# Temperature Impact on Remote Power Side-Channel Attacks on Shared FPGAs

Ognjen Glamočanin, Hajira Bazaz, Mathias Payer and Mirjana Stojilović
EPFL, School of Computer and Communication Sciences, Lausanne, Switzerland

***Abstract*—To answer the growing demand for hardware acceleration, Amazon, Microsoft, and many other major cloud service providers have included field-programmable gate arrays (FPGAs) in their datacenters. However, researchers have shown that cloud FPGAs, when shared between multiple tenants, face the threat of remote power side-channel analysis (SCA) attacks. FPGA time-to-digital converter (TDC) sensors enable adversaries to sense voltage fluctuations and, in turn, break cryptographic implementations or extract confidential information with the help of machine learning (ML). The operating temperature of the TDC sensor affects the traces it acquires, but its impact on the success of remote power SCA attacks has largely been ignored in literature. This paper attempts to fill in this gap. We focus on two attack scenarios: correlation power analysis (CPA) and ML-based profiling attacks. We show that the temperature impacts the success of the remote power SCA attacks: with the ambient temperature increasing, the success rate of the CPA attack decreases. In-depth analysis reveals that TDC sensor measurements suffer from temperature-dependent effects, which, if ignored, can lead to misleading and overly optimistic results of ML-based profiling attacks. We evaluate and stress the importance of following power side-channel trace acquisition guidelines for minimizing the temperature effects and, consequently, obtaining a more realistic measure of success for remote ML-based profiling attacks.**

***Index Terms*—FPGA, multitenancy, machine learning, side-channel attacks, temperature**

## I. Introduction

The flexibility of field-programmable gate arrays (FPGAs), coupled with their highly-parallel architecture and energy efficiency, has led to the integration of FPGAs in various systems—from small embedded devices to datacenters and, recently, the public cloud. As a result, increased efforts are being made to enable secure virtualization and sharing of FPGA hardware acceleration fabric [1], [2]. At the same time, sharing FPGA resources implies many security issues, most of which are due to the electrical-level coupling via the shared power distribution network (PDN) or long wires [3]. However, finding a comprehensive solution to these issues remains an open research question [4].

Shared FPGAs are prone to remote power side-channel analysis (SCA) attacks [5], [6], as the low-level programmability of FPGAs allows implementing remotely-accessible voltage-fluctuation sensors directly in the FPGA fabric [7]. Attackers can leverage these sensors to break the secret keys of advanced encryption standard (AES) and Rivest-Shamir-Adleman (RSA) hardware accelerators, as well as software implementations of AES and RSA running on an ARM CPU in FPGA-based SoCs [5], [8]. Recent research leverages machine learning (ML) methods for profiling attacks, where ML models are trained on power traces of open-source designs (e.g., a neural network accelerator) likely to be deployed on the cloud and exposed to attackers. FPGA voltage sensors also enable recognizing and classifying FPGAs workloads [9] or recovering the neural network topology or hyperparameters [10]–[12].

In remote power SCA attacks, attackers implement on-chip sensors using FPGA fabric, which, in turn, is vulnerable to temperature-induced delay changes [13], [14]. In previous work, authors assumed negligible temperature changes during the operation of the victim circuit. However, in the case of ML-based profiling attacks in particular, trace acquisition can take a very long time (even days [15]); therefore, temperature variations unavoidably occur. Ignoring them, as we demonstrate in this paper, can lead to erroneous observations. For example, in the presence of temperature variations, ML models, instead of learning the target side-channel leakage, may inadvertently learn temperature effects that are otherwise not present in a real-life setup. Although remote power SCA attacks can succeed at different temperatures [10], [16], we find a better understanding of the temperature effects is necessary for future work.

Our key contributions are:

1) *Analysis of the temperature impact on sensor measurements.* We show, mathematically and experimentally, that sensor traces suffer from the same drifting offset seen in oscilloscope traces [15], and that the variance of trace samples is temperature dependent. These effects directly impact the measured side-channel leakage and are reflected in the success of a correlation power analysis (CPA) attack on an AES encryption module.
2) *Analysis of the temperature impact on the accuracy of ML-based profiling attacks.* When the power side-channel leakage is limited, and the trace acquisition takes a non-negligible time, we show that incautious trace acquisition can lead to ML models biased by temperature, resulting in misleadingly high accuracy. We analyze this unwanted effect and quantify the impact of correct trace acquisition techniques on accuracy.

In the remainder of the paper, we first give a background on time-to-digital converters (TDCs), the most commonly used FPGA voltage sensors (Section II). Next, Section III discusses the impact of temperature on the sensor output. Section IV describes our experimental evaluation methodology. In Section V, we present the results. Section VI discusses related work. Finally, Section VII concludes the paper.

This work is partially supported by the Swiss National Science Foundation (grant No. 182428).

978-3-9819263-7-8/DATE23/© 2023 EDAA

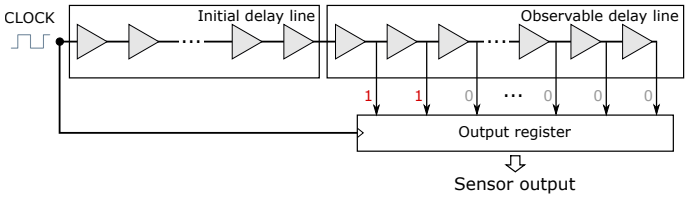


Fig. 1. Time-to-digital converter (TDC) sensor architecture.

## II. Time-to-Digital Converters as Voltage Sensors

TDCs and ring oscillators (ROs) are the most commonly used FPGA sensors in remote power SCA attacks. Neither TDCs nor ROs measure voltage fluctuations directly, as voltage measurement is only possible with dedicated FPGA system monitors with low sampling frequencies [17]. Instead, these sensors measure changes in the logic delay. Since the delay and voltage are (inversely) related, TDC and RO sensor readings can be used directly for the power SCA attacks. TDC sensors can capture voltage variations in time intervals as short as a few nanoseconds [7]. Compared to TDCs, RO-based sensors take up less FPGA resources, but they require a considerably longer time for a single measurement [18]. Therefore, when a high sampling rate is required, as is the case in remote power SCA attacks, TDCs are the adversary's preferred choice.

Fig. 1 shows a simplified view of a TDC sensor. This circuit estimates the supply voltage by measuring the propagation depth of a clock signal through a delay line [7]. The *initial* delay line, used for calibration, is typically implemented using FPGA logic with a higher propagation delay (e.g., LUTs, latches). The *observable* delay line, which measures the clock propagation depth, is implemented with fast carry-chain primitives for a high measurement resolution. The outputs of the observable delay line connect to a dedicated *output* register, sampled using the sensor clock. As the goal is to measure the propagation depth of the sensor clock through the delay line, proper calibration is necessary to ensure correct functionality. This procedure involves configuring the number of initial delay line elements so that the clock edge lands in the output register in every sample of the power trace.

## III. Temperature Impact on TDC Sensors

The cell delay impacts circuit performance and limits the maximum operating frequency. In a simple circuit delay model, e.g., the alpha-power law [19], the cell delay is inversely proportional to the drain current $I_d$ of a CMOS transistor. $I_d$ can be expressed as

$$I_d \propto \mu_e(T)(V_{dd} - V_{th}(T))^{\alpha}, \tag{1}$$

where $\mu_e$ represents the mobility, $V_{dd}$ the supply voltage, $V_{th}$ the threshold voltage, $\alpha$ a small positive constant, and $T$ the temperature [14]. The threshold voltage and mobility decrease with the rise of temperature, leading to two opposite effects on the drain current: decreased mobility reduces the drain current, while lower threshold voltage increases it. At high voltages, the mobility dominates Equation (1), resulting in a delay increase with the temperature. In contrast, at lower voltages, $V_{th}$ becomes the dominant factor, resulting in a delay decrease at higher temperatures. The voltage at which the temperature dependence inverts is called the *crossover voltage*, and it depends on the fabrication technology. This inverse temperature dependence (ITD) phenomenon was thoroughly studied across different fabrication technologies [14], [20].

Let us now formalize the temperature impact on the TDC sensor. As previously explained, the delay of a logic circuit $d(T, V)$ is indirectly proportional to the drain current $I_d$. Depending on the technology and the voltage $V$, $I_d$ can be directly or inversely proportional to the temperature. In addition to the carrier mobility and threshold voltage, $I_d$ depends on the Johnson-Nyquist thermal noise, which is constant across the spectrum and increases with temperature [21]–[23]. Moreover, the sub-threshold leakage current in lower technology nodes represents a source of noise that increases with the temperature [24]. Therefore, we can formalize the delay of a circuit under constant voltage as

$$d(T + \Delta T) \propto d(T) + \Delta d(\Delta T) + \delta(\Delta T), \tag{2}$$

where $\Delta T$ is the temperature change, $\Delta d$ is the change in delay, and $\delta$ is the thermal noise. When $\Delta T$ is positive, and the carrier mobility dominates Equation (1), $\Delta d(\Delta T)$ decreases. Otherwise, $\Delta d(\Delta T)$ increases with the temperature when the threshold voltage dominates.

The TDC sensor measures the number of delay elements through which an input clock has propagated during one sampling period $t_{sample}$. The relationship between the sampling period and the sensor output can be represented as $t_{sample} = O(T)d(T)$, where $d(T)$ is the delay of one delay element and $O(T)$ is the sensor output, i.e., the number of delay elements the input clock has traversed. When the temperature changes by $\Delta T$, the sensor output becomes

$$O(T + \Delta T) = \frac{t_{sample}}{d(T + \Delta T)} = \frac{O(T)d(T)}{d(T) + \Delta d(\Delta T) + \delta(\Delta T)}. \tag{3}$$

Therefore, when the delay increases with the temperature, the clock propagates through fewer elements in the delay line, resulting in lower sensor output. Otherwise, in the ITD case, temperature increase results in higher sensor output.

From (3), we find the expressions for the trace DC offset $\mu$ (i.e., the mean of all the samples in a trace) and the variance $\sigma^2$ (i.e., the dispersion of the values in a sensor trace):

$$\mu = \frac{1}{N}\sum_{i}^{N} O_i(T + \Delta T) \sim \frac{1}{\Delta d(\Delta T) + \delta(\Delta T)}, \tag{4}$$

$$\sigma^2 = \frac{1}{N}\sum_{i}^{N} (O_i(T + \Delta T) - \mu)^2 \sim \frac{1}{\Delta d^2(\Delta T) + \delta^2(\Delta T)}. \tag{5}$$

Here, $N$ is the number of sensor samples per trace. From (4) and (5), we can conclude that the trace DC offset is inversely proportional to the logic delay, while the variance is inversely proportional to the delay squared.

## IV. Evaluating The Impact of Temperature

In the context of remote power SCA attacks, we evaluate the impact of temperature on the sensor leakage and the ML-based power side-channel attacks.

### A. Leakage Analysis

In our first experiment, we evaluate how the sensor trace statistics change in the function of the temperature. In a thermal chamber, we start with a constant 40°C, and while recording AES encryptions, we increase the temperature in steps of 5°C up to 60°C. We measure the DC offset and variance of the sensor traces, two critical statistical parameters for SCA attacks.

Sudden ambient temperature variations—and their potential impact on the DC offset and variance of the sensor traces—could cause degradation in the Pearson correlation coefficient in the CPA attack, resulting in a higher number of traces to break the secret key. Therefore, in our second experiment, using the key rank (KR) estimation metric [25], we evaluate how transient temperature changes impact the success of the CPA attack against an AES hardware module. If the impact is significant, the temperature could severely interfere with conclusions between two different experiment runs (e.g., comparing the side-channel security of two cryptographic implementations).

Finally, we analyze the difference in side-channel leakage for traces recorded at different stable temperatures. In a thermal chamber with stable operating temperatures above 35°C, we record ten runs at 40°C, 45°C, 50°C, 55°C, and 60°C. For each temperature, we compute the average number of traces needed to break the key using the CPA attack and the KR estimation metric [25]. Significantly varying leakage at different external temperatures indicates a potential problem with lengthy experiments: traces acquired over a long time may result in skewed ML models, which are either degraded by the thermal noise or learn the temperature patterns instead of the actual leakage.

### B. ML Accuracy Evaluation

To evaluate the influence of the temperature and the trace acquisition method on ML classification problems, we devise three attack scenarios, i.e., victim workloads, each with a different classification complexity:

- *Hardware workload classification.* The victim contains several hardware modules, with only one running at a time. We choose four encryption cores: AES, PRESENT, KLEIN, and CRYPTON. All implementations are open source and available in the SCABox repository [26]. Using these cores, the attacker can train an ML model to identify the currently running hardware operation. This classification problem is considered easy [9], as the power consumption traces of entirely different hardware cores usually contain particular identifiers.
- *Soft-core CPU workload classification.* The victim is an open-source soft-core RISC-V CPU executing eight code snippets on random data. Each code snippet is intensive in one of the RV32I ISA instruction types: `load`, `store`, `branch`, `arith`, `compare`, `shift`, `logic`, and `jump`. The attacker, having access to the same CPU design and code, profiles the code snippets on many executions with random data inputs and trains a model to identify the one the victim is running. Gobulukoglu et al. showed that distinguishing between different soft-core CPU workloads is a difficult classification problem, and achieved an average classification accuracy of $\sim$ 50% [9].
- *Soft-core CPU instruction subset classification.* Here, the attacker is trying to identify instructions from the subset of the RV32I instruction set. The attacker trains on 10k instruction templates where the target instruction has randomized operands and data, and is surrounded by `nop` instructions. The templated instructions are `jal`, `add`, `xor`, `sll`, `lw`, `sw`, `bne` (not taken), `bne` (taken), and `slt`. Because individual CPU instructions have a short execution time, if the sensor has the same sampling frequency as the CPU, the leakage is limited, and the classification problem is considered hard.

To evaluate the temperature impact on the ML classification accuracy at room temperature, we use two trace acquisition methods for each workload, one incorrect and one recommended for power side-channel evaluation [25]:

- *Consecutive acquisition, room temperature (CR).* In this method—contrary to the recommended trace acquisition guidelines [25]—the traces of each ML class are acquired separately, by first recording all traces of class 1, then class 2, etc. When there is a large number of traces per class, and the trace acquisition takes hours, each class (i.e., specific workload) can be considered as recorded at a distinct temperature.
- *Interleaved acquisition, room temperature (IR).* In this recommended trace acquisition method—commonly used in power side-channel evaluation methods such as the *t*-test [25]—the traces of each ML class are acquired in an interleaved fashion, by recording a single trace of each class, in a randomized order, before continuing the acquisition of the next group of power traces. For many traces per class, interleaving the traces ensures equal temperature effects across all classes.

To evaluate model robustness and simulate exaggerated temperature changes during trace acquisition, we record two trace sets for hardware workload classification in a thermal chamber:

- *Consecutive acquisition, thermal chamber (CT).* The traces of each ML class are acquired separately. However, to exaggerate temperature variations, each class is recorded at a different but stable temperature: PRESENT at 38°C, AES at 43°C, KLEIN at 48°C, and CRYPTON at 53°C.
- *Interleaved acquisition, thermal chamber (IT).* The trace acquisition is interleaved, spreading the significant temperature changes across all classes. There are four sets of traces recorded at different stable temperatures: 38°C, 43°C, 48°C, and 53°C.

For classification, we implement five ML models commonly used in previous work: convolutional neural network (CNN1 and CNN2, a large and a small model), multilayer perceptron (MLP), long short-term memory (LSTM), and random forest classifier (RFC) [9]–[11]. Table I lists their architectural details. When training, we set the batch size to 64 and use the Adam optimizer while monitoring the loss to adapt the learning rate. We train on 90% of the dataset and use the remaining 10% for testing. The test/train split is performed randomly and in a stratified fashion. We train for 50 and 100 epochs for the hardware and software workload classification, respectively.

TABLE I
ARCHITECTURE DETAILS OF THE ML MODELS.

| Model | Architecture |
|---|---|
| LSTM | LSTM(100 units) + Dropout(0.2) + Dense(100 units, ReLU) + Dense(Softmax) |
| CNN1 | Conv1D(X filters, kernel size of Y) + MaxPool(2) \| (X,Y) = ((32, 12), (45, 10), (64, 8), (128, 4)) + Dropout(0.2) + Dense(100 units, ReLU) + Dense(Softmax) |
| CNN2 | Conv1D(64 filters, kernel size of 10) + MaxPool(2) + Conv1D(64 filters, kernel size of 4) + MaxPool(2) + Dropout(0.2) + Dense(100 units, ReLU) + Dense(Softmax) |
| RFC | number of estimators = 100 |
| MLP | Dense(X units, ReLU) \| X = (250, 350, 150, 50) + Dropout(0.2) + Dense(100, ReLU) + Dense(Softmax) |

## V. RESULTS AND DISCUSSION

### A. Leakage Analysis

Following the methodology in Section IV, we first evaluate the temperature impact on the sensor traces and the success of the CPA attack. We use a Digilent Basys3 (AMD Artix-7 XC7A36T FPGA): a cost-efficient FPGA platform suitable for potentially damaging thermal chamber experiments. With a single 128-bit TDC sensor (observable line with 128 elements) operating at 200 MHz, we record the power traces of an open-source AES-128 core clocked at 50 MHz [27]. To facilitate comparison between the experiments, we always use the same encryption key and the same set of plaintexts, and keep the sensor calibration constant.

In our first experiment, we record 900k AES traces in the thermal chamber, increasing the temperature over time: from 40°C to 60°C in steps of 5°C. Fig. 2 shows the trace DC offset, variance, and on-chip temperature in function of the elapsed time, as represented by the index of the recorded trace. We can observe that both the DC offset and the variance increase with the temperature; hence, the temperature-delay dependence lies in the ITD domain, where the threshold voltage dominates Equation (1). This experiment shows that the temperature significantly impacts the TDC sensor output and should not be overlooked when recording traces using on-chip sensors.

Next, we investigate how sudden temperature changes during the trace collection impact the success of the CPA attack. Before the experiment, we place the device in a cool place. Then, we record two datasets: 70k traces at a low temperature and 70k traces where the device is returned to room temperature after 10k traces to warm up gradually. Fig. 3 shows the KR estimation when attacking the key using CPA, in the function of the number of traces used in the attack. The temperature change has a visible impact: the orange line stops following the gray one and stagnates instead of decreasing. Consequently, the KR estimation drops to zero later, and the number of traces required to break the key increases. The reason is clear: as

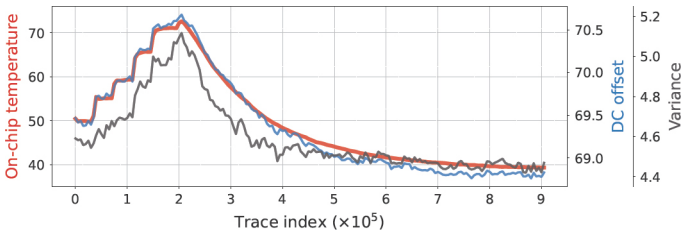


Fig. 2. The trace DC offset and variance at different on-chip temperatures, in the function of elapsed time, i.e., the trace acquisition index.

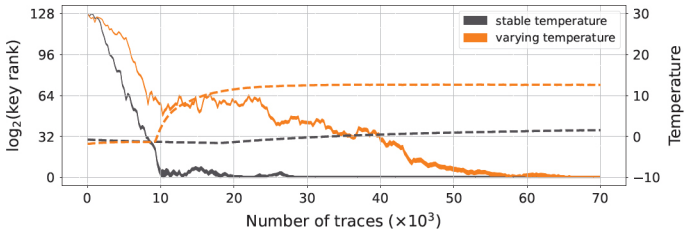


Fig. 3. Transient temperature impact on the key rank estimation using CPA.

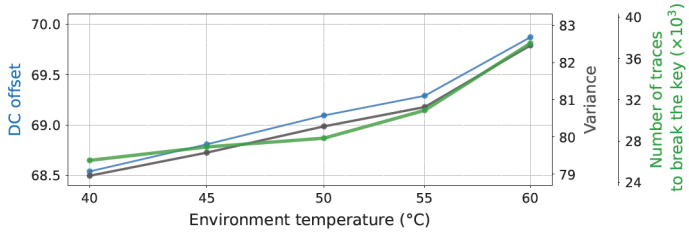


Fig. 4. Impact of the environment temperature on the sensor trace DC offset, variance, and the number of traces needed to break the key using CPA.

the CPA attack is performed using the Pearson correlation coefficient, any change in the trace DC offset and variance directly impacts the attack's success. This result shows that in security-sensitive experiments, such as comparing the side-channel security of cryptographic designs, it is important to consider environmental temperature changes and follow correct trace acquisition guidelines that minimize their impact [25].

Last, we examine the impact of stable temperature on the attack's success. For each temperature outlined in Section IV-A, we record ten experiment runs in the thermal chamber and compute the average trace DC offset, variance, and the number of traces required for a successful attack (when the KR estimation metric first drops to zero). Fig. 4 shows the results, averaged across ten runs. We can observe that more traces are required for a successful attack at higher temperatures and that thermal noise, more pronounced at higher temperatures, can increase the attack effort, resulting in approx. 1.4× more traces to break the key. Although the trace variance increases, the quality of sensor traces degrades at higher temperatures because the thermal noise becomes the dominant factor.

### B. ML Accuracy Evaluation

*1) Hardware workload classification:* For this experiment, we use SCABox [26], an open-source tool for side-channel evaluation on Digilent ZedBoard (AMD Zynq-7000 FPGA). We instantiate four cores working at 10 MHz: AES, PRESENT, KLEIN, and CRYPTON. As the AES is considerably larger

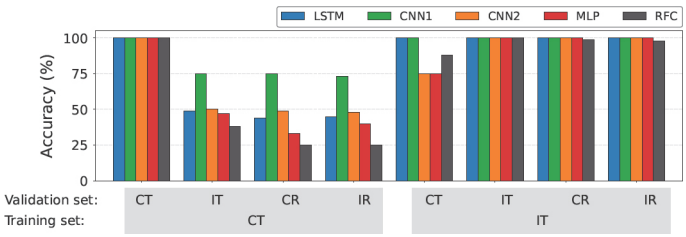


Fig. 5. Impact of the trace recording methodology on the ML model accuracy, in the case of hardware workload classification.

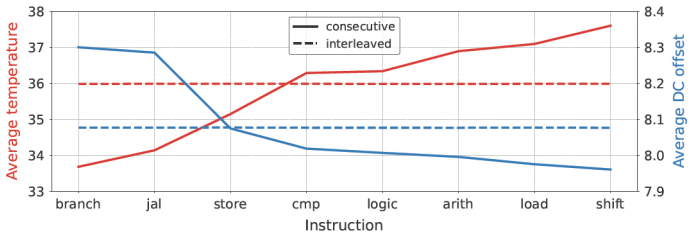


Fig. 6. Impact of the temperature on the average DC offset of each ML class, in the case of code snippet power side-channel traces.

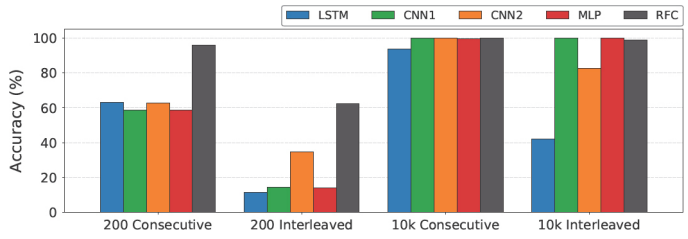


Fig. 7. Impact of the dataset size on the accuracy of the ML models, in the case of soft-core CPU workload classification.

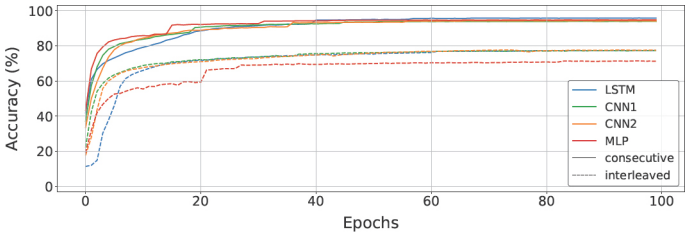


Fig. 8. Impact of trace acquisition on the evolution of the training accuracy, in the case of soft-core CPU instruction classification.

than other cores, we replicate the other cores eight times to obtain hardware workloads of approximately the same size and avoid classes with significantly different features. The SCABox instantiates eight TDC sensors, operating at 200 MHz.

For each core, we record 10k traces for IR, CR, IT, and CT datasets, and train five ML models (Section IV-B). In all cases, our models achieve 100% accuracy, showing that hardware workload classification is an easy problem and that the temperature does not impact the accuracy.

To evaluate the robustness of the trained models, we validate them using traces not seen during training and testing (the validation dataset size is 10% of the corresponding dataset). Fig. 5 shows that the models trained on the interleaved traces generalize well, and achieve high validation accuracy when tested on all the other datasets. As interleaved traces contain data samples from a wider range of temperatures, they help build more robust and generalized models.

*2) Software workload classification:* In this experiment, we use a high-end FPGA to evaluate the temperature impact on cloud FPGAs. On an AMD Alveo U200 datacenter accelerator card (UltraScale+ XCU200 FPGA), we place a PicoRV32 CPU [28] and 30 16-bit TDC sensors running at 320 MHz.

We start by reinvestigating the temperature impact on the DC offset of the sensor traces. We record 10k traces for each of the eight code snippets described in Section IV-B. To reduce noise, instead of recording one execution trace for the given code and data it operates on, we record and average 1k traces. Fig. 6 shows the average DC offset and the temperature of the traces of each class, for consecutive and interleaved trace acquisition. First, we can observe a direct temperature-delay dependence, because the sensor output drops as the temperature increases. Second, the DC offset of the traces recorded in the interleaved fashion does not correlate with the temperature, because the temperature variations impact all classes equally.

Next, we examine if the temperature impact on the classification accuracy changes with the dataset size (i.e., the difficulty of the classification problem). Using the acquired traces of the eight code snippets, we train the ML models twice: once with all 10k traces per code snippet and once with only 200 randomly selected traces per code snippet. We repeat the training with five random seeds and average the results, for more general conclusions. In addition to having randomized training parameters, the smaller dataset results in a unique random subset for each seed. The results in Fig. 7 show that, when using the entire dataset, both interleaved and consecutive datasets result in good accuracy (though lower for LSTM and CNN2 as they fail to converge for some seeds). However, training on incorrectly acquired traces results in considerably higher accuracy with the reduced dataset size, as the ML models learn the temperature effects instead of the leakage.

Let us now look at the training accuracy evolution. We record 10k traces for the nine instruction templates described in Section IV-B, interleaved and consecutive. Fig. 8 shows the results. Once again, we see that for complex classification problems (here, limited leakage of a single CPU instruction), incorrectly (i.e., consecutively) recorded traces mislead the ML models into learning the temperature effects instead of the actual leakage. In contrast, training on traces recorded in an interleaved fashion results in a lower, but more realistic classification accuracy.

Finally, we evaluate if common preprocessing techniques can alleviate the unwanted temperature effects from recorded traces. Table II shows the ML model accuracy when DC removal, filtering (high-pass with a 5 MHz cutoff), and normalization (MinMaxScaler) are applied. We see that neither of the three approaches significantly impacts the model's accuracy. Therefore, we can conclude that the temperature impacts not only the sensor measurements but possibly the power side-channel leakage generated by the victim.

TABLE II
ACCURACY OF ML MODELS WITH AND WITHOUT PREPROCESSING.

| Dataset | Preprocessing | Test accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | CNN1 | CNN2 | LSTM | MLP | RFC |
| Interleaved | None | 77.2 | 78.4 | 78.2 | 72.5 | 46.0 |
| Consecutive | None | 94.4 | 94.9 | 96.4 | 95.4 | 92.0 |
| Consecutive | DC removal | 94.5 | 94.6 | 92.4 | 94.7 | 81.0 |
| Consecutive | Filtering | 94.6 | 94.6 | 96.1 | 95.5 | 93.0 |
| Consecutive | Normalization | 98.4 | 98.2 | 98.0 | 98.3 | 92.0 |

Our results demonstrate that the impact of temperature on the TDC sensor measurements is important because it can lead to incorrect conclusions if trace acquisition guidelines are not followed. In the case of ML-based profiling attacks specifically, it can skew the accuracy and show better-than-expected results. Interleaving the trace recordings—the proper method of acquiring power traces—is necessary for spreading the temperature effects equally across the dataset.

## VI. RELATED WORK

Researchers often leverage ML models for remote power SCA attacks. Usually, attackers record one set of sensor traces, randomly splitting it into training (to profile the victim and train the ML model) and test traces (to evaluate the final accuracy). Gobulukoglu et al. used short-term Fourier transform and image classifiers to distinguish between cloud FPGA workloads [9], achieving high accuracy of 97.6%. They indicated that identifying soft-core CPU applications is challenging, resulting in comparatively low accuracy of approx. 50%. Zhang et al. showed that ML models could predict hyperparameters of a DNN accelerator with an accuracy of up to 100% [11]. In addition, Meyers et al. found that ML models can recover neural network folding [10].

To show model robustness, Meyers et al. [10] trained the model on traces recorded at room temperature and tested it on traces recorded at different ambient temperatures. The reported high accuracy (almost 100%) indicates that the traces contained substantial side-channel leakage, independent of temperature. We take a step further and show that when the leakage is limited (e.g., a small victim circuit), the temperature impact on the classification accuracy can be significant.

## VII. CONCLUSIONS

Varying environmental temperature impacts power side-channel traces recorded with TDC sensors. Our findings confirm that the temperature influences the sensor output and that this dependence varies across different FPGA families. We demonstrate that the temperature changes during trace acquisition impact the attack's success, as CPA requires more traces to break the AES encryption key if the temperature increases. Further, due to temperature, the trace acquisition method can significantly impact the robustness and the generality of models in ML-based profiling attacks. We demonstrate that for easily distinguished classes, (i.e., datasets with models converging to a 100% accuracy), trace acquisition has little to no impact on the final accuracy. However, for harder classification problems, ML models of incorrectly recorded traces learn temperature variations instead of leakage, resulting in misleadingly higher accuracy. Our research highlights the importance of adhering to appropriate trace acquisition guidelines, even in the context of shared FPGAs, if robust models and a realistic measure of classification accuracy are to be obtained.

## REFERENCES


[1] M. Asiatici, N. George, K. Vipin, S. A. Fahmy, and P. Ienne, "Virtualized execution runtime for FPGA accelerators in the cloud," *IEEE Access*, 2017.

[2] S. Byma, J. G. Steffan, H. Bannazadeh, A. L. Garcia, , and P. Chow, "FPGAs in the cloud: Booting virtualized hardware accelerators with OpenStack," in *FCCM*, 2014.

[3] S. S. Mirzargar and M. Stojilović, "Physical side-channel attacks and covert communication on FPGAs: A survey," in *FPL*, 2019.

[4] O. Glamočanin, D. G. Mahmoud, F. Regazzoni, and M. Stojilović, "Shared FPGAs and the Holy Grail: Protections against side-channel and fault attacks," in *DATE*, 2021.

[5] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," in *IEEE S&P*, 2018.

[6] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *DATE*, 2020.

[7] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs," in *FPGA*, 2013.

[8] J. Gravellier, J.-M. Dutertre, Y. Teglia, P. Loubet-Moundi, and F. Olivier, "Remote side-channel attacks on heterogeneous SoC," in *CARDIS*, 2019.

[9] M. Gobulukoglu, C. Drewes, W. Hunter, R. Kastner, and D. Richmond, "Classifying computations on multi-tenant FPGAs," in *DAC*, 2021.

[10] V. Meyers, D. R. Gnad, and M. Tahoori, "Reverse engineering neural network folding with remote FPGA power analysis," in *FCCM*, 2022.

[11] Y. Zhang, R. Yasaei, H. Chen, Z. Li, and M. A. A. Faruque, "Stealing neural network structure through remote FPGA side-channel analysis," in *FPGA*, 2021.

[12] S. Tian, S. Moini, A. Wolnikowski, D. Holcomb, R. Tessier, and J. Szefer, "Remote power attacks on the versatile tensor accelerator in multi-tenant FPGAs," in *FCCM*, 2021.

[13] D. R. Gnad, F. Oboril, S. Kiamehr, and M. B. Tahoori, "Analysis of transient voltage fluctuations in FPGAs," in *FPT*, 2016.

[14] A. Dasdan and I. Hom, "Handling inverted temperature dependence in static timing analysis," *TODAES*, 2006.

[15] A. Heuser, M. Kasper, W. Schindler, and S. Marc, "A new difference method for side-channel analysis with high-dimensional leakage models," in *CT-RSA*, 2012.

[16] B. Udugama, D. Jayasinghe, H. Saadat, A. Ignjatovic, and S. Parameswaran, "VITI: A tiny self-calibrating sensor for power-variation measurement in FPGAs," *TCHES*, 2021.

[17] *XADC User Guide UG480*, AMD Xilinx, 2023.

[18] S. Moini, A. Deric, X. Li, G. Provelengios, W. Burleson, R. Tessier, and D. Holcomb, "Voltage sensor implementations for remote power attacks on FPGAs," *TRETS*, 2022.

[19] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *JSSC*, 1990.

[20] T. Tsai, H.-C. Lin, and P.-W. Li, "Temperature-dependent narrow width effects of 28-nm CMOS transistors for cold electronics," *J-EDS*, 2022.

[21] S. Tedja, J. Van der Spiegel, and H. Williams, "Analytical and experimental studies of thermal noise in MOSFET's," *T-ED*, 1994.

[22] D. Triantis, A. Birbas, and D. Kondis, "Thermal noise modeling for short-channel MOSFETs," *T-ED*, 1996.

[23] J. Schurr, H. Moser, K. Pierz, G. Ramm, and B. P. Kibble, "Johnson-Nyquist noise of the quantized hall resistance," *IEEE I&M*, 2011.

[24] K. M. Zick and J. P. Hayes, "Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems," *TRETS*, 2012.

[25] K. Papagiannopoulos, O. Glamočanin, M. Azouaoui, D. Ros, F. Regazzoni, and M. Stojilović, "The side-channel metrics cheat sheet," *ACM Computing Surveys*, 2022.

[26] E. S. LAB, "SCABox," https://github.com/emse-sas-lab/SCAbox, Mines Saint-Etienne.

[27] *AES Encryption Core*, http://www.aoki.ecei.tohoku.ac.jp/crypto/, AIST and Tohoku University, 2019.

[28] *PicoRV CPU*, https://github.com/YosysHQ/picorv32, Yosys HQ, 2022.