# Detecting Cryptographic Hash Functions through Electromagnetic Side-Channel Analysis

Gayan Akmeemana[1], Dharshana Kasthurirathna[2], and Asanka P. Sayakkara[3]
[1]E Gravity Solutions (Pvt) Ltd., Sri Lanka
[2]Sri Lanka Institute of Information Technology, Sri Lanka
[3]University of Colombo School of Computing, Sri Lanka
[1]gayanakmeemana.mc@gmail.com [2]dharshana.k@sliit.lk [3]asa@ucsc.cmb.ac.lk

***Abstract*—In the era of Industry 4.0, the Internet of Things (IoT) has emerged as a transformative force, with the proliferation of IoT devices permeating various aspects of our daily lives. However, this rapid adoption of IoT technology has also given rise to an alarming increase in cyberattacks targeting these devices. Among many avenues of cybersecurity, Electromagnetic Side Channel Analysis (EM-SCA) stands as a crucial branch of information security that enables attackers to eavesdrop on and exfiltrate sensitive information, making it a critical concern for IoT security. Among various security measures taken on IoT platforms, data integrity is ensured through cryptographic hash functions. This work explores the potential of utilising EM-SCA to detect the presence of cryptographic hash functions on IoT devices, which would play an important role at the surveillance stage of an attack. In pursuit of this objective, this study employs a set of supervised Machine Learning (ML) algorithms that are intricately crafted to automatically identify distinct patterns of EM radiation emissions associated with different hash algorithms. The results of this investigation demonstrate that the proposed methods can achieve classification accuracy rates exceeding 80%. The findings of this work highlights that an attacker can inspect an IoT device in a non-invasive manner to identify its critical data integrity mechanisms before a suitable subsequent action is taken to compromise it.**

***Index Terms*—Electromagnetic Side-Channel Analysis, Cryptography, Hash Functions, Internet of Things**

## I. Introduction

The term *Industry 4.0*, also known as the Fourth Industrial Revolution, was officially unveiled at the World Economic Forum's yearly meeting in 2016 [1]. This revolution marks a significant transition towards a digital era, where the fabric of our world is increasingly woven with digital data at its core. This transformative shift is reshaping the way we conduct business, industry, and daily life, with digital data serving as the linchpin for these changes [2] [3].

The *Industry 4.0* primarily aims to leverage cutting-edge technologies in the production and manufacturing processes, leading to heightened automation and decreased human involvement [4]. Hermann et al. outlined four key pillars within *Industry 4.0*: Interconnection (which involves the seamless linkage of people, machines, and sensors to one another, creating a networked ecosystem.); Information Transparency (the focus here is on augmenting the number of interconnected objects and individuals, leading to improved data sharing and visibility); Decentralized Decisions (the physical world is managed automatically through decisions made by embedded computers and sensors, enabling greater autonomy and efficiency); and Technical Assistance (the integration of technical support for decision-making processes, enhancing the quality and speed of decision-making) [3]. These components collectively define the core principles of *Industry 4.0*, designed to usher in a new era of enhanced productivity and operational efficiency.

The main driver of *Industry 4.0* is connecting the Internet of Things (IoT) to manufacturing. In this context, *Things* refers to things such as sensors, actuators, and even mobile phones that work together to achieve common goals in Cyber-Physical Systems in Smart Factories. The core components of *Industry 4.0* include IoT, Cyber-Physical Systems (CPS), and Smart Factories. CPS are responsible for making the linkage between the physical world and the virtual world. Inside the CPS, computations are made by the embedded devices, and make decisions to control the physical process with the help of feedback loops. By integrating IoT and CPS, Smart Factories are playing a major role in *Industry 4.0* [3].

Verma et al. pinpointed seven potential threats and vulnerabilities that could impact the security of *Industry 4.0* [5]:

- Cyber Attacks: In Industry 4.0, the more things are connected and work together, the more they can be targeted by cyber-attacks. Bad actors, such as cybercriminals, can find weaknesses in IoT gadgets, network setups, and software to get into systems without permission. They might do this to take valuable information, mess up important tasks, or demand money to fix things. There are different types of attacks they can use, like DoS/DDoS attacks, Man-In-The-Middle attacks, and Ransomware attacks [5].
- Data Breaches: In Industry 4.0, a lot of data is floating around, which can be a problem. Bad actors, such as cybercriminals, might try to steal important data, e.g., business secrets, customer info, or financial records. This can harm a company's reputation and cost them money [5].
- Insider Threats: Sometimes, the people working inside a company can be a big risk for Industry 4.0 systems. This includes employees who might want to harm on purpose. They have the right to access important data and systems, and they can cause problems either on purpose or by mistake [5].
- Physical Threats: There are also physical dangers to

worry about in Industry 4.0. Things such as theft, vandalism, or natural disasters can harm the devices and structures that Industry 4.0 depends on. When these things get damaged or disrupted, it can cause serious problems [5].

- Supply Chain Vulnerabilities: Because Industry 4.0 systems are all linked together, they can be at risk from supply chain attacks. This means that cybercriminals might go after the companies that supply parts or make the things used in Industry 4.0. By doing this, they can mess with the supply chain and cause problems like malware infections, data breaches, and other security issues [5].
- Lack of Standards and Regulations: In Industry 4.0, it is a problem that there are no clear rules and standards. This can make things less secure. Without agreed-upon ways of keeping things safe and no set of best practices, companies that make things might have trouble making sure their systems are secure and work well [5].
- Human Error: Even though Industry 4.0 uses fancy technology, people can still make mistakes that cause security problems. Workers might accidentally mess things up by doing things, such as setting things up the wrong way, not managing passwords well, or falling for tricks in emails [5].

A significant security concern in Industry 4.0 is data privacy. In this context, people share their personal information with IoT devices, thinking of this data as valuable. However, due to security vulnerabilities in IoT devices, this information can become exposed to the world. Typically, IoT devices have limited resources, such as low-cost components, basic processors, and small random-access memory. Despite being able to connect to the internet, these devices often struggle to ensure their security. Adding extra security measures to them can be costly and might affect their performance [2] [5]. This research address this problem using machine learning mechanism to detect running hash algorithms inside an IoT device, such as NodeMCU, using Electromagnetic Radiation (EM) emission.

### A. Motivation and Research Gap

Most of the large security problems come from the hardware, where attackers can take information right from the physical parts that our safe and coded software uses. Side-channel analysis is one of the most serious dangers for hardware security [6]. There are several Side-channel attacks, and Electromagnetic Side-channel Analysis (EM-SCA) is one of the very important types of attack among them. Using EM-SCA, hackers have managed to steal important data, like cryptographic keys, from computers and even IoT devices [7]. This shows that Side-channel analysis works well against many security protections on computer systems [8].

When considering cost-effective IoT edge devices, the NodeMCU is a simple microcontroller that can connect to multiple sensors simultaneously, which makes it safer, affordable, and cost-effective [9]. It is good at handling tasks, such as controlling lights, keeping an eye on the temperature, and improving location security such as home security [10]. NodeMCU-based devices can be a victim of crypto mining if it is attacked by malware to manipulate their device functions and hardware [9]. In such scenarios, hash calculations will increase inside the device since crypto mining uses more hash calculations. As a result, the need arises to detect hash calculations happening inside the NodeMCU platform without accessing the device inside.

### B. Research Questions

In this work, the objective is to find a way to use EM-SCA to detect hash algorithms implemented on the NodeMCU platform. Towards this goal, the research focuses on answering the following two questions:

*1) Research Question 1:* What is the most suitable method to extract the internal program behaviour of NodeMCU devices in security and forensic analysis?

Hypothesis: Electromagnetic radiation emitted by NodeMCU devices reveals sufficient information about their internal operations. The application of Electromagnetic Side-Channel Analysis (EM-SCA) methods is feasible to extract this information, providing valuable insights for digital investigations.

*2) Research Question 2:* How to classify hash algorithms which are implemented on the NodeMCU using EM radiation data?

Hypothesis: Machine Learning algorithms can be used to classify hash algorithms by training against extracted EM radiation data.

The rest of the paper is organized as follows: Section II presents existing methods to recover information from electromagnetic side-channel analysis for hardware; Section III describes the experimental procedure for detecting hash algorithms by electromagnetic side-channel analysis in NodeMCU; Section IV discusses the experiment results and Section V concludes the paper.

## II. Related work

Side-channel Analysis (SCA) poses a significant danger to embedded systems. SCA encompasses various attack techniques that rely on the leakage of different side-channel information types. This includes signals related to power consumption, electromagnetic emissions, and timing data [11]. These can be categorized based on their exploitation of how the system consumes power, emits electromagnetic radiation, and handles timing during processes. Various power analysis methods, including statistical and machine learning approaches, have been successful in attacking practical systems containing encryption algorithms [11].

Kocher et al. performed timing attacks on Diffie-Hellman, RSA, and DSS algorithms. The attack can be viewed as a problem of detecting a *signal*. This signal is characterized by timing variations caused by the specific target exponent bit of the algorithm. On the other hand, *noise* arises from inaccuracies in measurements and timing variations due to unknown exponent bits. The signal and noise characteristics dictate the number of timing measurements needed for the

attack. This is a very old and very first experiment that tries to break the cryptosystem by using side-channel analysis [12]. Later, Kocher et al. demonstrated that the power consumption of a computing device can serve as a side-channel through which cryptographic keys can be extracted using Simple Power Analysis (SPA) and Differential Power Analysis (DPA) [13] [14].

During cryptographic operations executed by a CPU, the power consumption of the device corresponds to the values stored in its registers. Through the accumulation of an ample number of power consumption traces acquired during cryptographic operations with identical keys, it becomes feasible to disclose the key using techniques such as differential power analysis (DPA). The CPU's power consumption is closely linked to the electromagnetic (EM) radiation produced by the device, thus exposing the EM side-channel. Consequently, advanced versions of power analysis algorithms, such as differential electromagnetic analysis (DEMA), were subsequently developed to retrieve cryptographic keys by utilizing EM traces [15].

Electronic circuits consume power and emit Electromagnetic (EM) radiation as they operate. This power/EM side-channel information can be leveraged to uncover data processed within the internal states of a functioning system. For instance, side-channel data from a mathematically secure cryptographic algorithm running on a CPU, Field-Programmable Gate Array (FPGA), or Application-Specific Integrated Circuit (ASIC) can be employed to disclose the algorithm's operations and the secret key of it processes [16].

Aknesil et al. conducted work to detect memory leakage on Raspberry Pi 3 Model B v1.2 by measuring EM radiation [16]. They concluded that the electromagnetic traces of memory operations on a Raspberry Pi 3 provide insights into the data that is either read from or written to the main memory. They used deep learning-based side-channel analysis (SCA) and it accurately retrieved all bytes within the 32-bit data field of memory operations from a single trace, achieving accuracy levels between 49% and 86% [16].

Hatun et al. also performed side-channel analysis on Raspberry Pi by running the RSA algorithm. In their research, they outlined the stages of the RSA algorithm and applied different analyses to traces associated with specific segments of the algorithm. In this research, the RSA algorithms running on a Raspberry Pi were tested with SEMA and DEMA attacks. When the SEMA attack was used, it revealed that in an implementation without countermeasures, all the key bits could be obtained with just one measurement. However, it was found that the key could not be obtained using SEMA when the algorithm was designed to resist such attacks. The DEMA attack required more measurements and correlation analysis to retrieve the bit value of the key, and it successfully obtained the key [17].

Sayakkara et al. researched capturing weak EM signals and using computers with sufficient processing power to analyze them. It can capture and analyze against any kind of computing device to detect software anomaly detection and cryptographic key recovery [18]. Juyal et al. introduced a special high-gain flat antenna designed for capturing electromagnetic emissions

Fig. 1: Near field probes for oscilloscopes (adapted from [22]).

from smart electronic devices. Furthermore, they conducted an electromagnetic analysis attack on an IoT board, specifically the Raspberry Pi [19]. Gunathilake et al. employed an oscilloscope along with near-field EMC (electromagnetic compatibility) probes to gather EM signals originating from an Arduino UNO board running the PRESENT block cipher. To enhance the signals, they utilized a 40-dB wide-band amplifier before sending them to the oscilloscope. The primary aim of this research was to identify any information leakage from the Arduino board while the encryption algorithm was in operation. They employed SEMA and CEMA analyses, and the findings revealed that there was a possibility of seven key bytes leaking [20]. Sayakkara et al. utilized a software-defined radio (SDR), specifically the HackRF, to access the radio frequency emissions from the devices. SDRs have proven to be highly effective and are widely used for such side-channel attacks. They positioned an H-loop antenna close to a Raspberry Pi to capture electromagnetic radiation traces. All the collected electromagnetic traces were initially in the time domain and exhibited considerable variations in length. To facilitate analysis, they transformed these samples into the frequency domain using a Fourier transform. Subsequently, they standardized the values and employed neural network classifiers to assess these traces for different cryptographic algorithms [21].

Tirumaladass et al. have demonstrated the effective use of compact magnetic H-loop antennas for capturing electromagnetic (EM) radiation emitted by computing devices. In their research, they used deep learning (implemented using TensorFlow Keras) to classify three encryption algorithms (3DES, AES128 and AES256) with an accuracy of 95%. Near-field RF probes, also known as sniffer probes (see Fig 1), play a vital role in pinpointing the sources of emissions emanating from circuit boards or devices. These probes find significant use in conducting preliminary EMC testing on circuit boards or electronic products. Their primary purpose is to identify any electromagnetic interference (EMI) that exceeds the regulatory standards. These probes must be capable of precisely measuring both the electric and magnetic fields of a module. To be

effective, they need to maintain a consistent and unwavering frequency response to allow users to accurately locate emission sources [23].

Near-field probes can be broadly categorized into two types: Electric field (E-field) probes and Magnetic field (H-field) probes. E-field probes primarily detect electric fields and often have a stub antenna-like appearance. They are not significantly affected by their orientation when brought close to the device under test (DUT). E-field probes are well-suited for identifying emissions resulting from voltage changes in a circuit, including emissions on individual pins or PCB traces when they make direct contact with the circuits. H-field probes primarily respond to magnetic fields and often have a loop-like design. They are shielded to reduce sensitivity to electric field pickup. Magnetic fields result from changes in the current within a circuit. Unlike E-field probes, H-field probes are sensitive to their orientation when applied to the DUT. They specifically detect and respond to current within the same plane as the loop when scanning over the device. Magnetic field probes are particularly useful for detecting magnetic fields generated by sources such as clock signals, serial data streams, control signals, and switching power supplies, especially when these signals are created by oscillations or harmonics [23] [24].

A recently published work that is conceptually most similar to this research is presented by Robyns et al. [25]. They proposed a Convolutional Neural Network (CNN) architecture specifically designed for classifying operations carried out by the NodeMCU from a list of 8 potential operations. These include OpenSSL AES, native AES, TinyAES, OpenSSL DES, SHA1-PRF, HMAC-SHA1, SHA1, and SHA1Transform. Their CNN architecture was also used to forecast the beginning and end times of these operations, eliminating the need for firmware adjustments or manual triggers in SCA. To assess their approach, they employed a substantial dataset of 66 GB, encompassing 69,632 complex traces of EM leakage, acquired using an USRP B210 SDR device. The most successful version of their methodology demonstrated an accuracy rate of 96.47% in classification and managed to predict operation start and end times with an average deviation of 34 microseconds from the actual values.

Amrouche et al. emphasized the different research projects conducted within each individual category of SCA. In their study, they provided an overview of the primary SCA categories and their application of Machine Learning (ML) and Deep Learning (DL) techniques to access sensitive data [6].

## III. Methodology

### A. Experimental Setup

The experiments in this section uses a single *NodeMCU Amica* device as the Device Under Test (DUT). This device includes the *Espressif ESP8266MOD* WiFi module, which is commonly found in IoT devices, such as smart light bulbs, meters, and sensors. The ESP8266MOD is equipped with a Tensilica L106 32-bit RISC processor that normally runs at a default speed of 80 MHz. However, it can be overclocked to 160 MHz if needed [25].

The acquisition of EM radiation involved the utilization of a HackRF One Software-Defined Radio (SDR) Device

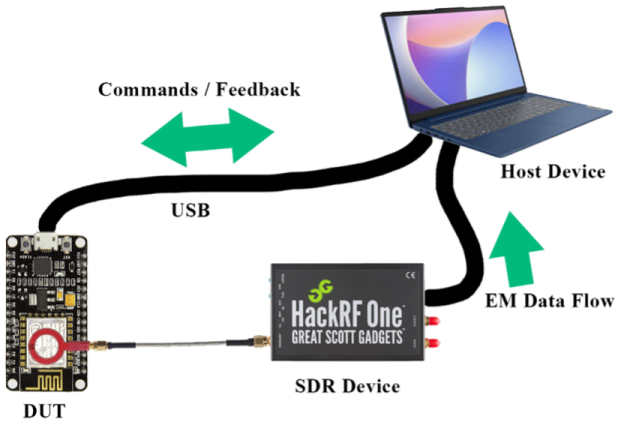



Fig. 2: The experimental setup used for the EM data acquisition from the target DUT.

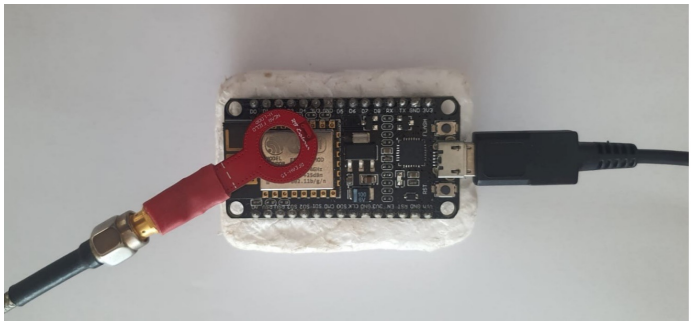


Fig. 3: The arrangement of the H-Loop antenna on top of the NodeMCU.

in conjunction with a compact H-Loop magnetic antenna, characterized by a diameter of 15 mm (see Fig 2). Establishing connectivity between the SDR device and the H-loop antenna was facilitated through a semi-rigid RF cable, permitting precise placement of the antenna in a predefined location throughout the extended experimental duration (see Fig 3). Both the SDR device and the Internet of Things (IoT) device under scrutiny were seamlessly linked to the same host computer.

This setup was strategically designed to foster experimental control and efficient storage of the captured EM data. By connecting the SDR device and the IoT device to a common host computer, a synchronized environment was established for streamlined experimentation and data management. This synchronized configuration not only facilitated the coordination of experiments but also ensured the seamless integration of EM data from both devices. The use of the HackRF One SDR Device, coupled with the H-Loop magnetic antenna, provided a robust platform for capturing and analyzing EM radiation, contributing essential insights to the overall experiment.

The host computer, a Dell Inspiron with a 3.25 GHz processor and 4 GB RAM running Ubuntu 22.4 LTS, played a pivotal role in the experimentation process. Equipped with essential tools, including the Arduino IDE, Gqrx SDR tool, and GNU Radio Companion, it provided a robust platform for

data acquisition. This computer served as the central hub for controlling the experiments, managing the connected HackRF One SDR Device, and processing the captured electromagnetic data. The specifications of the Dell Inspiron ensured a reliable and efficient environment for conducting experiments and analyzing the outcomes seamlessly. For data analysis, a virtual machine with a sufficient configuration was deployed, featuring a 3.10 GHz processor, 128 GB RAM, and operating on Windows 10. The Python environment utilized Anaconda 3, incorporating essential libraries such as NumPy, Pandas, SciPy, Matplotlib, TensorFlow, Scikit-learn, and Seaborn. This well-equipped virtual machine provided the computational power and software infrastructure necessary for conducting in-depth analyses on the captured electromagnetic data. The inclusion of prominent Python libraries ensured a comprehensive toolkit for statistical analysis, machine learning, and data visualization, contributing to a thorough exploration of the experiment's outcomes.

### *B. Custom Firmware and Protocol*

To streamline the acquisition of a pristine dataset of electromagnetic (EM) traces, a dedicated firmware was developed. The firmware was specifically designed to implement MD5 and SHA1 hash algorithms for capturing electromagnetic radiation. The experimentation process was segmented into three phases. Initially, the MD5 hash algorithm was executed in a loop to gather EM data. Algorithm 1 was employed to generate MD5 hashes, utilizing a lengthy text paragraph as the variable input. Throughout this procedure, 8 distinct datasets were obtained utilizing the HackRF one device in conjunction with the GNU Radio Companion.

**Algorithm 1** Execute MD5 Loop

```
Include Arduino.h
Include MD5Builder.h
MD5Builder _md5
function SETUP                          ▷ No setup code for now
end function
function LOOP
    String text                         ▷ Long paragraph text to hash
    md5(text)
end function
function MD5(String str)
    _md5.begin()
    _md5.add(str)
    _md5.calculate()
    Return _md5.toString()
end function
```

In the subsequent phase of experimentation, the focus shifted to the execution of the SHA1 Hash algorithm in a loop to amass electromagnetic (EM) data. Algorithm 2 played a central role in this process, leveraging the same extensive text paragraph as the variable input. A total of 8 datasets were meticulously acquired during this phase, employing the HackRF device in tandem with the GNU Radio Companion.

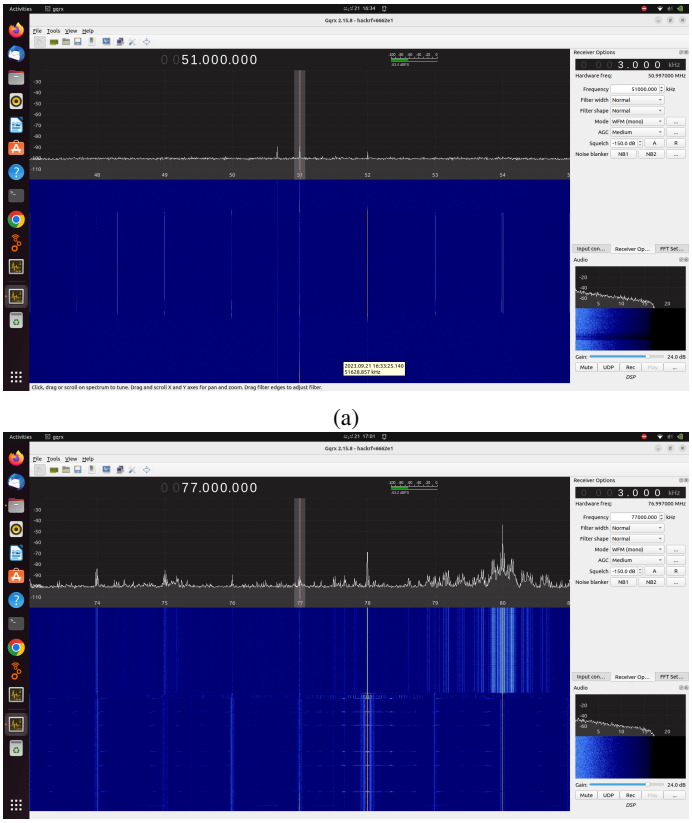
In the last segment of the experiment data collection, EM radiation data was collected without the execution of any specific algorithms, solely relying on the operational status of the NodeMCU device. During this phase, the NodeMCU was in a functional state, yet no dedicated algorithms were actively running. This approach aimed to capture and analyze the inherent EM radiation emitted by the NodeMCU in its normal operational state without the influence of additional algorithmic processes. This data collection phase provided valuable insights into the ambient EM emission of the NodeMCU device without the intentional modulation or interference introduced by algorithmic operations. Understanding the baseline EM radiation contributes to a comprehensive assessment of the device's natural electromagnetic behavior, which is essential for contextualizing and interpreting the results obtained during other experimental phases where specific algorithms were employed.

**Algorithm 2** Execute SHA1 Loop

```
1: Include Arduino.h
2: Include Hash.h
3: function SETUP                       ▷ No setup code for now
4: end function
5: function LOOP
6:     String text                      ▷ Long paragraph text to hash
7:     sha1(text)
8: end function
```

### *C. Data Capture and Storage*

In the experiment, three datasets of EM emanations were captured during the above-mentioned three parts with the labels *md5*, *sha1* and *nothing* using GNU Radio Companion, totalling 53.3 GB with the sample rate of 20 MHz [26]. One dataset contains eight data files in eight different frequencies, namely 51.7 MHz, 51.5 MHz, 52.8 MHz, 77.6 MHz, 79.5 MHz, 78.5 MHz, 81.2 MHz, and 82.3 MHz. To successfully detect EM radiation from a target IoT device, it's crucial to identify the frequency at which information is leaking. Various frequencies can provide valuable forensic insights, but there's currently no systematic method for precise identification. In such cases, one reliable frequency available is the clock frequency of the IoT device's MCU chip [27]. For example, a NodeMCU device typically operates at 80 MHz. Sometimes, external EM noise from other sources may overlap with this frequency, causing interference in EM radiation observation. In such situations, using a higher harmonic frequency of the clock frequency — essentially a multiple of the original frequency — can be employed as the information-leaking frequency [28]. Initially, two capturable data leaking frequencies, 52 MHz, and 80 MHz were identified by visualizing the captured EM data through the GQRX tool [29]. Fig 4 illustrates the view of GQRX software while observing the EM radiation signals. To determine the frequency, the EM radiation was observed by changing the center frequency of HackRF One while moving the H-Loop antenna closer to and farther from the NodeMCU.




(a)

(b)

Fig. 4: Visualization of the EM radiation emitted from the NodeMCU. (a) The center frequency is 51 MHz and the yellow line in the 52 MHz is the NodeMCU leakage frequency. (b) The center frequency is 77 MHz and the yellow line in the 80 MHz is the NodeMCU leakage frequency.

### *D. Development of the Classifier*

Machine-learning classification problem that is focused on this study includes classifying the pattern of electromagnetic radiation signals to comprehend the internal behaviours of Internet of Things devices. Neural network-based classifiers, which are very flexible by modifying several parameters, are used in this study. One fundamental neural network type is the Multi-layer Perceptron (MLP) [30], where information flows from the input layer to the output layer through one or more hidden layers.

Additionally, this study employs Long Short-term Memory (LSTM) architecture within neural networks [31]. An LSTM network includes feedback connections, enabling it to analyze sequences of data points collectively, in contrast to classical neural networks that process individual data points. This characteristic makes LSTM networks better suited for identifying patterns in time series data, such as EM traces [32].

An EM trace is essentially a vector that depicts how a signal's amplitude changes over time. Given the high-speed sampling employed in signal acquisition hardware, even a brief EM trace spanning milliseconds can comprise millions of data points. Using this raw data directly to train and test machine learning models can have drawbacks, primarily in terms of the time and computing resources required due to its high dimensionality. Consequently, the EM traces obtained through the described hardware setup are unsuitable for direct use in training machine learning models. To address this, the EM traces need preprocessing. This process transforms them from a continuous time-domain signal into a format that provides a manageable feature vector for machine learning models. To classify EM traces related to Hash algorithm activity, labelled EM traces are required. Initially, each EM trace is in the time domain, which makes it susceptible to external noise-induced fluctuations. To mitigate this, each trace is transformed into the frequency domain using the Fast Fourier Transform (FFT) while employing an overlapping sliding window approach [32]. This operation generates a set of FFT vectors that represent consecutive time intervals for each EM trace.

The dimensions of these FFT vectors are still too large, though, to be employed straight away as a feature vector for machine learning classification. The size of the FFT vectors are further lowered to remedy this. To do this, each FFT vector's elements are divided into 10,000 uniformly spaced bins. Without compromising generalizability, the maximum element from each bin is chosen to serve as the representative value. For each EM trace time period, this procedure yields a feature vector with 10,000 entries.

An empirical analysis was carried out to ascertain the best setup for a neural network that was supposed to identify EM traces. An input layer, two hidden layers, and an output layer made up the final neural network's four layers. Based on this empirical evaluation, the number of hidden layers and hidden nodes inside each hidden layer were chosen.

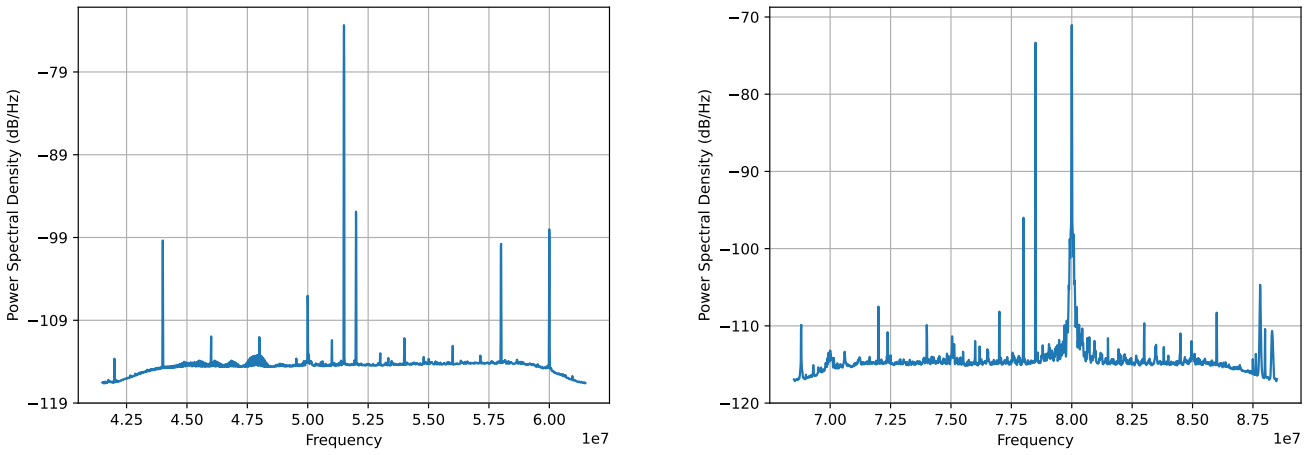
## IV. Results and Discussion

While using the GQRX tool to capture EM radiation, multiple frequencies were observed as adjusted the H-Loop antenna's distance from the NodeMCU.

After careful evaluation, it was determined that 52 MHz and 80 MHz were the most significant leakage frequencies due to their higher signal strength. The ESP8266 chip's datasheet [13] specifies an external crystal frequency range of 24 MHz to 52 MHz. Therefore, it is possible to reasonably attribute the 52 MHz leakage frequency to the NodeMCU. This conclusion is supported by the observation that the frequency appeared on the display when the H-Loop antenna was close to the chip and disappeared when the H-Loop antenna was moved away from the chip. Similarly, we identified 80 MHz as another significant frequency while adjusting the H-Loop antenna's position [25].

When plotting the Power Spectral Density (PSD) of the EM radiation emitted by the NodeMCU, a significant peak was observed at the centre of the plot (see Fig 5). The large spike is produced by the HackRF device and is referred to as a *DC offset* [33]. The documentation for the HackRF One device advises users to disregard it as it is a common occurrence in every measurement.

The development of an effective neural network for the precise classification of Electromagnetic (EM) traces was a crucial aspect of this study. The neural network architecture




(a) (b)

Fig. 5: Visualization of the EM radiation emitted from the NodeMCU. (a) The center frequency is 51 MHz and the yellow line in the 52 MHz is the NodeMCU leakage frequency. (b) The center frequency is 77 MHz and the yellow line in the 80 MHz is the NodeMCU leakage frequency.

TABLE I: Classification accuracy of hash algorithms.

| Activity | Precision | Recall | F1-Score |
|---|---|---|---|
| md5 | 0.84 | 0.92 | 0.88 |
| nothing | 0.94 | 0.95 | 0.95 |
| sha1 | 0.94 | 0.84 | 0.89 |

comprised seven layers, encompassing one input layer, five hidden layers, and one output layer. The determination of the optimal number of hidden layers and hidden nodes within each layer was grounded in empirical evaluation to ensure the network's efficacy.

The configuration of the hidden layers involved a strategic distribution of hidden nodes. The first, second, third, fourth, and fifth hidden layers comprised of 800, 500, 200, 100, and 50 hidden nodes in order. This hierarchical arrangement allowed the neural network to progressively extract and learn intricate features from the input data, facilitating a nuanced understanding of the underlying patterns in the EM traces.

The input layer played a pivotal role in processing the feature vector, accommodating 1024 input nodes. This comprehensive input layer design was instrumental in capturing the diverse aspects of the EM traces, providing the neural network with a rich set of information for classification. On the other end, the output layer was structured with three nodes, representing the three distinct classes corresponding to the hash algorithms and scenarios involving no activity.

The training process was a crucial phase in enhancing the neural network's classification capabilities. It involved the utilization of 240 samples for each class, culminating in a total of 720 training samples across the three classes. This robust training dataset ensured that the neural network gained a comprehensive understanding of the unique features associated with each class, enabling it to make accurate classifications during subsequent testing.

The empirical setting of the learning rate at 0.001 played a pivotal role in optimizing the neural network's training process. The learning rate influenced the magnitude of adjustments made to the network's weights during the training phase, striking a balance between convergence speed and avoiding overshooting the optimal weights. The empirical fine-tuning of this parameter underscored the meticulous approach taken to enhance the neural network's overall performance.

The culmination of these architectural considerations and training processes resulted in a neural network classifier that demonstrated commendable accuracy. The classification outcomes, as outlined in Table I, indicated an accuracy exceeding 80% in correctly categorizing the two hash algorithms and scenarios characterized by no activity. The training and validation loss curves of the trained model is illustrated in the Fig 6. Furthermore, the confusion matrix of the classifier's tests are illustrated in Fig 7. This level of accuracy signifies the efficacy of the developed neural network in discerning subtle patterns within the EM traces and highlights its potential applicability in real-world scenarios, particularly in the context of digital investigations involving NodeMCU devices.

## V. Discussion and Conclusion

The research addressed the primary inquiries posed at the commencement of the investigation. The inaugural question sought an effective methodology for extracting forensic insights from NodeMCU devices. The investigation has conclusively demonstrated that the application of EM-SCA serves as a viable means to obtain these crucial insights. By leveraging this technique, the study has successfully unveiled a pathway for security and forensic experts to extract valuable information from NodeMCU devices. Turning to the second research question, which sought to elucidate the classification

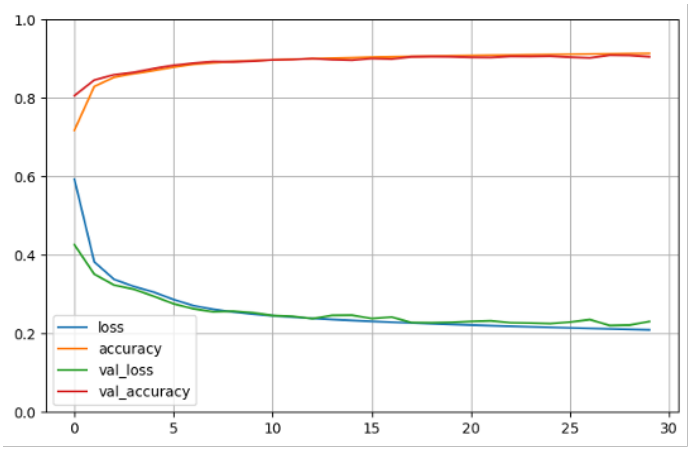



Fig. 6: Training and Validation Loss Curves for implemented model.

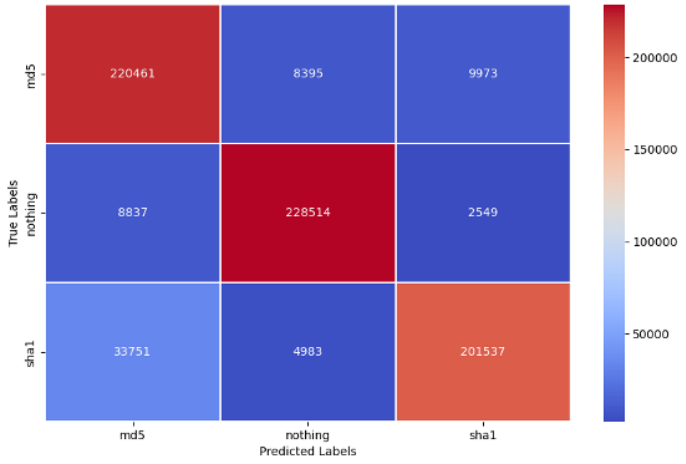



Fig. 7: The confusion matrix of classifying Hash Algorithms.

of hash algorithms implemented on the NodeMCU, the findings, delineated in Table 1, present a noteworthy outcome. The experimental results unequivocally indicate that an ML approach can be adeptly employed for the classification of these hash algorithms, achieving an accuracy rate surpassing 80%. This revelatory insight opens new possibilities for enhancing the efficiency and accuracy of forensic investigations involving NodeMCU devices. The success in addressing these two pivotal research questions not only contributes to the growing body of knowledge in the field but also has immediate practical implications. Forensic analysts and security experts can now rely on EM-SCA as a robust method for extracting pertinent information from NodeMCU devices. Furthermore, the integration of ML in hash algorithm classification on NodeMCU devices promises a significant leap in the accuracy of forensic analyses, providing a valuable tool for investigators grappling with the intricacies of embedded systems. In conclusion, the research has not only met but exceeded expectations in providing comprehensive answers to the research questions. The incorporation of innovative methodologies, such as EM-SCA and ML, showcases the potential for advancing forensic practices in the realm of NodeMCU device investigations. These findings mark a significant stride forward, underscoring the dynamism and adaptability required in the ever-evolving landscape of digital forensics.

### A. *Limitations*

This study is not without its inherent limitations, and a primary constraint stems from the challenges associated with conducting data analysis on virtual machines. The restrictive nature of virtual environments, particularly in terms of available random access memory (RAM), poses a significant obstacle to the seamless execution of model fitting processes. The intricate computational demands of model fitting, exacerbated by the limitations of virtual machine configurations, hinder the comprehensive exploration of the data, and may lead to sub-optimal results. Consequently, the findings derived from this study should be interpreted within the context of these computational constraints, acknowledging the potential impact on the overall analytical outcomes.

Another noteworthy limitation pertains to the physical characteristics of the NodeMCU chip, specifically its metal shield. The research primarily focused on the electromagnetic radiation emanating from the shield, recognizing it as a crucial aspect of the side-channel analysis. However, it is imperative to acknowledge that the presence of the metal shield introduces complexities in capturing and interpreting the full spectrum of electromagnetic signals. The shielding effect can potentially attenuate or modify the emitted radiation, influencing the accuracy and completeness of the forensic insights obtained. Consequently, the scope of this research is delimited by the challenges inherent in precisely characterizing the EM emission from within the shielded environment of the NodeMCU chip.

### B. *Future Works*

The culmination of the present experiment not only provides valuable insights into NodeMCU device forensics but also lays the foundation for potential avenues of future research and exploration. These promising directions encompass diverse aspects that could further enhance our understanding and capabilities in the realm of embedded system analysis.

One compelling avenue for future investigation is the exploration of alternative hash functions in the context of SCA. While the current study delved into the classification of hash algorithms on NodeMCU devices, focusing on commonly used ones, such as MD5 and SHA-1, there remains a vast landscape of cryptographic hash functions that warrant examination. For instance, the prevalence of SHA-256 and Scrypt in cryptocurrency applications raises intriguing possibilities for extending side-channel analysis to these algorithms. Given the unique characteristics of these hash functions and their paramount importance in securing digital transactions, understanding their behavior under electromagnetic radiation scrutiny could unravel new dimensions in forensic analysis. This avenue of research not only addresses the evolving cryptographic landscape but also bolsters the applicability of side-channel analysis to a broader spectrum of hash algorithms.

Another promising trajectory for future exploration involves leveraging convolutional neural networks (CNN) for the classification of hash algorithms on NodeMCU devices. The current study harnessed machine learning techniques for this purpose, achieving an accuracy exceeding 90%. However, the application of CNN, with its capacity to automatically learn hierarchical features, may provide a more nuanced and sophisticated approach to algorithm classification. CNN's ability to discern intricate patterns within the EM radiation data could potentially enhance the accuracy and efficiency of the classification process. This avenue of research aligns with the ongoing advancements in deep learning and artificial intelligence, offering a contemporary perspective on the intersection of machine learning and embedded system forensics.

Furthermore, a compelling avenue for future research lies in the development of Python scripts to identify the boundary lines of hash algorithms within the internal processes of NodeMCU devices. Creating a script that discerns and delineates the specific operations and transitions associated with hash algorithm execution can contribute significantly to forensic analyses. This script could aid investigators in pinpointing the exact moments when hash algorithms are invoked, enabling a more granular understanding of the internal workings of NodeMCU devices. Such insights could prove invaluable in reconstructing digital timelines and establishing a comprehensive narrative of events, enhancing the forensic toolkit available for NodeMCU investigations.

The findings of the current experiment not only underscore the potential applications of side-channel analysis in NodeMCU device forensics but also illuminate exciting avenues for future research. Exploring alternative hash functions, incorporating advanced ML techniques like CNN, and developing scripts for delineating internal processes all represent promising directions that can elevate the field of embedded system forensics to new heights. These prospective investigations not only respond to current technological trends but also anticipate the evolving challenges posed by emerging cryptographic practices and the intricate internal workings of embedded devices.

## REFERENCES


[1] W. E. Forum, "The fourth industrial revolution: what it means, how to respond," https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/, accessed On: 2024-06-12.

[2] M. M. H. Onik, K. Chul-Soo, and Y. Jinhong, "Personal data privacy challenges of the fourth industrial revolution," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2019, pp. 635–638.

[3] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016, pp. 3928–3937.

[4] S. J. Joshi, S. Mamaniya, and R. Shah, "Integration of intelligent manufacturing in smart factories as part of industry 4.0-a review," in *2022 Sardar Patel International Conference on Industry 4.0-Nascent Technologies and Sustainability for'Make in India'Initiative*. IEEE, 2022, pp. 1–5.

[5] P. Verma and N. Bharot, "A review on security trends and solutions against cyber threats in industry 4.0," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2023, pp. 397–402.

[6] A. Amrouche, L. Boubchir, and S. Yahiaoui, "Side channel attack using machine learning," in *2022 Ninth International Conference on Software Defined Systems (SDS)*. IEEE, 2022, pp. 1–5.

[7] A. P. Sayakkara and N.-A. Le-Khac, "Electromagnetic side-channel analysis for iot forensics: Challenges, framework, and datasets," *Ieee Access*, vol. 9, pp. 113 585–113 598, 2021.

[8] R. Poussier, V. Grosso, and F.-X. Standaert, "Comparing approaches to rank estimation for side-channel security evaluations," in *Smart Card Research and Advanced Applications: 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers 14*. Springer, 2016, pp. 125–142.

[9] K. T. Joseph *et al.*, "Analysis on iot networks security: Threats, risks, esp8266 based penetration testing device and defense framework for iot infrastructure," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*. IEEE, 2023, pp. 1–7.

[10] N. Rathour, V. Kumar, S. S. Kundu, Y. Gehlot, A. Gurung *et al.*, "Sigma home: An iot-based home automation using node mcu," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. IEEE, 2023, pp. 1317–1322.

[11] N. Mukhtar and Y. Kong, "Hyper-parameter optimization for machine-learning based electromagnetic side-channel analysis," in *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE, 2018, pp. 1–7.

[12] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16*. Springer, 1996, pp. 104–113.

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer, 1999, pp. 388–397.

[14] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, pp. 5–27, 2011.

[15] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," in *Smart Card Programming and Security: International Conference on Research in Smart Cards, E-smart 2001 Cannes, France, September 19–21, 2001 Proceedings*. Springer, 2001, pp. 200–210.

[16] C. Aknesil and E. Dubrova, "Towards generic power/em side-channel attacks: Memory leakage on general-purpose computers," in *2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2022, pp. 1–6.

[17] E. Hatun, G. Kaya, E. Buyukkaya, and B. O. Yalcin, "Side channel analysis using em radiation of rsa algorithm implemented on raspberry pi," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2019, pp. 1–6.

[18] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics," *Digital Investigation*, vol. 29, pp. 43–54, 2019.

[19] P. Juyal, S. Adibelli, N. Sehatbakhsh, and A. Zajic, "A directive antenna based on conducting disks for detecting unintentional em emissions at large distances," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 6751–6761, 2018.

[20] N. A. Gunathilake, A. Al-Dubai, W. J. Buchanan, and O. Lo, "Electromagnetic side-channel attack resilience against present lightweight block cipher," in *2022 6th International Conference on Cryptography, Security and Privacy (CSP)*. IEEE, 2022, pp. 51–55.

[21] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, "Leveraging electromagnetic side-channel analysis for the investigation of iot devices," *Digital Investigation*, vol. 29, pp. S94–S103, 2019.

[22] EMC Fast Pass, "Tekbox near field probe set (tbps01)," https://www.emcfastpass.com/test-equipment/shop/near-field-probes/near-field-probe-set/, accessed On: 2024-06-14.

[23] V. Tirumaladass, S. Axelsson, M. Dougherty, M. A. Rasool, and M. H. Eldefrawy, "Deep learning-based electromagnetic side-channel analysis for the investigation of iot devices," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2020, pp. 150–156.

[24] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration*, vol. 40, no. 1, pp. 52–60, 2007.

[25] P. Robyns, M. Di Martino, D. Giese, W. Lamotte, P. Quax, and G. Noubir, "Practical operation extraction from electromagnetic leakage for side-channel analysis and reverse engineering," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 161–172.

[26] B. Hilburn, "Gnu radio - the free and open source radio ecosystem," https://www.gnuradio.org, accessed On: 2024-06-14.
[27] R. Callan, F. Behrang, A. Zajic, M. Prvulovic, and A. Orso, "Zero-overhead profiling via em emanations," in *Proceedings of the 25th international symposium on software testing and analysis*, 2016, pp. 401–412.
[28] A. P. Sayakkara, "Electromagnetic side-channel analysis methods for digital forensics on internet of things," Ph.D. dissertation, University College Dublin. School of Computer Science, 2020.
[29] A. Csete, "Gqrx sdr," https://gqrx.dk, accessed On: 2024-06-14.
[30] B. Cheng and D. M. Titterington, "Neural networks: A review from a statistical perspective," *Statistical science*, pp. 2–30, 1994.
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[32] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying lstm to time series predictable through time-window approaches," in *International conference on artificial neural networks*. Springer, 2001, pp. 669–676.
[33] HackRF, "Hackrf's documentation," https://hackrf.readthedocs.io/_/downloads/en/latest/pdf/, accessed On: 2023-10-28.