, in this paper we propose the use of Hsiao codes instead of Extended Hamming codes inside a Matrix code. We have used the same logical bit layout for both types of Matrix codes.

Thus, in this section, we compare the error coverage of the different Matrix codes commented before, as well as the overheads introduced by these codes.

This evaluation has been carried out with two different processes. During the first one, we have injected faults in C models of the ECCs for error coverage evaluation. Then, in a second step, we have implemented the different ECCs in VHDL, and we have synthesized them, to estimate area, power, and delay overheads. This section finishes with a global comparison of the ECCs.

3.1 Error Coverage Evaluation

In coding theory, the term *random error* commonly refers to one or more bits in error, distributed randomly in the encoded word (data bits plus code bits generated by the ECC). Random errors can be *single* (only one bit affected) or *multiple*. *Single errors* only affect a single memory cell. They are commonly produced by single event upsets (SEU, in random access memories) or single event transients (SET, in combinational logic) [2][3][4][5].

As commented in the Introduction section, multiple errors are becoming more frequent due to the continuous increasing of the integration scale [2][7][8][25]. The main physical causes of multiple errors in the context of RAM memories are diverse: high energy cosmic particles that hit some neighbor cells, crosstalk between neighbor cells, etc. [26][27].

In this way, to study the error coverage of the different ECCs, we have developed a simulator that allows injecting different types of error. Particularly, we have injected *single errors*, as well as random *multiple errors*.

The basic scheme of our fault injection tool is shown in Fig. 5. By comparing the input and output words, the simulator can check if the error injected leads to a right or wrong decoding. Also, the decoder circuit can activate the NRE (*Non Recoverable Error*) signal when an error is detected but it cannot be corrected.

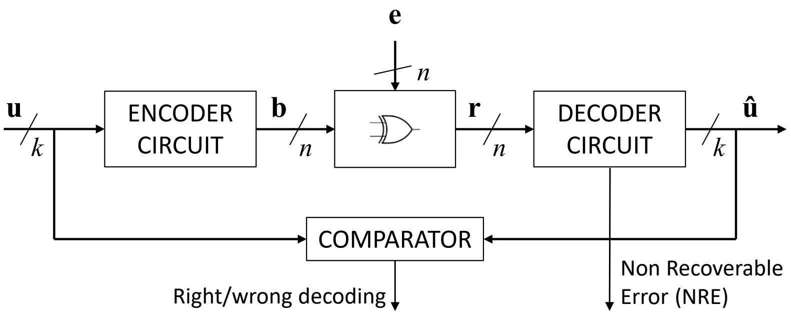

Fig. 5. Block diagram of the fault injector simulator

Repeating the process for all errors of a given size and model, we have been able to count the number of corrected and/or detected errors with respect to the total number of injected errors. That is, we have been able to calculate the coverage of each ECC.

We must remark that we have not injected errors according to their probability of occurrence, as our goal is to measure the correction and detection coverages, that represent percentages. Specifically, we have injected each type of error (*single* errors or *multiple* errors of different lengths) in all bits of the codeword (data and check bits) to verify the error correction/detection capabilities of the different ECCs. Obviously, random errors of length 8 will be much less frequent than random errors of length 2, as bibliography shows [4][25].

All blocks of the fault injection tool have been developed in C, using bitwise logic operators for an accurate simulation of the hardware behavior. These circuits are implemented in C as encoding and decoding functions. Adapting the simulator for a different ECC is as simple as adjusting the word lengths and replacing the encoding and decoding functions for the new ECC, extracted from the parity matrix **H**.

Correction coverage has been calculated as:

$$C_{correc} = \frac{Errors\_Corrected}{Errors\_Injected} \times 100 \tag{9}$$

where *Errors_Corrected* are the number of errors corrected by the ECC, and *Errors_Injected* are the number of errors injected. Although errors are injected in both data and check bits, notice that error corrected means that data bits have no errors.

On the other hand, detection coverage is calculated as:

$$C_{detec} = \frac{Errors\_Corrected + Errors\_Detected}{Errors\_Injected} \times 100 \tag{10}$$

where *Errors_Detected* corresponds to the number of errors detected but not corrected by the ECCs.

Fig. 6, Fig. 7, and Fig. 8 show the error coverages for Matrix-Extended Hamming and Matrix-Hsiao codes. Errors have been injected in the layouts of Fig. 2, Fig. 3, and Fig. 4, corresponding to 16, 32 and 64 data bits. Single and multiple random errors with length from 2 to 8 have been injected. This range is representative of typical values of MCUs in terrestrial and critical environments, such as space environment [4][25].

As it can be seen in Fig. 6, Fig. 7 and Fig. 8, all ECC show the same trend, and almost the same data. On the one hand, when using Matrix-Hsiao codes it is possible to correct all single errors as well as all 2-bit random errors. In the case of the Matrix-Extended Hamming codes, and as explained before, there exist some patterns of 2-bit errors that cannot be corrected. For 3-bit and longer, we can see that the correction coverage decreases, and both ECCs present very similar error correction coverages. On the other hand, all ECCs can detect 100% of single and 2-bit random errors, and more than 97% of longer errors.

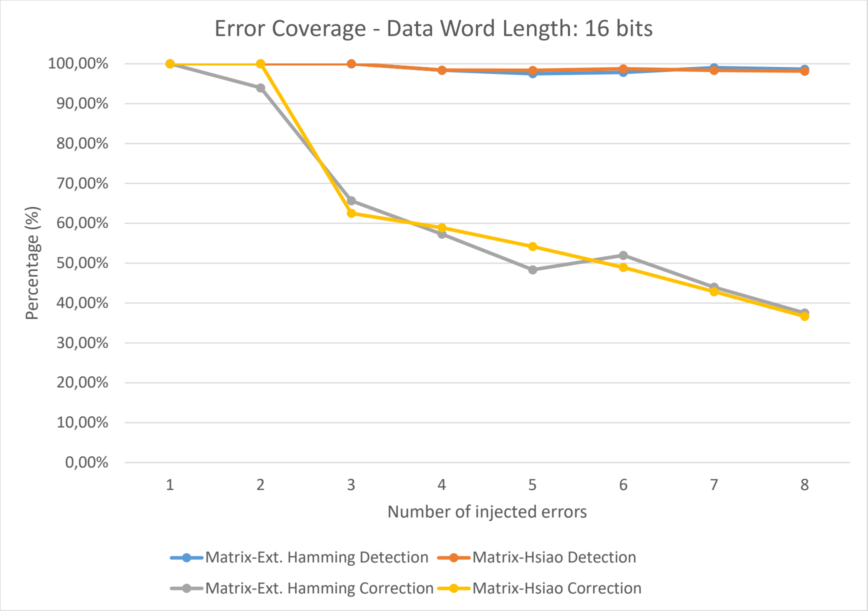

Fig. 6. Correction and detection coverages. Data word length = 16 bits

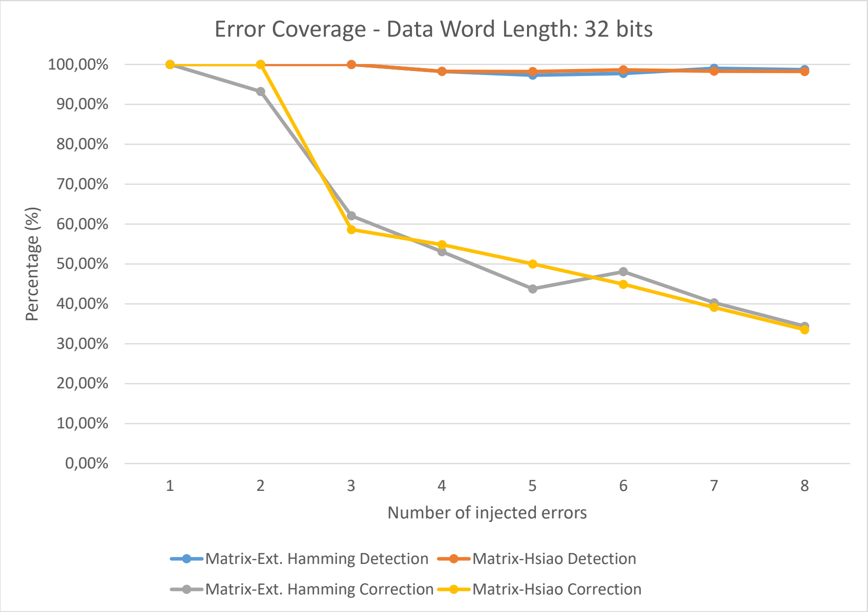

Fig. 7. Correction and detection coverages. Data word length = 32 bits

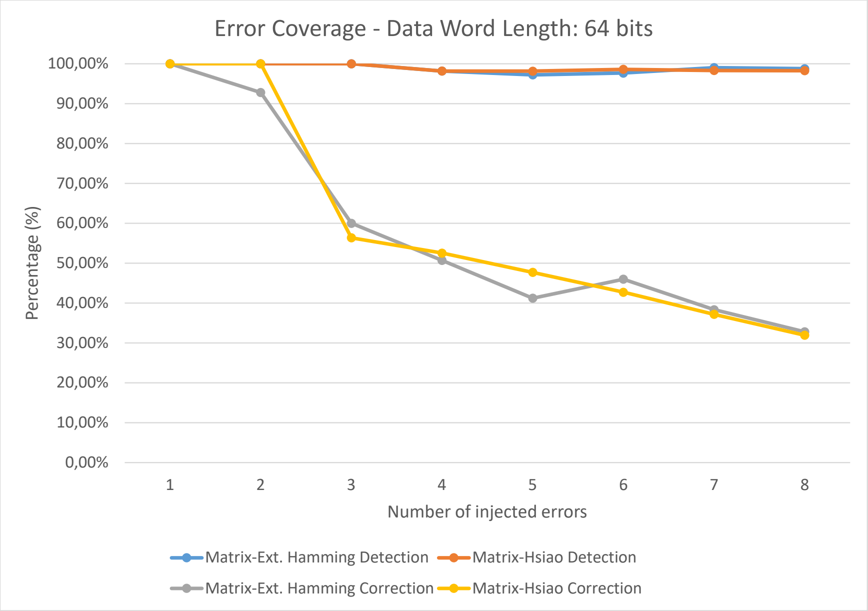

Fig. 8. Correction and detection coverages. Data word length = 64 bits

3.2 Synthesis results

We have synthesized all ECCs for 16, 32, and 64 data word length, using Extended Hamming and Hsiao codes. In total, six different Matrix codes. To do this, we have implemented them in VHDL, and using CADENCE software [30], we have carried out a logic synthesis for 45 nm technology by using the NanGate FreePDK45 Open Cell Library [31][32].

In this way, Fig. 9 shows the area occupied by the different circuits (in nm^2, 1 nm=10^{-9} m). Matrix-Hsiao codes present an improvement of the area overhead for the three data word lengths. As commented in Section 2.C, the design of the parity check matrix for Hsiao codes minimizes the number of 1's, which provokes less hardware area. This reduction in hardware gives an additional benefit, as it tends to lower the chance of failure, increasing the reliability of the ECCs. Notice that the decoder's area is bigger than the encoder's area. The decoder must calculate the syndrome, locate the error, and correct it. Besides, longer data words provoke a higher area because the size of encoders and decoders increases.

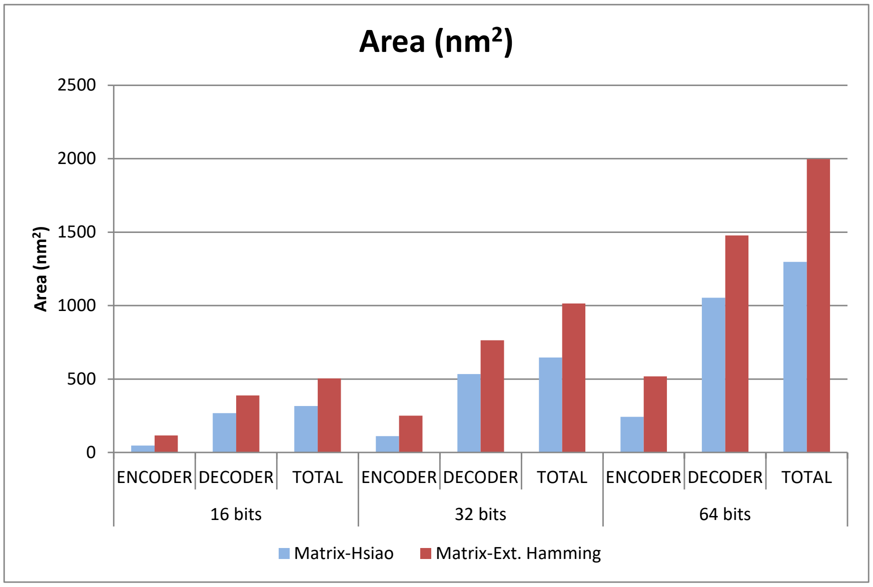

Fig. 9. Area overhead (in nm^2)

On the other hand, Fig. 10 shows the power consumption (static (leakage) plus dynamic) of the different encoders and decoders (in µW, 1 µW=10^{-6} W). As it can be observed, power consumption follows the same trend than area overhead. That is, power overhead is lower when using Matrix-Hsiao codes (a half, more or less) for all data word lengths. As explained in Section 2.C, the reduction of the power consumption is an expected result due to the lower area and the limitation of the logic transitions.

Besides, longer data words provoke higher power consumption (because the size of encoders and decoders increases). In any case, for the same data word length, Matrix-Hsiao codes power overhead is lower than the Matrix-Extended Hamming one.

Fig. 11 and Fig. 12 show static and dynamic power separately. As expected, the trend is maintained. That is, Matrix-Hsiao codes show better results in both cases.

Lastly, Fig. 13 shows the delay introduced by the different Matrix codes (in ps, 1 ps=10^{-12} s). As observed, encoders for Matrix-Extended Hamming codes present a greater delay than the same circuits for Matrix-Hsiao codes. This is caused by the balanced number of 1's in each row of the Hsiao parity check matrix, which provokes a reduction on the number of logic levels. Faster encoding and decoding imply faster memory writing and reading cycles.

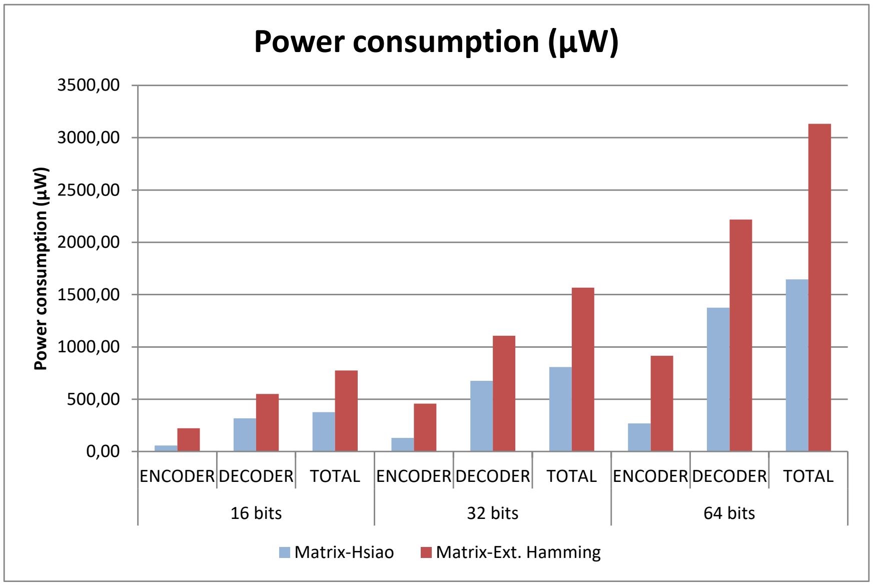

Fig. 10. Power consumption (in µW)

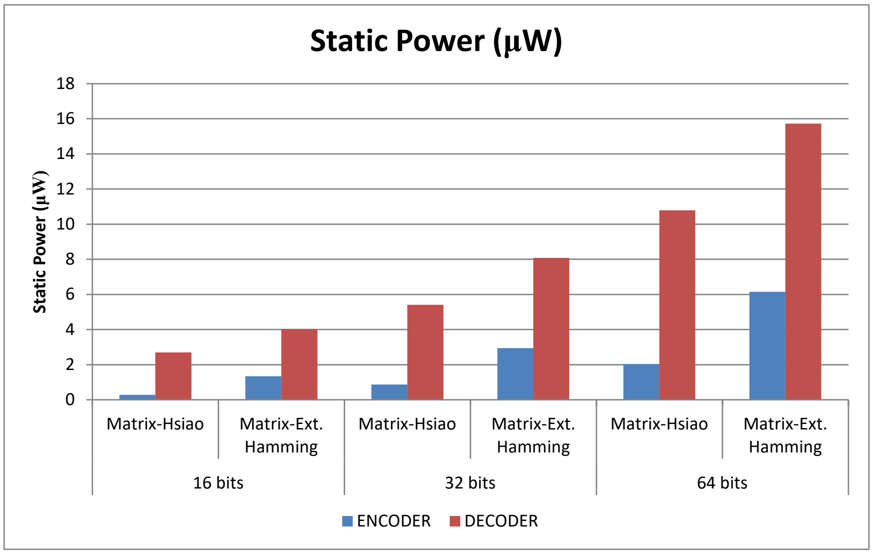

Fig. 11. Static power consumption (in µW)

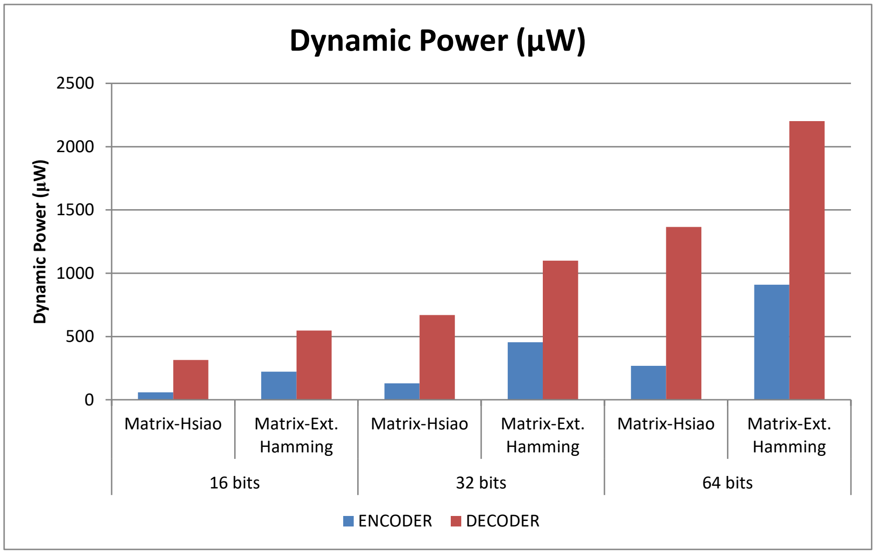

Fig. 12. Dynamic power consumption (in µW)

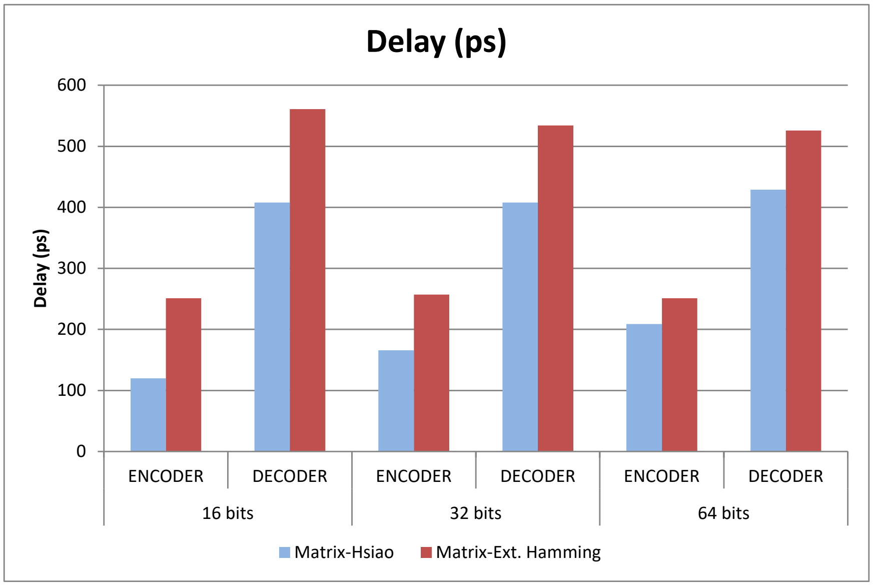

Fig. 13. Delay overhead (in ps)

3.3 Global evaluation of the ECCs. M metric

Results obtained in previous sections show the advantages in using Hsiao codes to improve Matrix codes. In order to provide a global evaluation of the different codes, several metrics have been proposed in the literature [15][16][23][33].

Introduced in [23], **M** metric is a complete and accurate metric that can be used to tradeoff area, power, delay, redundancy, and error coverage. This allows comparing different codes in a global sense. Also, this metric can be used to enhance a parameter in a specific application by weighting it appropriately. **M** metric is defined as [23]:

$$M = \frac{C_{correc} \times C_{detec}}{Area \times Power \times Delay \times Redundancy} \tag{11}$$

Fig. 14 shows the results of the calculus of the **M** metric for all Matrix codes. In this way, and according to **M** metric, it is worthwhile the use of the Hsiao code to form a Matrix code. As we can see, and for a specific codeword length, **M** metric for the respective Matrix-Hsiao code is better than the Matrix-Extended Hamming one. For 3-bit errors or longer, **M** metric decreases due to the reduction of error coverages.

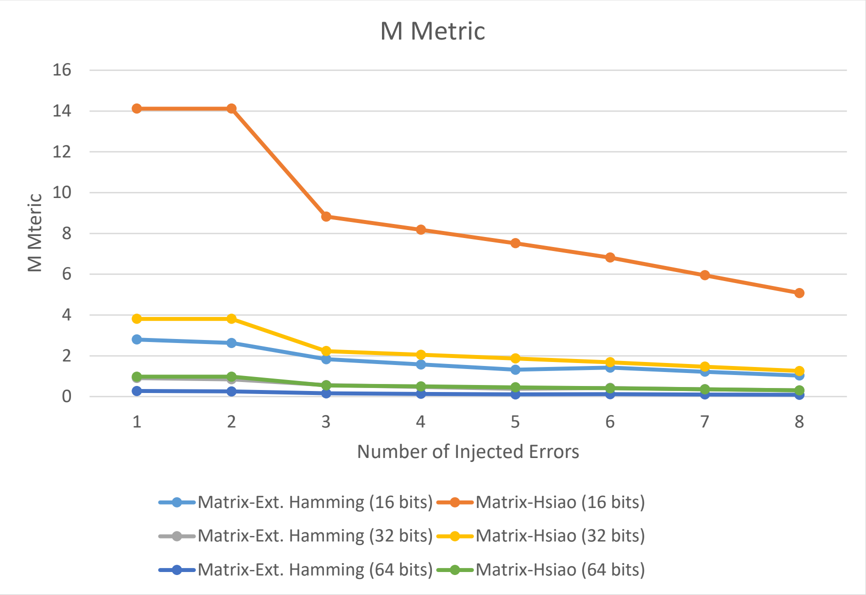

Fig. 14. M metric for both ECC Matrix codes

5 Conclusions

In this work, we have introduced a series of Matrix codes that use Hsiao codes instead of Extended Hamming codes. We have observed savings in the area, power, and delay overhead of the ECC circuits, with the same redundancy.

Concerning error detection and correction, we have injected random single and multiple errors in the codeword layouts. We have checked that Matrix-Hsiao codes are able to correct all single and 2-bit errors, while Matrix-Extended Hamming codes can only correct 100% of single errors.

To evaluate the global goodness of the codes, we have used the **M** metric, a figure of merit that combines area, power, delay, error coverage, and redundancy factors. As can be seen, Matrix-Hsiao codes present better values of **M** in all the cases.

In the future, we want to continue developing ECCs to decrease area, power, and delay overheads, while maintaining, or even increasing, the error coverage.

References

1. V. Sridharan, et al., "Memory Errors in Modern Systems: The Good, The Bad, and The Ugly", 20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2015), pp. 297–310, March 2015.
2. B. Sangchoolie, K. Pattabiraman, and J. Karlsson. An empirical study of the impact of single and multiple bit-flip errors in programs. IEEE Trans. on Dependable and Secure Computing, 19(3):1988–2006, 2022.
3. G. Tsiligiannis et al., "Multiple cell upset classification in commercial SRAMs," IEEE Transactions on Nuclear Science, vol. 61, no. 4, pp. 1747–1754, 2014.
4. N.G. Chechenin and M. Sajid, "Multiple cell upsets rate estimation for 65 nm SRAM bitcell in space radiation environment", 3rd International Conference and Exhibition on Satellite & Space Missions, May 2017.
5. Y. Bentoutou, "Program memories error detection and correction on-board earth observation satellites", International Journal of electrical and Computer Engineering, vol. 4, nº 6, pp. 933-936, 2010.
6. M. Murat, A. Akkerman, and J. Barak, "Electron and ion tracks in silicon: Spatial and temporal evolution," IEEE Transactions on Nuclear Science, vol. 55, no. 6, pp. 3046–3054, December 2008.
7. G. Tsiligiannis et. al., "Multiple Cell Upset Classification in Commercial SRAMs", IEEE Transactions on Nuclear Science, vol. 61, no. 4, August 2014.
8. M. Wirthlin, D. Lee, G. Swift, and H. Quinn, "A method and case study on identifying physically adjacent multiple-cell upsets using 28-nm, interleaved and SECDED-protected arrays," IEEE Transactions on Nuclear Science, vol. 61, no. 6, pp. 3080–3087, Dec. 2014.
9. E. Fujiwara, Code Design for Dependable Systems: Theory and Practical Application, Ed. Wiley-Interscience, 2006.
10. R. W. Hamming, "Error detecting and error correcting codes," Bell System Technical Journal, vol. 29, pp. 147–160, 1950.
11. C.L. Chen and M.Y. Hsiao, "Error-correcting codes for semiconductor memory applications: a state-of-the-art review", IBM Journal of Research and Development, vol. 58, no. 2, pp. 124–134, March 1984.

12. L. J. Saiz-Adalid, J. Gracia-Morán, D. Gil-Tomás, J.-C. Baraza-Calvo and P.-J. Gil-Vicente, "Ultrafast Codes for Multiple Adjacent Error Correction and Double Error Detection," in IEEE Access, vol. 7, pp. 151131-151143, 2019.
13. A. Sánchez-Macián, P. Reviriego, J. Tabero, A. Regadío, and J.A. Maestro, "SEFI protection for Nanosat 16-bit Chip On-Board Computer Memories", IEEE Transactions on Device and Materials Reliability, DOI 10.1109/TDMR.2017.2750718, 2017.
14. J. Gracia-Moran, L. -J. Saiz-Adalid, J. -C. Baraza-Calvo and P. Gil, "Correction of Adjacent Errors with Low Redundant Matrix Error Correction Codes," 2018 Eighth Latin-American Symposium on Dependable Computing (LADC), Foz do Iguacu, Brazil, 2018, pp. 107-114, doi: 10.1109/LADC.2018.00021
15. C. Argyrides, D.K. Pradhan, and T. Kocak, "Matrix codes for reliable and cost efficient memory chips", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 19, nº 3, pp.420–428, March 2011.
16. H.S. de Castro, et al. "A correction code for multiple cells upsets in memory devices for space applications", 2016 14th IEEE International New Circuits and Systems Conference (NEWCAS 2016), pp.1–4, June 2016.
17. S. Liu, L. Xiao, J. Li, Y. Zhou, and Z. Mao, "Low Redundancy Matrix-Based codes for Adjacent Error Correction with Parity Sharing", 2017 18th International Symposium on Quality Electronic Design (ISQED 2017, March 2017.
18. P. Reviriego and J.A. Maestro, "Efficient Error Detection Codes for Multiple-Bit Upset Correction in SRAMs with BICS", ACM Transactions on Design Automation of Electronic Systems (TODAES) Vol. 14 nº 1, January 2009.
19. M. Zhu, L. Xiao, S. Li, and Y. Zhang, "Efficient Two-Dimensional Error Codes for Multiple Bit Upsets Mitigation in Memory", 2010 25th International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2010), pp. 129-135, October 2010.
20. M.Y. Hsiao, "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes", IBM Journal of Research and Development, vol. 14, no. 4, pp. 395-401, 1970.
21. Saiz-Adalid, L.-J.; Gracia-Morán, J.; Gil-Tomás, D.; Baraza-Calvo, J.-C.; Gil-Vicente, P.-J. Reducing the Overhead of BCH Codes: New Double Error Correction Codes. Electronics 2020, 9, 1897. https://doi.org/10.3390/electronics9111897
22. F. Aymen, H. Belgacem, and K. Chiraz, "A new efficient self-checking Hsiao SEC-DED memory error correcting code", 2011 International Conference on Microelectronics (ICM 2011), pp. 1-5, December 2011.
23. J. Gracia-Moran, L.J. Saiz-Adalid, D. Gil-Tomás, and P.J. Gil-Vicente, "Improving Error Correction Codes for Multiple Cell Upsets in Space Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, DOI: 10.1109/TVLSI.2018.2837220, 2018.
24. A. Neubauer, J. Freudenberger, and V. Kühn, Coding Theory: Algorithms, Architectures and Applications. John Wiley & Sons, 2007.
25. G.I. Zebrev, "Multiple Cell Upset Cross-Section Uncertainty in Nanoscale Memories: Microdosimetric Approach", 15th European Conference on Radiation and its Effects on Components and Systems (RADECS 2015), September 2015.
26. M. Greenberg, "Reliability, availability, and serviceability (ras) for ddr dram interfaces," in memcon.com. Available at: http://www.memcon.com/pdfs/proceedings2014/NET105.pdf, 2014.
27. IEEE International Roadmap for Devices and Systems 2030. [Online]. Available at: https://irds.ieee.org/editions/2023
28. J. Gracia-Moran, D. Gil-Tomas, L.J. Saiz-Adalid, J.C. Baraza-Calvo, P.J. Gil-Vicente, "Experimental Validation of a Fault Tolerant Microcomputer System against Intermittent Faults", 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010), pp. 413-418, June 2010.
29. D. Gil-Tomás, J. Gracia-Morán, J.C.Baraza-Calvo, L.J. Saiz-Adalid, P.J. Gil-Vicente, "Injecting Intermittent Faults for the Dependability Assessment of a Fault-Tolerant Micro-

computer System", IEEE Transactions on Reliability, Vol. 65, nº 2, pp. 648-661, June 2016.

30. https://www.cadence.com/
31. J.E Stine et al., "FreePDK: An Open-Source Variation-Aware Design Kit", IEEE International Conference on Microelectronic Systems Education (MSE'07), June 2007.
32. http://www.nangate.com/?page_id=2325
33. C. Argyrides, H.R. Zarandi and D.K. Pradhan, "Matrix Codes: Multiple Bit Upsets Tolerant Method for SRAM Memories", 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2007.

Analysis of the impact of faults in a Convolutional Neural Network implemented in a Raspberry Pi

J. Gracia-Morán[1], J. Bazán-Andría[2], Juan Carlos Ruiz[1],
David de Andrés[1], L.J. Saiz-Adalid[1]

[1] Instituto ITACA, Universitat Politècncia de València
Camino de Vera, s/n, 46022 Valencia, Spain
{jgracia, ljsaiz, jcruizg, ddandres}@itaca.upv.es
https://gstf.blogs.upv.es/

[2] Escuela Técnica Superior de Ingeniería Industrial
Universitat Politècncia de València
Camino de Vera, s/n, 46022 Valencia, Spain
jubaan @ etsii.upv.es

Abstract. During these last years, the use of embedded systems has grown exponentially. New applications are continuously implemented in these systems, such as neural networks. This type of machine learning software is often highly accurate and efficient, and it has been used in a wide variety of applications. Also, it makes an extensive use of memory. A problem that carries out the expansive use of embedded systems is its reliability. Embedded systems are built with low-reliable components, reduced weight and volume, and not very high computing and memory capacity for low power consumption. Usually, memory system is the component most affected by faults in a computer system.
With these conditions, some questions arise. How can we rely on the results obtained by a neural network implemented in an embedded system? How will affect to the behavior of the neural network a faulty bit?

1 Introduction

Nowadays, neural networks are being widely used in a huge number of applications, and their implementation ranges from high-performance computing systems [1] to embedded systems [2].

Embedded systems are being used massively, for instance, on the Internet of Things (IoT) or in safety-critical applications. This massive use has provoked a huge amount of generated data [3], that must be processed.

Neural networks are one of these data-processing applications moving to embedded systems [2]. In this way, an aspect that is gaining importance is the effects of faults in neural networks, especially when they execute safety-critical applications [4][5][6].

Currently, CMOS technology integration scale has allowed the design of memory systems with a great storage capacity. However, this aggressive scaling has also caused an increment in the memory fault rate [7]. As the memory cell critical charge

C.Fernandez-Llatas and M.Guillen (Ed.):
Workshop on Innovation on
Information and Communication Technologies
(ITACA-WIICT 2024)

ISBN 978-84-09-66630-0

and the energy needed to cause a Single Cell Upset (SCU) is also reduced, Multiple Cell Upsets (MCUs), that is, simultaneous errors in more than one memory cell, can also be induced by a single particle hit [8].

If an embedded system is going to execute memory-intensive programs, such as neural networks do, we must rely on this processing. Thus, it is interesting to know if memory errors would affect the neural network's prediction. Embedded systems have been built with low-reliable components, reduced weight and volume, and not very high computing and memory capacity for low power consumption. In this way, if fault tolerance is needed, it must be implemented by software, as the low-cost hardware used doesn't have this protection.

In this work, we have implemented a Convolutional Neural Network (CNN) [9]10] in a Raspberry Pi, and then, we have carried out a series of fault injection campaigns in the tensors of the CNN, as they are an essential part of the CNN. Tensors are massively used, and they are stored in SRAM. Thus, we have studied the effects of simple and multiple adjacent faults in the tensors of the CNN by injecting different fault models, such as bit-flip and stuck-at ('0', '1').

This work is organized as follows. Section 2 summarizes previous works analyzing sis the reliability of neural networks. Section 3 describes the system used, while Section 4 explains the experiments carried out. Finally, Section 5 concludes this paper.

2 Related Works

Different works have proposed several approaches to increment the reliability of neural networks. For instance, in [11], authors evaluate the reliability of different GPUs executing various neural networks. One of these GPUs implements a SEC-DED (Single Error Correction-Double Error Detection) ECC and a parity check. Authors checked that error detection property provokes a high number of application crashes in the GPU (they used three different NVIDIA GPUs: i) TeslaK40; ii) TegraX1; and iii) TitanX). This result has been confirmed in [12], where authors assure that it is better to leave memory unprotected than use error detection codes.

The use of an ECC together with a new training scheme of the neural network is proposed in [13], trying to increment the neural network reliability. Specifically, the ECC used is a SEC-DED, and authors employed devices equipped with a 40-core 2.2GHz Intel Xeon Silver 4114 processor, 128GB of RAM, and an NVIDIA TITAN Xp GPU with 12GB memory.

On the other hand, authors in [14] used different ECCs to selectively protect certain bits in the weights of the neural network. The idea is to achieve a trade-off between redundancy and neural network performance. This idea is also used in [15], where some non-critical bits are replaced with Hamming ECCs. In any case, all these works do not protect all data bits.

A different type of ECC is presented in [19], where a new error correction scheme for analog neural network accelerators based on arithmetic codes is presented. They simulate the accelerator used.

Fault injection has been massively used to study neural network reliability. For example, in [16], the impact of faults in different data types is studied. They inject bit-flips in the weights of the neural networ. Single-event upsets are injected in [4], in data paths and buffers. They classify error propagation with respect to neural network structure, data types, and layers properties.

The effects of permanent faults have been studied in [17]. Specifically, they inject stuck-at (‘0’, ‘1’) faults. On the other hand, in [18], statistical fault injection is used to reduce the number of fault injections experiments.

Summarizing, the works presented in this section analyses the effects of faults in neural networks, but they only studied transient faults (such as bit-flips), or permanent faults (such as stuck-at (‘0’, ‘1’)) in systems with a great capability of memory and calculus. In the work we present here, we have studied the effects of transient and permanent faults in an embedded system, that is characterized by a no so powerful processor and a limited memory capacity.

3 System description

3.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a subset of Deep Neural Networks [9]. Their name comes from the mathematical linear operation called convolution, usually used between matrixes. CNNs include multiple layers, such as convolutional, non-linearity, pooling, and fully-connected. CNNs have a very good performance in applications that deal with image classification, computer vision, and natural language processing.

In the present work, we have implemented a simplified version of LeNet CNN [20]. Our CNN uses the MINST database for training and test [21]. Specifically, our CNN presents the diagram shown in Fig. 1. It has almost 45000 weights, stored in memory in 32-bit floating point format.

3.2 Hardware used

We have implemented the CNN explained before in a Raspberry Pi Model B+ V1.2. This model integrates the BCM2835 SoC with CPU, GPU, DSP and SDRAM. It uses a 700MHz ARM 1176JZF-S processor, and it works with 32-bit RISC. The Broadcom VideoCore IV model GPU features OPEN GL ES 2.0 MPEG-2 and VC-1 at 1080p resolution. Its RAM memory is 512MB shared with the GPU.

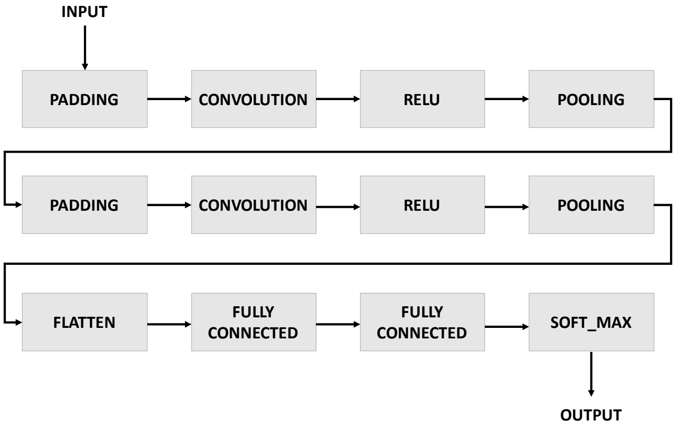

Fig. 1. Convolutional Neural Network organization

4 Study of the effects of transient and permanent faults in a Convolutional Neural Network

As commented in Section 2, the impact of faults in the weights of neural networks is being extensively studied, by injecting permanent and/or transient faults [6][17][26]. When studying specifically transient faults, usually bit-flips are injected. For instance, in [5], extensive fault injection campaigns in the weights of neural networks were made to study the capability of different memory encryption schemes to detect faults.

On the other hand, under the denomination of Bit-Flip Attack (BFA), different works study the effects of bit-flip errors in the weights of a neural network from reliability and security points of view. This is the case of [22]. This work arrives at the same conclusion: the closer the faulty bit is to the Most Significant Bit (MSB), the higher the probability of damage provoked.

The main idea of this work is to study the effects of transient and permanent faults in the behavior of a CNN implemented in a Raspberry Pi. To do this, we have carried out a series of fault injection experiments to check the validity of our approach. Fault injection is a very well-known technique used to assess the reliability of computer systems [23][24][25]. It allows a precise introduction of faults in the system.

In this work, we have injected single and multiple adjacent bit-flips and stuck-at (‘0’, ‘1’) in the weights of the two convolution layers of the CNN (see Fig. 1). Multiple faults range from 2 to 4. Fig. 2 show only the percentage of no correct predictions. As it can be seen, the biggest part of injected errors is tolerated (they don’t cause any misprediction). This is provoked by the intrinsic redundancy of the neural network.

However, there exist a non-negligible percentage of mispredictions even with a unique bit in error in a single layer. Thus, a bigger number of bits in error provokes a bigger percentage of mispredictions. We must remark that a fault (single or multiple) is injected in a unique weight. That is, an incorrect bit of a unique weight out of the set of weights of a specific layer can cause a misprediction.

If we study in more detail Fig. 2, we can see that for both layers, stuck-at ‘0’ errors are not very harmful (the percentage of mispredictions are almost zero). The reason is that the biggest part of weight bits is 0, provoking that the errors do not activate.

On the other hand, for Convolution layer 1, we can observe that bit-flips and stuck-at ‘1’ errors present similar percentages of mispredictions, while for Convolution layer 2, bit-flips errors provoke a bigger percentage of mispredictions. Another interesting fact we can observe is that the first layer is more vulnerable to errors than the second layer.

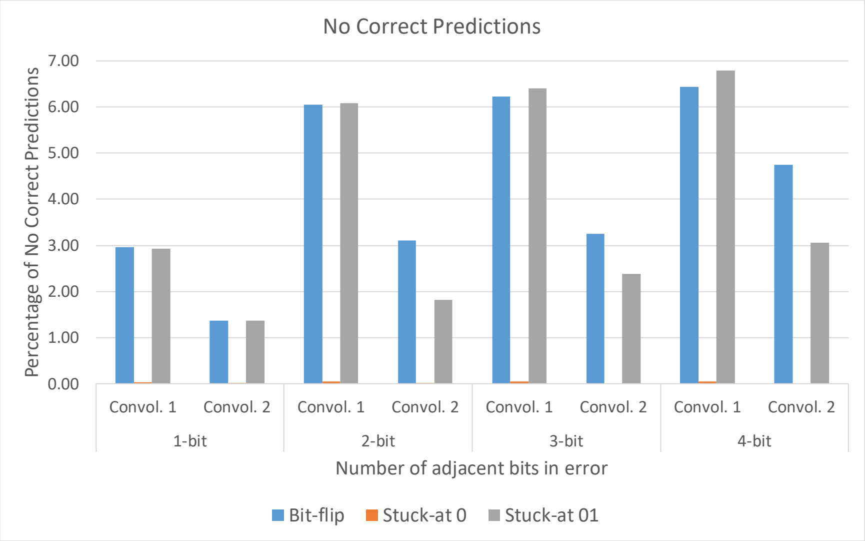

Fig. 2. Percentage of No Correct Predictions for both convolutional layers

Once checked that an error in one weight can provoke a misprediction, next step is to study whether all bits of the weights are equally harmful. That is, we want to study if a fault in a unique bit is equally risky that a fault in a different bit. Fig. 3 shows the percentage of mispredictions according to bit position for single and multiple adjacent bit-flips in both Convolutional layers. As we can see, all mispredictions are provoked when the Most Significant Bits (MSBs) are modified. Specially, when bit 30 is perturbed. Weights are stored in IEEE754 simple precision format (see Fig. 4). MSBs include the sign bit, the 8 bits of the exponent, and the MSBs of the magnitude (excluding the implicit bit). Particularly, a change in the bit 30 provokes a change in the MSB of the exponent, causing a really big change in the weight. Thus, a change that involves bit 30 has a great probability of causing a misprediction.

On the other hand, from bit 18th to 0th, no miscorrections are shown. That is, even modifying 4 adjacent bits at the same time, the CNN predictions are correct.

This trend is the same for both convolutional layers. Modifications of MSBs provoke the mispredictions. Almost all problems are provoked when bit 30 is perturbed. There exist also significative mispredictions up to bit 23.

Fig 3. also shows a substantial percentage of mispredictions when single faults affect bit 25 of the second convolutional layer. Weights values are between -1 and 1, so bit 25 is normally set to 0. A change in this bit causes a large change in the weight value, provoking the misprediction.

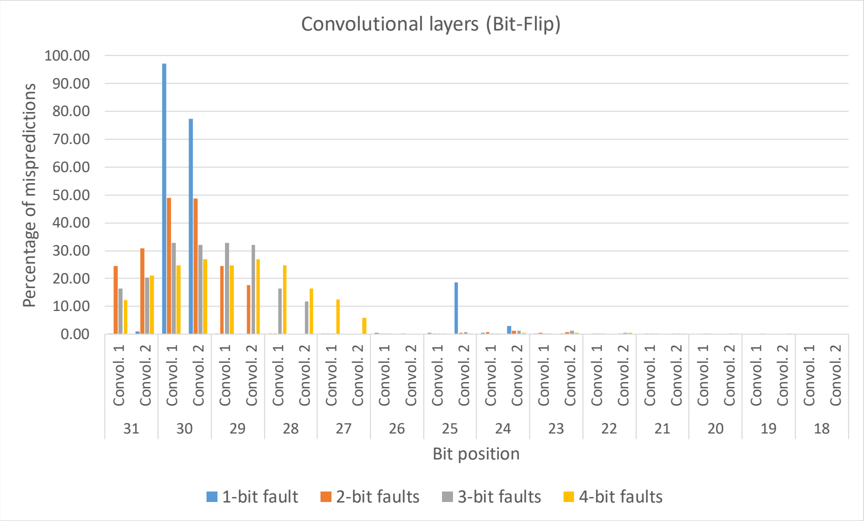

Fig. 3. Percentage of mispredictions according to bit position (bit-flip) for both convolutional layers

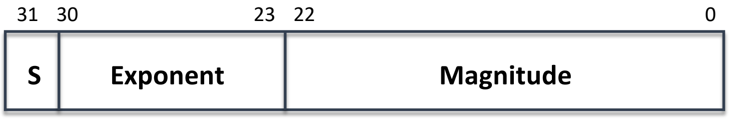

Fig. 4. IEEE754 simple precision format

Fig. 5 shows the percentage of mispredictions according to bit position when injecting stuck-at '1' faults. As we can see, the trend is the same that the shown before for bit-flips. That is, perturbations of bit 30 causes the biggest part of mispredictions, Convolution layer 2 has more problematic bits, and changes in bit 25 of Convolution layer 2 causes a significative percentage of mispredictions.

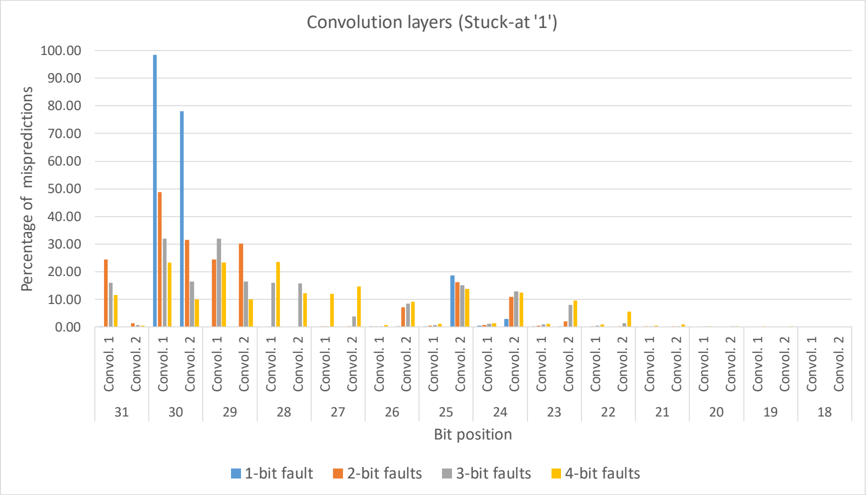

Fig. 5. Percentage of mispredictions according to bit position (stuck-at '1') for both convolutional layers

Finally, Fig 6 shows the percentage of mispredictions when stuck-at '0' faults are injected. A fact to consider is that percentages of mispredictions are really low (as shown in Fig. 2). In any case, we can see some interesting data. In Convolution layer 1, faults affecting MSB are equally important, as the percentages of mispredictions are equally distributed. Mispredictions are provoked by faults injected in the exponent bits (see Fig. 4). On the other hand, the higher intrinsic redundancy of Convolution layer 2 provokes mispredictions only when bits 31 and 30 are perturbed.

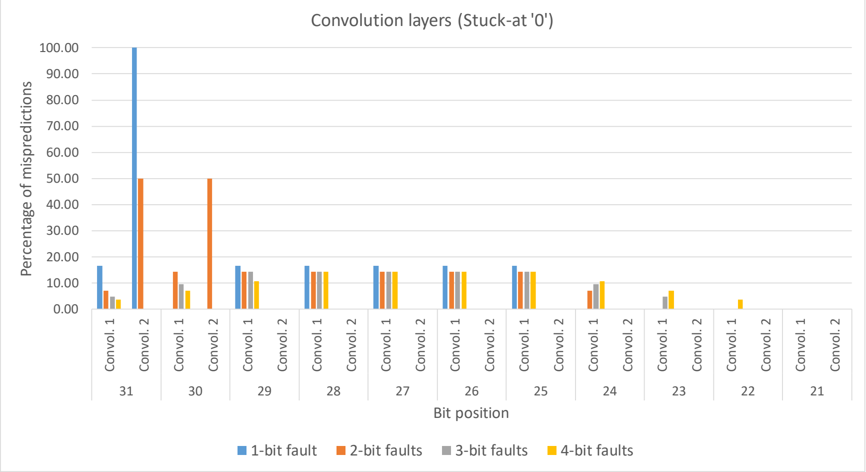

Fig. 6. Percentage of mispredictions according to bit position (stuck-at '0') for both convolutional layers

5 Conclusions

In this work, we have studied the reliability of a Convolutional Neural Network (CNN) implemented in a Raspberry Pi. Particularly, we have used a simplified version of LeNet. To carry out this study, we have executed an exhaustive fault injection campaign. We have injected single and multiple adjacent bit-flip and stuck-at ('0', '1') faults in all the bits of all the weights of both convolutional layers of the CNN.

We have studied if a unique erroneous weight can provoke a misprediction, that is, an incorrect behaviour of the complete CNN.

We have seen that there exist a non-negligible percentage of mispredictions caused even by a unique bit in error in both convolutional layers. These mispredictions are mainly caused by bit-flip and stuck-at '1' faults. Stuck-at '0' faults provoke a marginal percentage of mispredictions. Also, Convolutional layer 2 is more reliable than Convolutional layer 1. This is caused by the intrinsic redundancy of this second layer.

As expected, a bigger number of erroneous bits provoke a bigger percentage of mispredictions.

We have also studied which bits are more susceptible to cause an incorrect behaviour of the CNN. As weights are stored in IEEE754 simple precision format, MSBs are more sensitive to provoke mispredictions. Particularly, we have seen that faults in bit 30 are especially prone to provoke mispredictions.

In future works, we will continue studying the reliability of other types of network layers, as well as we want to define and implement fault tolerant mechanisms able to augment neural network's reliability without adding great overheads.

References

1. Dargan, S., Kumar, M., Ayyagari, M.R. et al. A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. Arch Computat Methods Eng 27, 1071–1092 (2020). https://doi.org/10.1007/s11831-019-09344-w
2. F. Wang, M. Zhang, X. Wang, X. Ma and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey", in IEEE Access, vol. 8, pp. 58322-58336, 2020, doi: 10.1109/ACCESS.2020.2982411.
3. "Cisco Global Cloud Index: Forecast and Methodology, 2016–2021", White Paper, 2018.
4. G. Li et al., "Understanding error propagation in deep learning neural network (DNN) accelerators and applications", International Conference for High Performance Computing, Networking, Storage and Analysis (SC '17), Article 8, 1–12, 2017 https://doi.org/10.1145/3126908.3126964.
5. N.I. Deligiannis, R. Cantoro, M. S. Reorda, M. Traiola and E. Valea, "Towards the Integration of Reliability and Security Mechanisms to Enhance the Fault Resilience of Neural Networks", IEEE Access, vol. 9, pp. 155998-156012, 2021, doi: 10.1109/ACCESS.2021.3129149.
6. J. C. Ruiz, D. de Andrés, L. J. Saiz-Adalid, J. Gracia-Morán, "Zero-Space In-Weight and In-Bias Protection for Floating-Point-based CNNs", 2024 19th European Dependable Computing Conference (EDCC), pp. 89-96, 2024.
7. International Roadmap for Devices and Systems. [Online]. https://irds.ieee.org/editions (Accessed May 28, 2024).

8. B.L. Bhuva et al.: "Multi-cell soft errors at advanced technology nodes", IEEE Transactions on Nuclear Science, vol. 62(6), pp. 2585-2591, 2015.
9. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
10. J. Bazán Andría, (2022). Implementación en una RaspBerry Pi de una red neuronal convolucional con propiedades de tolerancia a fallos. Universitat Politècnica de València. http://hdl.handle.net/10251/187563
11. F.F. d. Santos et al., "Analyzing and Increasing the Reliability of Convolutional Neural Networks on GPUs", IEEE Transactions on Reliability, vol. 68, no. 2, pp. 663-677, June 2019, doi: 10.1109/TR.2018.2878387.
12. S. Liu, P. Reviriego, P. Montuschi, and F. Lombardi, "Less-is-Better Protection (LBP) for memory errors in kNNs classifiers", Future Generation Computer Systems, vol. 117, pp. 401-411, 2021, https://doi.org/10.1016/j.future.2020.12.015.
13. H. Guan et al., "In-Place Zero-Space Memory Protection for CNN", 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Canada, 2019.
14. K. Huang et al., "Improve Robustness of Deep Neural Networks by Coding", 2020 Information Theory and Applications Workshop (ITA), 2020, pp. 1-7, doi: 10.1109/ITA50056.2020.9244998.
15. D.T. Nguyen, N.M. Ho, and Ik-Joon Chang, "St-DRC: Stretchable DRAM Refresh Controller with No Parity-overhead Error Correction Scheme for Energy-efficient DNNs", 56th Annual Design Automation Conference 2019, June, Article No.: 205, Pages 1–6, https://doi.org/10.1145/3316781.3317915.
16. S. Burel, A. Evans and L. Anghel, "Zero-Overhead Protection for CNN Weights," 2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Athens, Greece, 2021, pp. 1-6, doi: 10.1109/DFT52944.2021.9568363.
17. A. Bosio, P. Bernardi, A. Ruospo and E. Sanchez, "A Reliability Analysis of a Deep Neural Network", 2019 IEEE Latin American Test Symposium (LATS), Santiago, Chile, 2019, pp. 1-6, doi: 10.1109/LATW.2019.8704548.
18. A. Ruospo et al., "Assessing Convolutional Neural Networks Reliability through Statistical Fault Injections," 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1-6, doi: 10.23919/DATE56975.2023.10136998.
19. B. Feinberg, S. Wang and E. Ipek, "Making Memristive Neural Network Accelerators Reliable", 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2018, pp. 52-65, doi: 10.1109/HPCA.2018.00015.
20. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
21. MNIST database: http://yann.lecun.com/exdb/mnist/
22. A.S. Rakin, Z. He and D. Fan, "Bit-Flip Attack: Crushing Neural Network with Progressive Bit Search", 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1211-1220, doi: 10.1109/ICCV.2019.00130.
23. A. Benso and P. Prinetto, editors. Fault injection techniques and tools for VLSI reliability evaluation. Kluwer Academic Publishers; 2003.
24. D. Gil-Tomas, J. Gracia-Moran, J.C. Baraza-Calvo, L.J. Saiz-Adalid, P.J. Gil-Vicente, "Injecting Intermittent Faults for the Dependability Assessment of a Fault-Tolerant Microcomputer System", IEEE Transactions on Reliability, Vol. 65(2), pp. 648 – 661, 2016.
25. J. C. Ruiz, D. de Andrés, L.J. Saiz-Adalid, J. Gracia-Morán, "In-Memory Zero-Space Floating-Point-based CNN Protection Using *Non-Significant* and *Invariant* Bits", Accepted, 43rd International Conference on Computer Safety, Reliability and Security (SafeComp), 2024.

In situ dielectric characterization as a tool towards more sustainable industrial processes

Beatriz García-Baños, Adrian Miró-Sanz, Jose M. Catalá-Civera

[1] ITACA Institute, Universitat Politècnica de València, Camino de Vera s/n
46022, Valencia, Spain
beagarba@upvnet.upv.es
admisan@itaca.upv.es
jmcatala@dcom.upv.es

Abstract. In this study, microwave dielectric thermal analysis (MW-DETA) was employed to improve the sustainability of two typically energy-intensive processes: pigment synthesis and the recycling of steel industry wastes. This real-time characterization technique facilitated the microwave process optimization by enabling reductions in reaction temperatures and simplifications of raw material mixtures. Practical examples are provided, demonstrating how the analysis of dielectric properties in relation to various processing parameters yielded optimization strategies. These strategies enhanced process efficiency, minimized resource consumption, and improved product quality. The insights gained from MW-DETA contribute to the optimization of microwave processes, thereby promoting their lower environmental impact.

1 Introduction

In order to minimize the climate change, reducing CO2 emissions and enhancing resource efficiency are critical objectives. These goals are particularly important for energy-intensive industries with typically high environmental impact such as ceramics and steel. Electrifying this type of processes using microwave technology presents a promising solution, offering an alternative based on electricity and at the same time providing advantages like shorter processing times, higher efficiencies, and compact device design. In this context, microwave technology appears as a promising candidate for sustainability by enabling the use of renewable energy sources instead of fossil fuels.

Despite the well-documented advantages of microwave-driven processes, their industrial application remains limited due to insufficient understanding of the fundamental interactions between microwave fields and matter. Consequently, as microwave applicators are developed for industrial applications, there is a growing demand for analytical techniques that assist process designers in precisely defining and optimizing process steps and conditions for these new microwave-based processes [1-3].

Microwave-driven processes are predominantly influenced by the dielectric properties (dielectric constant and loss factor) of the materials involved. These properties, which

C.Fernandez-Llatas and M.Guillen (Ed.):
Workshop on Innovation on
Information and Communication Technologies
(ITACA-WIICT 2024)

ISBN 978-84-09-66630-0

are unique to each material, determine whether a material will reflect, absorb, or transmit the microwave field. They are intrinsically linked to the material's composition and microstructure and are also influenced by parameters such as frequency, temperature, density, and moisture content. Accurate knowledge of these dielectric properties provides crucial insights into the behavior of materials during processing. The absence of this knowledge has been identified as a significant barrier by numerous researchers and engineers in the field [1,4].

In this context, microwave dielectric thermal analysis (MW-DETA) has recently emerged as a novel approach for providing in situ, accurate values of material dielectric properties as a function of temperature under the influence of microwave fields [3]. The aim of this study is to demonstrate how this technique serves as a convenient and valuable tool for acquiring insights into microwave processes and facilitating their optimization. To this end, two distinct processes were chosen: the synthesis of ceramic pigments and the recycling of steel wastes (carbothermic reduction of iron-bearing products). These processes are representative of industries characterized by high energy and resource consumption, thus presenting clear opportunities for leveraging microwave technology and the information obtained from the application of MW-DETA.

1.1 Experimental setup

The MW-DETA setup (Figure 1) is a dual-mode cylindrical cavity, with one mode (TE111 at a frequency 2.45 GHz) for heating a sample (15mm height and 10mm diameter) and another mode (TM010) for measuring its dielectric properties as a function of the temperature (from room temperature to approx. 1200ºC). The sample is inserted in a cylindrical quartz holder, and the cavity has holes for sample inspection. The MW-DETA can be combined with in situ Raman and mass spectrometry as described elsewhere [3].

The sample *bulk* temperature is determined from surface temperature measurements from a IR pyrometer after the application of a thorough calibration procedure described in [5].

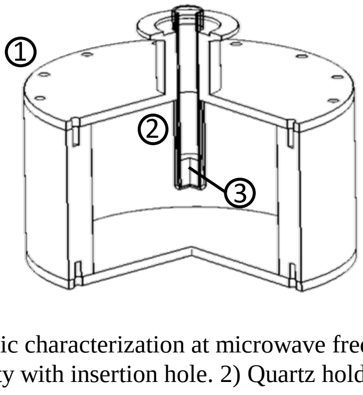

Fig. 1. Set up for dielectric characterization at microwave frequencies. 1) Microwave cylindrical cavity with insertion hole. 2) Quartz holder. 3) Sample.

The characterization method is based on the measurement of some microwave magnitudes as the resonant frequency and quality factor of the cylindrical cavity with the test sample being heated with microwave energy. The presence of the sample modifies the resonance properties of the microwave resonant cavity and these changes are employed to determine the dielectric properties. The test has an accuracy of ±(1-2)% for the dielectric constant and ±(2-5%) for the loss factor [3].

2 Application of MW-DETA to the steel industry

The recycling of residues from the steel industry involves the following process: the residues (mixture of iron oxides) are mixed with a certain percentage of carbon and heated at high temperatures (>1000ºC), causing their carbothermic reduction to recover metals such as iron and zinc. This kind of residues presents very good absorption of microwave energy, thus microwave technology is a very promising option with high energy savings compared to conventional heating methods.

Carbothermic reduction of an iron- and zinc-bearing waste was studied through MW-DETA up to 1000ºC, and the results are presented in Figure 2. Differential Scanning Calorimetry (DSC) of the same sample is also presented.

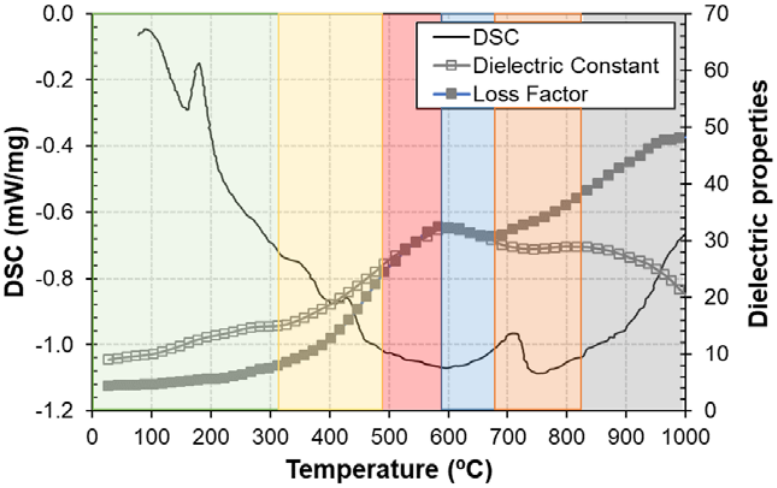

Fig. 2. Measurements of dielectric properties (dielectric constant and loss factor) and Differential Scanning Calorimetry (DSC) of an iron-bearing residue. The color bands represent the different process stages.

The information provided by these two techniques allowed the identification of different stages during the carbothermic reduction process, represented as different color bands in the Figure:

- Green: (23°C - 330°C) Dehydration of the sample and gradual increase of dielectric properties with the temperature
- Yellow (330°C - 480°C): Dehydroxilation of calcium hydroxide that leads to a higher loss factor (higher capacity to absorb microwave energy)
- Red (480°C - 590°C): Decomposition of zinc ferrite, magnetite and coke, with further increase of dielectric properties
- Blue (590°C - 680°C): Reduction of magnetite to wüstite, accompanied by a drastic decrease of dielectric properties (sub-products have less capacity to absorb microwaves)
- Orange (680°C - 840°C): Wüstite reduction to metallic iron (first desirable product) that again increases the microwave absorption
- Grey (840°C - 1000°C): Reduction of zinc oxide and evaporation of zinc (second valuable product)

The most interesting result from this characterization, is that the reduction of magnetite and wüstite (blue and orange stages) that lead to the release of pure metallic iron occur at temperatures above 580ºC as opposed to the 650ºC required when the same process is performed with conventional heating [6]. This is in accordance to previous observations in the literature [6,7] and can be related to the particular capacity of microwaves to act as a reducing agent [8].

The MW-DETA analysis was also performed to evaluate the effect of adding a certain quantity of carbon to the initial mixture, which is the additive commonly employed to enhance the reduction process in the conventional procedure. Figure 3 shows the dielectric properties of two samples with different carbon content (C/O ratios of 50% and 100%).

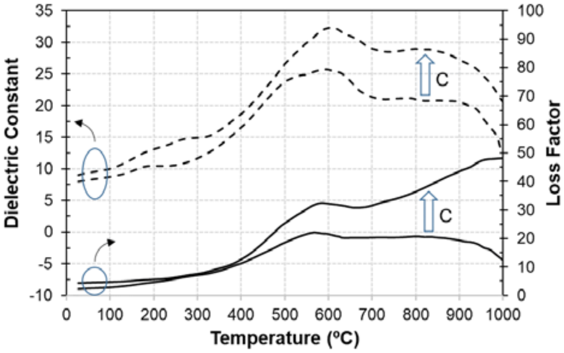

Fig. 3. Measurements of dielectric properties (dielectric constant and loss factor) of iron-bearing residues mixed with carbon (C). The arrows indicate the direction of higher carbon content.

As expected, the sample with higher carbon content presents higher dielectric properties due to the good absorbance of carbon. However, the dielectric curves with temperature reveal that the different stages of the process occur in approximately the same temperature ranges. The main conclusion derived from this analysis is that a high carbon content can be avoided in the microwave-driven process, thus it is possible to reduce the quantity of this additive optimizing the use of resources and the emitted CO_2.

3 Application of MW-DETA to the ceramic industry

In a second example, MW-DETA was applied to the ceramic sector, where microwave technology allows efficient and environmental-friendly synthesis of pigments.

MW-DETA was employed in this case to evaluate the microwave synthesis of ceramic pigments. Chromium black hematite was selected as a reference pigment because it is most extensively used for coloring glazes and ceramic bodies. The traditional synthesis of this ceramic pigment includes the calcination of the raw mixture at high temperature, 1100ºC during 2 hours. At this temperature the metal oxides react with each other to generate the new crystalline structure that forms the pigment:

$$\boldsymbol{Cr_2O_3 + Fe_2O_3 \xrightarrow{1050-1100℃} (Fe,Cr)_2O_3}$$

In this case, a sample of the raw mixture was introduced and measured in the MW-DETA setup under microwave irradiation, and the dielectric curves of the sample where analysed with temperature during the synthesis.

Figure 4 shows the main results from this analysis, where the dielectric constant revealed the following stages:

1. 20-600°C: At low temperatures, the dielectric properties increase slowly with temperature, with gradual increase of its capacity to absorb microwave energy.
2. 650°C: The dielectric constant values <1 indicate that a metallic behavior starts to predominate, so the material cannot be considered as a pure dielectric [9]. At the same time, the loss factor increases leading to a higher absorption of microwave energy.
3. 1000ºC: Minimum in dielectric constant indicates the synthesis temperature, which perfectly correlates with the endothermal peak observed in DSC. This means that the pigment crystal formation occurs at the same temperature than in the conventional process.
4. >1000ºC: Crystal growth is observed by a drastic increase of the dielectric properties at high temperatures.

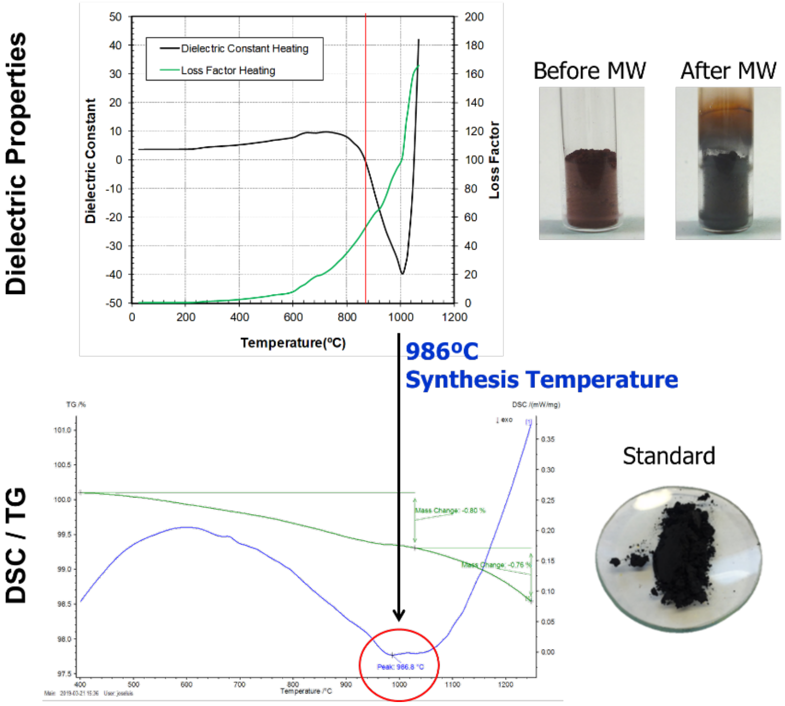

Fig. 4. Dielectric properties (dielectric constant and loss factor) of a ceramic pigment (Chromium black hematite) and correlation with DSC. Up-right: sample before and after the microwave treatment, Down-right: standard pigment obtained with the conventional method.

The pigment sample was processed at the required temperature (1000ºC according to the peak in the dielectric properties) during 15 minutes. Despite of the shorter processing time compared to the conventional process (2 hours), the composition analysis (X-ray Diffraction) showed the same results regardless of the processing method (See Figure 5).

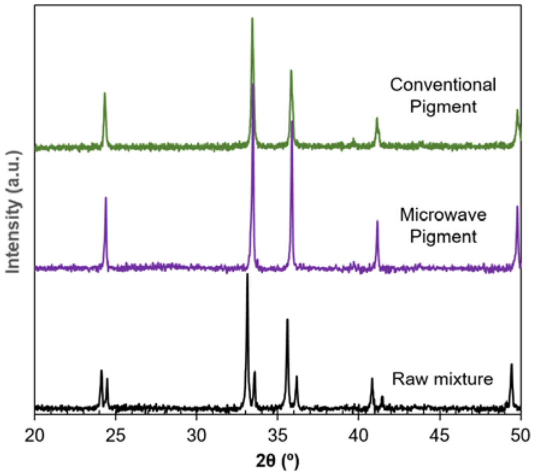

Fig. 5. X-ray diffraction of the raw mixture, the microwave and the conventional pigment showing the same final composition.

MW-DETA was further applied to optimize the raw materials mixture. For this purpose, two different types of chromium where considered in the mixture: Type A with specific surface area of 5 and Type B, with specific surface area of 2. The objective was to evaluate the process by studying the dielectric curves and select the most appropriate chromium.

Figure 6 shows the comparison of the pigment dielectric constant containing both types of chromium. The position of the minimum which represents the synthesis temperature is clearly higher (1000ºC) in the case of having chromium Type A, compared to the mixture containing chromium type B (900ºC). Thus, selecting chromium Type B for the pigment composition allows to reduce in approx. 100ºC the process temperature, having a clear benefit on the total energy consumption.

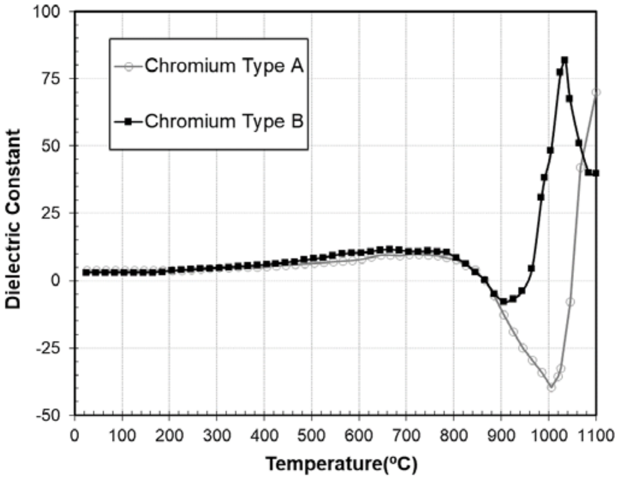

Fig. 6. Comparison of the dielectric constant of the ceramic pigment with two different types of Chromium in the raw materials mixture.

4 Conclusion

MW-DETA has been proposed as an efficient and precise tool for in situ determination of dielectric properties of materials during microwave heating processes. The analysis of dielectric curves has yielded critical insights into material behavior at various temperatures, reaction stages, and the influence of different parameters.

In the first case, MW-DETA was employed to analyze the recycling of steel wastes. The dielectric properties demonstrated the capability of microwave fields to act as a reducing agent, which significantly lowers the required temperatures compared to conventional processes. Furthermore, the dielectric curves indicated that the amount of carbon necessary in the mixture is reduced in the microwave-driven process. This information directly translates into substantial savings in energy consumption and resource utilization.

In the second case, the synthesis of ceramic pigment was investigated using MW-DETA. While the required temperatures were consistent across heating methods, the microwave synthesis required only 15 minutes compared to the 2 hours needed in the conventional method. The synthesis temperature was clearly identifiable in the dielectric constant curves, allowing for the selection of raw material compositions that required lower synthesis temperatures, thereby optimizing energy consumption.

In conclusion, MW-DETA analysis supplied essential information for the development of potential microwave industrial plants across various energy-intensive sectors, significantly reducing their environmental impact. This work represents a step towards a more sustainable industry by promoting the electrification of industrial processes and enhancing resource efficiency.

7 Acknowledgements

This work was funded by the European Union’s Horizon 2020 research and innovation
programme under grant agreement number 820783.
The authors wish to thank Al Farben S.A. for providing the pigment samples, and K1-MET for providing the steel wastes.

References

1. P. Ramos, D. Albuquerque and J. Pereira. “Numerical simulation and optimization of the ceramic pigments production process using microwave heating”, Chem. Eng. and Proc. – Proc. Intens. 169, 108567, 2021.
2. B. Garcia-Baños, J.M. Catalá-Civera, F.L. Peñaranda-Foix, P. Plaza-González and G. Llorens-Vallés, “In Situ Monitoring of Microwave Processing of Materials at High Temperatures through Dielectric Properties Measurement”, Materials 2016, 9(5), 349.
3. J.M. Catala-Civera, A.J. Canós-Marín, P. Plaza-González, J.D. Gutiérrez Cano, B. García-Baños and F.L. Penaranda-Foix, “Dynamic Measurement of Dielectric Properties of Materials at High Temperature During Microwave Heating in a Dual Mode Cylindrical Cavity” IEEE Trans. on Microw. Theory Techn., Vol. 63, pp. 2905-2914, 2015.
4. Rao, K.J.; Vaidhyanathan, B.; Ganguli, M.; Ramakrishnan, P.A. Synthesis of Inorganic Solids Using Microwaves. Chem. Mater. 1999, 11, 882–895.
5. García-Baños, B.; Jimenez-Reinosa, J.; Penaranda-Foix, F.L.; Fernandez, J.F.; Catalá Civera, J.M. Temperature Assessment of Microwave-Enhanced Heating Processes. Sci. Rep. 2019, 1, 10809.
6. Agrawal, D. Microwave sintering of metal powders. In Advances in Powder Metallurgy; Chang, I., Zhao, Y., Eds.; Woodhead Publishing Limited: Sawston, Cambridge, UK, 2013; pp. 361–379.
7. Omran, M.; Fabritius, T.; Heikkinen, E.; Chen, G. Dielectric properties and carbothermic reduction of zinc oxide and zinc ferrite by microwave heating. R. Soc. Open Sci. 2017, 4, 170710.
8. Serra, J.M.; Borrás-Morell, J.F.; García-Baños, B.; Balaguer, M.; Plaza-González, P.; Santos-Blasco, J.; Catalán-Martínez, D.; Navarrete, L.; Catalá-Civera, J.M. Hydrogen production via microwave-induced water splitting at low temperature. Nat. Energy 2020, 5, 910–919.
9. Yao, X.; Kou, X.; Qiu, J.; Moloney, M. Generation Mechanism of Negative Dielectric Properties of Metallic Oxide Crystals/Polyaniline Composites. J. Phys. Chem. C 2016, 120, 4937–4944.

The ITACA-WIICT is a meeting forum for scientifics, technicians and other professionals who are dedicated to Information and communication technologies study and research. Its fundamental scope is to promote the contact among scientific and professionals, improving the cooperation as well as the technological transfer among professionals.

Institute ITACA

Information and Communication Technologies