Machine Learning on the Thermal Side-Channel: Analysis of Accelerator-Rich Architectures

David Werner* and Mark Hempstead†
Tufts Universty
Medford, MA 02155
Email: *david.werner@tufts.edu, †mark.hempstead@tufts.edu

Kyle Juretus‡ and Ioannis Savidis§
Drexel Universty
Philadelphia, PA 19104
Email: ‡kjj39@drexel.edu, §isavidis@coe.drexel.edu

***Abstract*—The thermal profiles of integrated circuits (ICs) have been leveraged as a side-channel in multiple circuit and architectural scenarios. Applications range from identifying hardware Trojans to estimating the per-core power consumption of homogeneous multicore processors. Such scenarios leverage the correlation between the on-chip location of the consumed power with some target information of interest, such as correlating the extra power consumption at a specific circuit position with the presence of a hardware Trojan. While the spatial correlation between the power consumption and thermal profiles applies to all ICs, there is a fundamental difference in the context of modern SoCs. The difference stems from the presence of hardware accelerators, in which localized power consumption corresponds to the system performing the specific task that a given accelerator executes.**

The work described in the paper demonstrates the implications of correlating the thermal and power profiles of SoCs by presenting two working case studies that determine, at runtime, 1) the activity factor of each accelerator and 2) whether or not a system is infected by malware. This work relies on pre-processing thermal images in order to obtain a spatial profile of the estimated power density and uses a modified version of a previously developed technique that is tailored for use with accelerator-rich ICs. The resulting power estimates are fed into machine learning models that predict the core activity factor with mean average errors between 3% and 5% for the highest performing core. The statistical models used for malware detection result in an AuROC score of up to 1.0 and 0.9 when the malware offsets the activity factor of a single core by 2.5% and the 3-sigma width of the workload activity factor distribution is 2.5% and 5%, respectively.

I. Introduction

Side-channels are unintended sources of information leakage. Commonly used side-channels include electromagnetic radiation (EM), timing, and power [1]–[4]. The information extracted from any given side-channel varies significantly depending on the scenario and can include information such as a private encryption key, the dynamic instruction trace of a program, or any other part of the state of the system that is not part of the ICs designed I/O interface.

The thermal side-channel is often dismissed due to the spatial and temporal low-pass filtering of information governed by the heat diffusion equations. EM is often seen as a superior replacement for the thermal side-channel, but is prone to environmental noise and other techniques that mask the signal including both passive and active shielding [5].

In spite of the challenges of extracting information from the thermal channel, researchers have characterized the leaked information from a variety of ICs. One study was able to extract the Hamming weight of repeatedly written data from a microcontroller [6]. Other studies have focused on conventional multicore processors, leveraging architectural properties such as concurrency to enable the extraction or communication of useful information. As a covert channel, arbitrary information is sent between colluding cores on the same IC [7], [8]. As a side-channel, hardware Trojans have been detected [9] and per-core power consumption has been estimated [10], [11].

In contrast to such studies, the work presented in this paper analyzes the effect of modern architectures on the thermal side-channel. Concerns regarding high power density have led companies like Intel, Apple, Qualcomm, and NVIDIA to incorporate specialized logic into the IC in the form of accelerators [12]. In contrast to general-purpose multicore processors, the accelerator-rich architectures are heterogeneous, composed of general-purpose cores and fixed-function accelerators that perform tasks such as video encoding/decoding, encryption, or digital signal processing (DSP) [13].

The work described in this paper aims to analyze how existing challenges in the field of security and side-channel analysis are affected by the integration of hardware accelerators. The following contributions are presented: (1) a thermal modeling framework for accelerator-rich architectures, (2) a technique to estimate the power-consumption profile of accelerator-rich ICs, (3) a machine learning model using a Deep Neural Network (DNN) that predicts the activity factor of each core in an IC based on the thermal side-channel, and (4) a methodology to produce statistical models that can detect malware using either the activity factor estimates from the DNN or by using the estimated power-consumption profile directly.

The rest of the paper is organized as follows: An overview of the modeling infrastructure used in this work is provided in Section II, which motivates the pre-processing step to estimate power-profiles from thermal image data presented in Section III. Two seperate case-studies are presented in Sections IV and V, and closing remarks are provided in Section VI.

II. Overview

The objective of this work is to extract system information by observing the thermal profile of an accelerator-rich IC. An

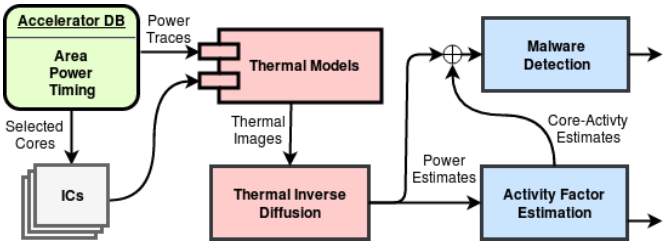


Fig. 1. Overview of the methodology used to model and implement the thermal side-channel.

overview of the methodology presented in this paper is shown in Fig. 1. For the sake of brevity, the non-memory components of a system—fixed function accelerators and general-purpose cores—are collectively referred to as *cores* for the remainder of this paper. Additionally, the term *activity factor* refers to the ratio of clock cycles that the core is actively processing data from the total number of completed clock cycles.

A. Thermal Image Pre-processing

As previously mentioned, heat transfer through thermal diffusion is a low-pass filter both spatially and temporally [14]. The filtering in time is due to the combination of the thermal conductance and thermal capacitance of the IC and the surrounding materials. The spatial filter is due to the non-zero lateral thermal conductance of the IC. The result is a loss of the high-frequency information of the power-consumption across the IC. Specifically, the heat generated at a discrete location in the circuit leads to increased temperatures across the rest of the die.

Initial experiments that directly analyzed the thermal images produced poor results. A trained Convolutional Neural Network (CNN) was only able to predict whether a core was 100% or 0% active with an average accuracy of less than 60%, with many core types resulting in an accuracy of around 50%. This motivated pre-processing of the thermal images to account for spatial low-pass filtering due to thermal diffusion. Therefore, in this work, all thermal images are first pre-processed by solving the inverse of the heat transfer equations, as shown in Fig. 1. The method used for pre-processing is described in detail in Section III.

B. System Types

In order to evaluate the methods presented in this paper, a variety of floorplans were generated with random assortments of cores. The goal is to demonstrate that the developed methods are not restricted to a given type of system or floorplan. When designing an IC, care is taken to avoid thermal hotspots and an unevenly distributed power density, which causes timing errors and accelerates device wearout [15], [16].

To account for thermal hotspots, the power-density of the core was factored in when selecting cores for a given IC. Each core was first labeled as either high-power-density (*high-pd*) or low-power-density (*low-pd*) using $10\frac{W}{cm^2}$ as a threshold when actively consuming power. Then, three types of systems were developed; systems using only *high-pd* cores, systems using only *low-pd* cores, and systems using any core regardless of power-density (*any-pd*). The three proposed configurations model a variety of systems ranging from low-power SoCs to high power server processors.

TABLE I
CORE POWER, AREA, AND TIMING CHARACTERISTICS USING SAED32-RVT AT $1.16V$, $25°C$. THE RATIO OF ACTIVE TO IDLE POWER IS REFERRED TO AS POWER RATIO. SHADED ROWS INDICATE *high-pd* CORES AND UNSHADED ROWS ARE *low-pd* CORES. ALL CORES WERE ACQUIRED FROM OPENCORES [17], EXCEPT FFT128, WHICH IS FROM THE SPIRAL FFT GENERATOR [18]

Core	Power-Density ($\mathbf{W/cm^2}$)			Area (μm^2)	Frequency
	Active	Idle	Ratio		
aes-128	28.40	2.04	13.94	17430	300MHz
aes-192	27.73	1.91	14.53	24064	300MHz
ECG_add	3.90	1.88	2.08	230605	100MHz
ECG_mult	2.97	1.84	1.62	229642	70MHz
fft128	22.62	1.51	14.94	1459551	100MHz
hpdmc	48.42	13.07	3.70	4248	700MHz
jpegencode	2.03	1.68	1.21	1186501	20MHz
neo430	2.44	1.32	1.84	370573	50MHz
RS_dec	47.47	1.73	27.40	115013	185MHz
wf3d	4.62	3.71	1.25	40700	150MHz

C. Modeling and Simulation Methodology

This work uses an end-to-end methodology developed to accurately model and characterize the thermal side-channel leakage of accelerator-rich ICs. Each core is modeled at the RTL level and synthesized with *Synopsys Design Compiler* using the *SAED32* standard cell library. Power consumption is determined using *Synopsys Primetime* with detailed activity traces obtained from simulating a testbench for each core using multiple test vectors. The characteristics of each of the cores are shown in Table I when using the SAED32-rvt standard cell library operating at $1.16V$ and $25°C$.

Floorplanning is completed using *HotFloorplan* and all thermal simulations are done using a slightly modified version of *HotSpot 6.0* [19]. The modifications include adding access to the temperature grids of all layers of the die as well as enabling the modeling of bare-silicon ICs, such as those that use wafer level chip scale packaging (WLCSP) [20], [21].

III. THERMAL INVERSE DIFFUSION

Solving the thermal inverse diffusion problem is a vital pre-prossing step that occurs prior to the training and evaluation of the models as shown in Fig. 1. The solution to the problem provides an estimate of the power-density profile of an IC based on a thermal image. The method to estimate the power-density profile that produced a given thermal map is similar to that of previous work [9]. The following section provides an overview of the analytical equations used in the model and the modifications required to adapt the model to accelerator-rich architectures.

A. Base Model

Thermal simulators such as HotSpot apply a resistor network that functions as a discretized version of the heat equation equation given by:

$$\mathbf{Rp} + \mathbf{e} = \Delta\mathbf{t}. \tag{1}$$

In (1), the matrix $\mathbf{R}$ is the resistive network that represents the thermal resistances of the system, $\mathbf{p}$ is the 2-D array of power densities, $\mathbf{e}$ is all sources of error in the system, and $\Delta\mathbf{t}$ is a 2-D thermal image normalized to the ambient temperature. If $\mathbf{R}$ and $\mathbf{p}$ are known, $\Delta\mathbf{t}$ is computed by performing a matrix multiplication after assuming $\mathbf{e}$ is $\vec{0}$. The inverse problem—computing $\mathbf{p}$ given $\Delta\mathbf{t}$—is more difficult and requires that $\mathbf{R}$ is either known or can be estimated.

Estimation of $\mathbf{R}$ is possible either experimentally or through simulation. For this work, $\mathbf{R}$ is derived by simulation following a methodology similar to that found in [9]. Deriving $\mathbf{R}$ in this way leverages the linearity of the model and is achieved by first partitioning the IC into a grid of $n_1 \times n_2$ blocks. Next, each block is activated one at a time by setting the power within the target block to some constant value and the power in the remaining blocks to 0. The thermal images produced by the impulse responses are $m_1 \times m_2$ thermal pixels and are used as the elements of $\mathbf{R}$. The values in $\mathbf{p}$ correspond to the power consumption within each block. In the case where the IC is split into $n_1 \times n_2$ blocks and each thermal image is $m_1 \times m_2$ pixels, the dimensions of $\mathbf{R}$ are $n_1 \times n_2 \times m_1 \times m_2$. In this work, the grid and thermal image dimensions are chosen as: n_1 and n_2 are 128 blocks and m_1 and m_2 are 32 pixels.

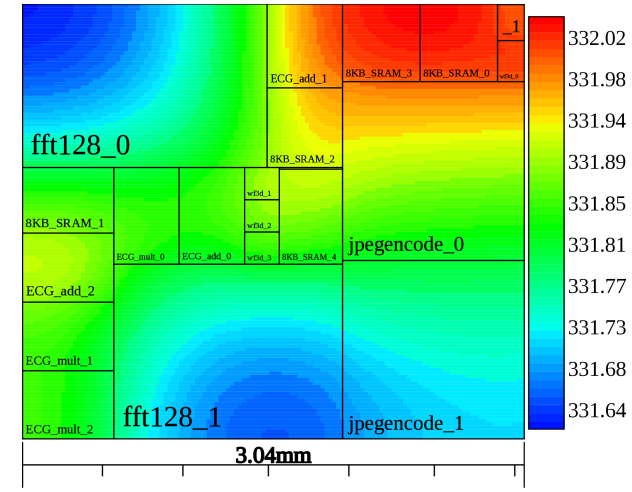
Given $\mathbf{t}$ and $\mathbf{R}$, $\mathbf{p}$ is estimated by solving the optimization problem that minimizes the error term $\mathbf{e}$ and is given by

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} ||\mathbf{Rp} - \Delta\mathbf{t}||_2^2. \tag{2}$$

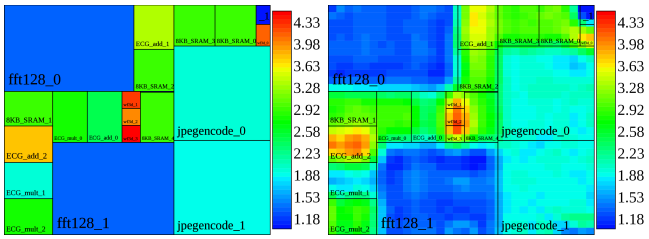
B. Regularization

While (2) can be solved without further modification, the fact that $\mathbf{R}$ is an ill-conditioned matrix means that small errors in $\mathbf{t}$ result in large errors in $\hat{\mathbf{p}}$ [22]. A method to minimize the error in problems that are ill-conditioned is through the addition of a regularization term, which serves many purposes, including but not limited to reducing noise, preventing overfitting, encouraging model sparsity, or leveraging assumptions that are made regarding the nature of the solution [23]. Regularization is essential in this work as the performance of the models is directly affected by the quality of the power-density estimates. This work proposes and demonstrates the advantages of a novel regularization term that simultaneously reduces noise and improves the accuracy of the estimated power map $\hat{\mathbf{p}}$.

The regularization term was developed based on the observation that the power-density of ICs with multiple discrete hardware accelerators tend to be piecewise-constant, which is due to each accelerator being physically disjoint and functionally independent from the others. Therefore, at any given time, any subset of the accelerators may be active (within thermal limits and other system limitations). The described characteristics tend to result in systems that have instantaneous power-densities that vary at the granularity of accelerators, with an overall power-density profile that is the weighted sum



(a) Thermal Image (output of HotSpot)



(b) Input Power Densities (c) Estimated Power Densities

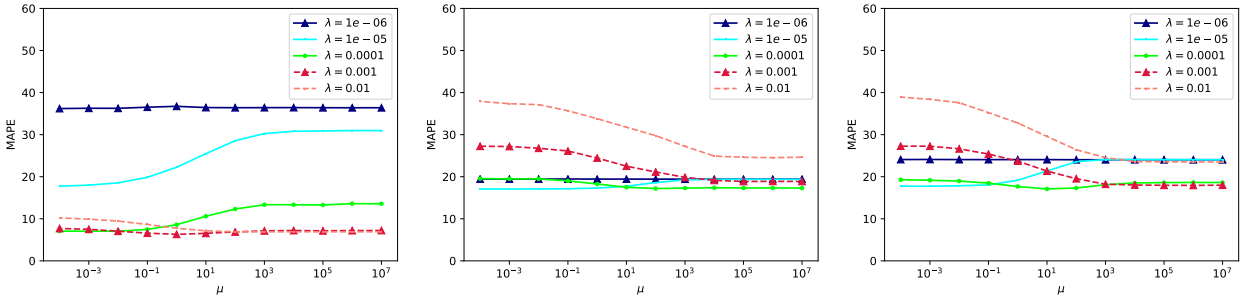
Fig. 2. An example circuit topology along with thermal simulation results corresponding to the thermal inverse solution using the modified cost function in (3). The optimization of the cost function successfully compensated for the low-pass filtering of the heat diffusion equation that occurred between Fig. 2b and 2a. Temperatures are displayed in K and power densities are in $\frac{Watts}{cm^2}$.

of the power consumption of each accelerator. A regularization term is added that simultaneously reduces the effects of random noise and accounts for the piecewise-constant nature of the IC power profile. The minimization problem using the modified cost function is given by

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} ||\mathbf{Rp} - \Delta\mathbf{t}||_2^2 + \lambda||\nabla\mathbf{p}||_1. \tag{3}$$

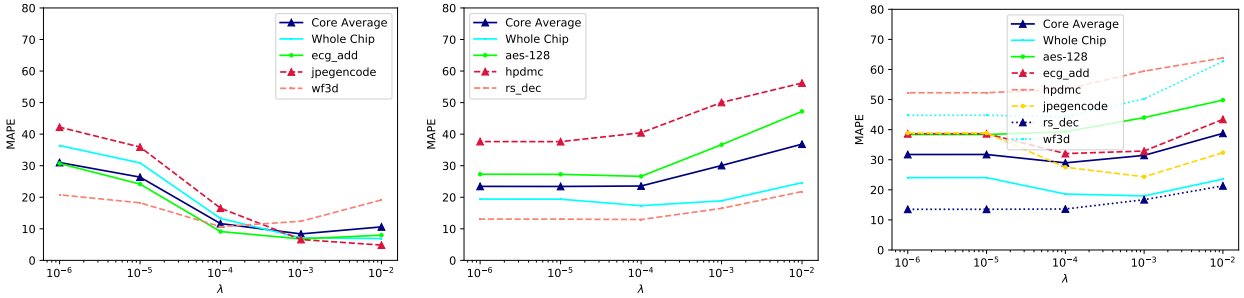
In (3), $\nabla\mathbf{p}$ is the gradient of $\mathbf{p}$. Since $\mathbf{p}$ is a 2-D grid of values, $\nabla\mathbf{p}$ is defined as the sum of the gradients in the x and y directions. Using the definition of $\nabla\mathbf{p}$, $||\nabla\mathbf{p}||_1$ is the sum of the absolute values of the gradients of $\mathbf{p}$. The λ is the relative weight of the penalty that is tuned to produce the most desirable results in each scenario. The effect of the penalty term is that adjacent blocks with different power-densities are penalized, which reduces noise and encourages piecewise-constant values in $\mathbf{p}$, both of which are well suited for accelerator-rich ICs. The process of determining the optimal value of λ is described in Section III-C.

To perform the minimization, prior work that solves a mathematically similar problem from the field of magnetic resonance imaging is applied [23]. The derivative of the penalty term is approximated using the following smoothing



(a) MAPE: *low-pd* systems (b) MAPE: *high-pd* systems (c) MAPE: *any-pd* systems

Fig. 3. Error of power-desnity estimate $\hat{\mathbf{p}}$ for the entire IC on systems with varied power-density (pd) as a function of μ and λ.



(a) MAPE: *low-pd* systems (b) MAPE: *high-pd* systems (c) MAPE: *any-pd* systems

Fig. 4. Error of power-density estimate $\hat{\mathbf{p}}$ for the entire IC and for individual cores as a function of λ with μ of $100,000$.

equation for each value $p_i \in \mathbf{p}$,

$$\nabla\lambda|p_i| = \frac{\lambda p_i}{\sqrt{p_i^2 + \mu}}, \tag{4}$$

where μ is a smoothing parameter. In effect, μ sets a *soft threshold* for values in $\nabla\mathbf{p}$. When a value in $\nabla\mathbf{p}$ is below the threshold, the penalty term is dominated by the contribution of μ, which implies that changing p_i has a small effect on the value of the cost function. Conversely, when p_i is greater than the soft threshold, changes in p_i result in significant changes to the value of the cost function. Given that $\mathbf{p}$ is measured in $\frac{Watts}{cm^2}$ and that $\nabla\mathbf{p}$ is the difference in $\mathbf{p}$ between adjacent blocks, μ indirectly sets a limit for tolerable differences in power density between adjacent blocks.

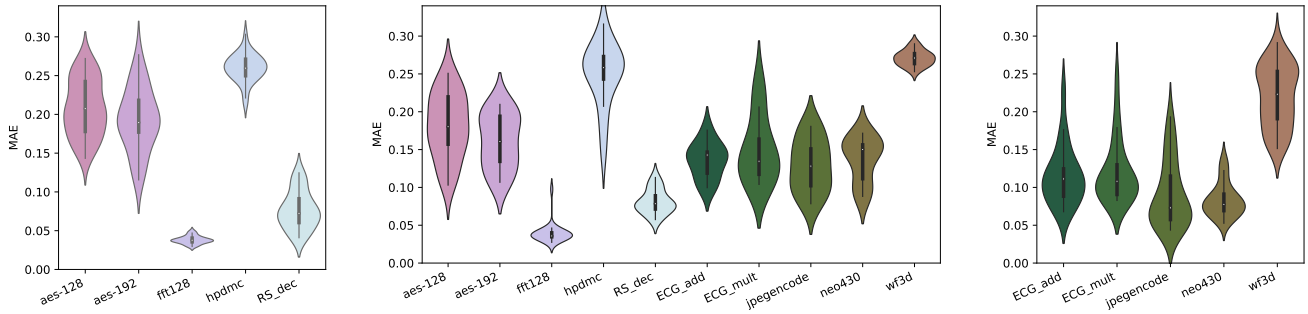
The result of using the modified cost function given by (3) is shown in Fig. 2, where μ is set to 10^5 and λ is set to 10^{-4}. As the figure indicates, the low-pass filtering of the heat diffusion equation is compensated for by the optimization of the cost function. The power-density estimate is noticeably less accurate for smaller cores such as *wf3d*. The lower accuracy is due to 1) the inability to compensate for *all* of the filtering caused by the process of heat diffusion and 2) the edges of the cores do not perfectly align with the grid of $\hat{\mathbf{p}}$. Both affect smaller cores more than larger cores.

C. Choosing optimal values for μ and λ

Choosing values for the smoothing threshold μ and the weight of the penalty term λ is dependant on a variety of factors including, 1) the size and power density of the IC, 2) the size of each core, and 3) the target use of the power estimates. In some cases, such as evalutating the overall power profile of an IC, μ and λ are chosen such that the error of the entire $\mathbf{p}$ array is minimized. In other cases, such as isolating the power consumption of a specific core, optimization of the error of the $\mathbf{p}$ values that correspond to the location of the specific core is preferred.

This section explores the optimization of μ and λ by performing a parameter sweep and evaluating the quality of the power estimates in $\mathbf{p}$. Each set of parameter settings are evaluated on over 100 randomly selected core configurations. For each simulation, a discretized version of $\mathbf{p}$ is compared with $\hat{\mathbf{p}}$ using Mean Absolute Percent Error (MAPE). The metric is computed for the entire power map as well as on a per core basis, where only the pixels strictly inside a given core are considered.

1) Effect on MAPE of the entire IC: The effects of μ and λ on the overall MAPE of $\hat{\mathbf{p}}$ are shown in Fig. 3. Each series within a given graph represents a different value for λ. For low-pd systems (Fig. 3a) and when λ is less than or equal to 10^{-5}, the regularization term is not weighted heavily and, therefore, the desired piecewise-constant behavior is not achieved. When λ equals 0.01, the smoothing effect dominates the cost function, resulting in a $\hat{\mathbf{p}}$ that filters useful information instead of removing only noise as intended. Between the two edge cases, for a λ of 0.001 for example, the optimization



(a) *high-pd* systems (b) *any-pd* systems (c) *low-pd* systems

Fig. 5. Mean average error of the Deep Neural Networks that estimate the core activity factor within systems with different power-density (pd) constraints.

function performs well across all values of μ and produces the lowest overall error. Similar trends are observed from the results shown in Figs. 3b and 3c, where a choice for λ that is either too low or too high results in increased error.

2) Effect on MAPE of individual cores: The effect of λ on the MAPE of cores for *low-pd*, *high-pd*, and *any-pd* systems is shown in Fig. 4. The plots indicate that there is no single value of λ that works well for all cores in all types of systems. From the results shown in Fig. 4a, a $\lambda \in [10^{-3}, 10^{-2}]$ produces the lowest errors for almost all cores and the IC as a whole. The *wf3d* core is smaller, and therefore, more negatively affected by smoothing, which results in an increase in MAPE when λ is greater than 10^{-4}. Similar trends are observed for other small cores including *aes-128* and *hpdmc*.

This work aims to extract per-core information from the thermal side-channel by tuning μ and λ such that the per-core MAPE is minimized. As previously described, there is no single set of values for μ and λ that works optimally for all system and core types. Therefore, μ and λ are chosen such that the MAPE is minimized as much as possible across all cores. For the remainder of this work, μ is set to 10^5 and λ is set to 10^{-3}, 10^{-4}, and 10^{-3} for *low-pd*, *high-pd*, and *any-pd* systems, respectively.

IV. Core Activity Estimation

In this section, the thermal channel is used to determine the activity factor of each core. The approach is intentionally generic and applies to a variety of attack scenarios, including reverse engineering proprietary software, being used in a timing attack, identifying if a vulnerable core is in use, and being used as a covert-channel.

A. Attack Vector

For this case study, an IoT device is targeted and the attacker is attempting to determine the activity factor of one of more of the cores on the device. As with most IoT devices, the attacker does *not* have direct software access to the device, but does have physical access.

In order to develop learning models, the attacker will perform characterization on device(s) on which arbitrary code is executing that sets the desired activity-factor for each core. The training data consists of a series of thermal images that are labeled with the activity factor of the target core(s). The number of thermal images used to develop the models is limited to 75 since the images are manually collected by the attacker. Additional examples are generated for model evaluation only.

The attacker creates workload components that target each of the accelerator cores present in the system, which is accomplished by repeatedly executing calls to the API of the device that leverage accelerators, such as a call that encrypts data using AES-encryption. From these components, the attacker creates a workload that activates any combination of cores on the system, thereby generating a wide range of data that is used to develop thermal and power models.

B. Model Implementation

The machine learning models used in this case study are Deep Neural Networks (DNNs), which are created using the *Keras* [24] machine learning framework in *python*. As previously mentioned, Convolutional Neural Networks (CNNs) were also evaluated and produced poor results due to the tendency to generalize spatially.

The attacker must also construct $\mathbf{R}$ in order to estimate the power profile of the IC. It is possible to construct 'impulse responses' for each core by directly and individually activating each core and collecting the corresponding thermal profiles. This, however, may prove difficult for the attacker depending on the software interface to the cores as well as any shared hardware or interdependancies between cores. Therefore, for this work, $\mathbf{R}$ is constructed through simulation of impulse responses of each block within a grid, which requires only basic knowledge of the IC (surface area and thickness) and the general thermal properties of silicon.

C. Results

The performance of the DNN models is summarized through the results shown in Fig. 5. The prediction quality of each model is characterized using Mean Average Error (MAE). The data is categorized based on the type of system being evaluated and is aggregated across multiple floorplans for all

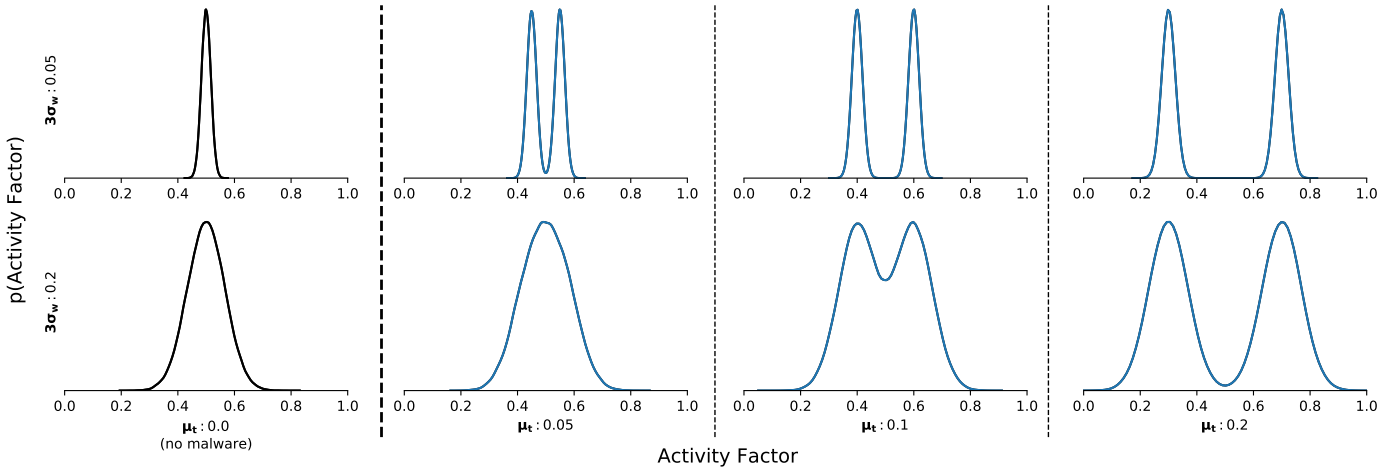


Fig. 6. Distributions of core activity factor with and without added malware. Each row includes a different workload standard deviation, indicated by the size of the three standard deviations, $3\sigma_w$. Each column specifies the mean activity factor, μ_t, of the malware that is added to or subtracted from the activity factors of a core. Workload mean is statically set to 50% for this example only, and is randomly chosen for each core in the case study.

of the threshold-voltages available in the SAED32 standard cell library.

The two major factors that contribute to the error in activity-factor prediction of any given core are the core size and power density, which are summarized in Table I. The following discussion will be limited to the results for *high-pd* (Fig. 5a) and *low-pd* (Fig. 5c) systems at first. These results are then compared with the results of *any-pd* systems in order to isolate the effects of varied power-density on model accuracy.

In general, models for cores that are either large or have a high active/idle power ratio have lower MAE. For example, *jpegencode* and *RS_dec* both have a MAE around 7% but have very different characteristics; the *jpegencode* is very large with an active/idle ratio of only 1.21, while *RS_dec* is 10× smaller but has a very high active/idle ratio of 27.4. The best performing models are for the *fft128* core, which is slightly larger than the *jpegencode* core but has a much greater active/idle ratio of 14.94.

Conversely, the models that perform poorly include those that are small and have active/idle ratios close to 1. For example, the *hpdmc* is approximately 300× smaller than the largest core (*fft128*), but has a MAE of 25% despite having an active/idle ratio of 3.7. While the *wf3d* core is approximately 10× larger than *hpdmc*, the active/idle ratio is only 1.25, which results in a high MAE of 23%.

Small cores tend to have lower accuracy for the same reason that they exhibit larger MAPE in $\hat{\mathbf{p}}$, namely that heat diffusion and regularization both filter high frequency information. Spatial low-pass filtering also contributes to the reduced accuracy of the activity factor estimates for cores with low active/idle power ratios; the temperature of the target core is influenced more by the higher power consumption of surrounding cores than any change in the activity factor. Such cores are also more sensitive to error in $\hat{\mathbf{p}}$ as the signal strength is weaker. If the error in $\hat{\mathbf{p}}$ is large relative to the change in $\mathbf{p}$ when the core transitions from an active to an idle state, the signal to noise ratio is lower, resulting in a decrease in accuracy of the model.

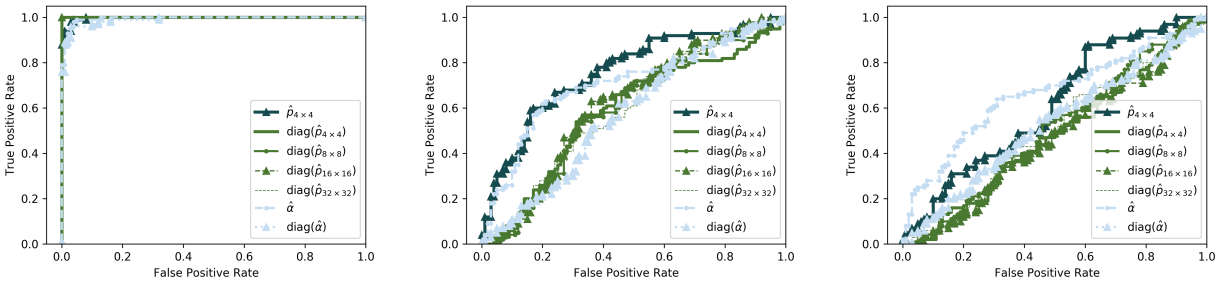
A secondary factor that affects the error in the predicted activity-factor of a core is the overall uniformity of the power-densities of the IC. The accuracy of the models for *any-pd* systems (Fig. 5b) is lower for the majority of cores included in *low-pd* and *high-pd* systems only; all of the *low-pd* cores include approximately 5% greater error and most of the *high-pd* cores perform similarly or slightly worse. The two exceptions are the *aes* cores, which perform slightly better in *any-pd* systems. The likely reason is the relatively small size of the *aes* cores. In the *high-pd* systems, the activity of the *aes* cores is obfuscated by the much larger and higher power *fft128*, whereas in the *any-pd* systems, the *aes* cores are surrounded by cores with much lower power consumption, making it easier to identify the thermal signature of *aes*.

V. Malware Detection

In this section, the thermal side-channel is used to detect malware. Traditional malware detection schemes operate on the same system that is being monitored, exposing the checker to the malicious software intended to be detected. Using the thermal side-channel for malware detection has the advantage of operating in a manner that is completely independent from the target system, removing any vulnerability.

A. Model Definitions

1) Workload: The case study is designed to model workloads that repeatedly execute on the target system. This is common in many IoT devices that continually process data, such as a security camera or a sensor node that locally performs data processing. Even with the given definition of a workload, the execution time of each core potentially varies due to a variety of reasons including scheduling by the OS, contention for shared resources such as caches, and execution of highly variable operations like network communications. Therefore, a workload is defined such that the activity factor



(a) Highest performing core: $fft128$ (b) Average performing core: *aes128* (c) Worst performing core: *hpdmc*

Fig. 7. Receiver Operating Characteristic (ROC) curves for select cores from a system with high-power density cores. The workload has a 3-sigma value of 20% and the malware mean is 10%. The $\hat{p}$ models are 32×32 before being scaled to the specified size. All $\hat{p}$ models use the sample covariance matrix, Σ_{MLE}, except those marked with *diag*, which use only the diagonal elements.

of each core is normally distributed around some nominal value. While the activity factors of cores in a real system are likely correlated due to contention for shared resources or inter-accelerator data dependencies, this work makes the conservative assumption that no such correlations exist, which makes anomaly detection more difficult.

The width of the distribution of the activity factor, defined as 3 standard deviations (3σ), also varies across systems. Realtime sensor systems tend to have less variation while more complex computing platforms like those found in servers operate with more variation. In order to model the entire range of possible systems, the 3σ value for each workload is varied over the range of 2.5% to 20%.

2) Malware: While malware takes countless forms, this work limits the scope to malicious activity that is repeatedly executed in order to mimic the behavior of a system that is unwittingly part of a bot-net or has been hijacked to steal compute resources to perform an undesired distributed task. Therefore, malware is also defined such that the effect on the activity factor of a single core is normally distributed around a nominal value given by μ_t. The magnitude of μ_t is varied between 2.5% and 40% to model a variety of possible malware types.

If malware is executing on a system, the impact is either to add to (increase the amount of work allotted to a given core) or subtract from (slow down a core that serves as a producer in a producer-consumer relationship) the activity factor of a core. In this work, malware is limited to affecting the activity factor of a single core within the system.

The combined effect of $3\sigma_w$ and μ_t on the distribution of the observed activity factor for a given core is shown in Fig. 6. In the absence of malware, the activity factor of a core is normally distributed. When malware is added, the distribution becomes multi-modal, with peaks above and below the original mean.

B. Anomaly Detection Model

One approach to performing anomaly detection is using parametric distributions [25]. In cases where each sample is labeled as either *normal* or *anomalous*, the labeling of a sample with an unknown class is accomplished by

$$\text{Label } normal \text{ if } f(\mathbf{x}|\Omega) > \tau \text{ else label } anomaly, \tag{5}$$

where f is the parametric model, Ω are the given parameters of the model, $\mathbf{x}$ is the sample being labeled, and τ is a threshold that is tuned to trade-off between the number of false positives and false negatives.

One commonly used parametric model is the multivariate Gaussian, which is given by

$$f(\mathbf{x}|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left[\tfrac{-1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right], \tag{6}$$

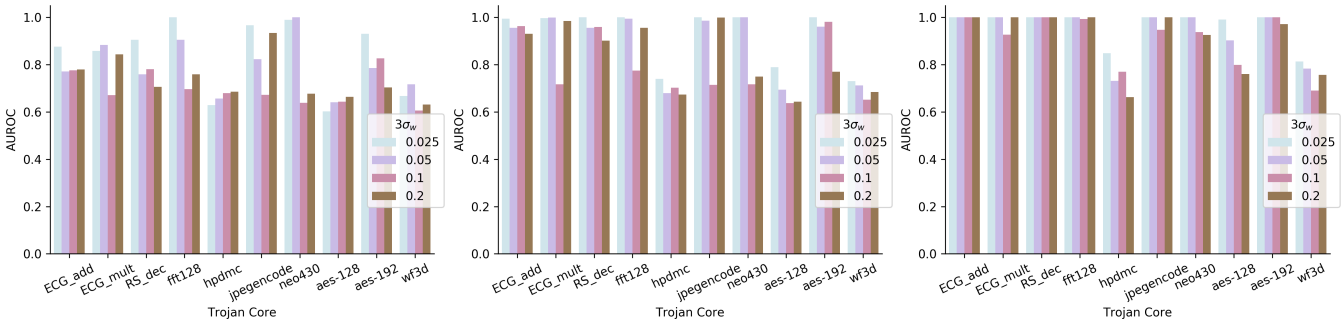
where $\mathbf{x}$ is the observation being considered, μ is a vector containing the mean values for each *feature* in $\mathbf{x}$, Σ is the covariance matrix of the *features* in $\mathbf{x}$, and d is the number of *features* in $\mathbf{x}$. The value commonly used for Σ is the *Maximum Likelihood Estimate* (MLE), which is given by

$$\Sigma_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_i)(x_i-\mu_i)^T. \tag{7}$$

While the MLE estimate works well in some cases, there are limitations. One such limitation is due to the fact that the *pdf*, given by $f(\mathbf{x}|\mu,\Sigma)$, contains Σ^{-1} in the definition, requiring Σ to be invertible. In order for Σ to be invertible, the number of samples N must be larger than the number of features n. While $N > n$ samples is sufficient to produce an invertible Σ, in practice, the number of *observations* must be much greater than the number of features (in the order of $N > 10n$) for the resulting estimate of Σ to be accurate [26]. In many situations, especially in fields such as IoT, it is impractical to have more *observations* than *features*. In such cases, other techniques must be applied to accurately estimate Σ. One simple method is to use only the diagonal elements of the covariance matrix

$$\hat{\Sigma} = diag(\Sigma_{MLE}), \tag{8}$$

which has the advantage of working with any number of samples so long as all of the variances are non-zero. However, the model no longer accounts for any correlations between



(a) 2.5% malware mean (b) 5% malware mean (c) 10% malware mean

Fig. 8. Area under ROC (AuROC) values for the best model for each core among any floorplan. The width of the distribution of the workload activity factor is indicated by $3\sigma_w$ for malware strengths of 2.5%, 5%, and 10%.

different *features*. While more complicated techniques exist [26], this work evaluates the use of Σ_{MLE} and $diag(\Sigma_{MLE})$.

C. Example ROC Curves for Malware Detection

The Receiver Operating Characteristic (ROC) curves for three selected cores in a *high-pd* system are shown in Fig. 7. The True Positive Rate (TPR) corresponds to the number of times malware is correctly detected and the False Positive Rate (FPR) corresponds to the number of times the model predicts that malware is present when it is not. The sub-figures include results that characterize all possible trade-offs between TPR and FPR made by adjusting the threshold τ in (5). Each series of each sub-figure represents a model that is constructed from a different set of features; the $\hat{\mathbf{p}}$ models are resized versions of the power-density estimates, and the *Activity Factor* models are generated using the predicted activity factor of all cores using the DNN models developed in Section IV. The models that only include the diagonal components of the covariance matrix are denoted as 'diag'; otherwise, the fully populated covariance matrix is used.

The ROC curves indicate large variation in results based on the type of core affected by malware. Similar to the results for activity factor estimation in Section IV, large cores with a high active/idle power ratio such as *fft128* (shown in Fig. 7a) produce the highest prediction accuracy, while small cores such as *hpdmc* (shown in Fig. 7c), exhibit lower prediction accuracy.

The model that produces the best ROC curve also varies from core to core. For cores like the *fft128* that more significantly impact the power consumption of the IC, $\hat{\mathbf{p}}$ produces accurate models even when down-sampled to a very low resolution of 4×4. In this case, using only the diagonal components of the covariance matrix degrades performance by filtering out important information regarding the state of each core. Conversely, smaller cores like *aes128* and *hpdmc* are generally not affected by the resolution of $\hat{\mathbf{p}}$, as most resolutions produced equally poor results. Instead, the biggest factor in determining the performance of the model is whether the dense covariance matrix is used or only the diagonal components are used. In both cases, lowering the resolution of $\hat{\mathbf{p}}$ results in the *loss* of information. However, a better model accuracy is possible as the noise and error in the estimate of Σ_{MLE} decreases due to having less degrees of freedom for the same number of samples.

For all three cores, the activity factor models perform comparably to that of the most optimal $\hat{\mathbf{p}}$ model. In addition, the performance of the activity factor model does not require changes to the resolution of $\hat{\mathbf{p}}$, whereas the $\hat{\mathbf{p}}$ models do. If a non-optimal resolution of $\hat{\mathbf{p}}$ is chosen, then the $\hat{\mathbf{p}}$ model performs poorly relative to the activity factor model. In the case of *hpdmc*, the activity factor model is the highest performing at low *FPR* and only slightly outperformed by a $\hat{\mathbf{p}}$ model at high *FPR*, where the model becomes unusable as it almost always predicts that malware is present even when it is not.

The activity factor models also exhibit decreased model performance when only the diagonal elements of the covariance matrix are used, which is counter-intuitive given that the actual activity factors of the cores in a system are *not* correlated, but demonstrates that activity factor estimates from the DNN models *are*. The correlation is due to the fact that $\hat{\mathbf{p}}$ errors in one location likely result in an opposite compensatory error at a nearby location, as dictated by the solution of the heat diffusion equation. When such an error lies near the boundary of two cores, the result is the under-estimation of the activity factor of one core and the over-estimation of the other.

D. AuROC Summary of Malware Detection

The Area Under ROC (AuROC) is a metric that characterizes the quality of the model that produced a given ROC curve. The metric is calculated by computing the area under the curve, which ranges from 0.0 to 1.0, with a value of 1.0 being ideal. The interdependence between workload variation ($3\sigma_w$) and the mean of the added malware distribution (μ_t) is shown in Fig. 8, where the included values are the best AuROC scores across all models for a given core. As expected, malware with a higher mean (Fig. 8c) are easier to detect for all cores relative to malware with a lower mean (Fig. 8a). Similarly, large values of $3\sigma_w$ also result in poor model performance.

More interesting trends are seen when comparing between cores. The cores that had the largest model errors in the

estimates of activity factor, such as *fft-128* and *wf3d*, only begin to accurately detect the presence of malware when the mean activity factor of the malware is 10%, even when the $3\sigma_w$ is small. In contrast, for the cores that had the lowest error in the estimate of activity factor, such as *RS_dec*, *neo430*, and *aes192*, malware that offsets the core activity factor by only 2.5% is still detected with a high degree of accuracy when the $3\sigma_w$ is small. As $3\sigma_w$ increases, the model is limited by the overlap between the activity factor distributions, as shown in Fig. 6. The trend poses a fundamental limit to the accuracy of any model, even if the activity factor of each core is known precisely. Therefore, the overall performance of the model is a combination of the accuracy of the estimated activity factor and the relative size of μ_t and $3\sigma_w$.

VI. Conclusions

The work presented in this paper evaluated a novel approach to solving the thermal inverse diffusion problem specifically in the context of accelerator-rich ICs. The technique was shown to increase the accuracy of power-density estimates and reduce noise. The analysis of the optimal values for the hyperparameters of the model, specifically μ and λ, demonstrated that there is not a single optimal set of parameters, but rather, there is a tradeoff between the accuracy of the estimates for different cores and the IC as a whole.

Given the power density estimates, models were constructed that extracted information from the thermal side-channel. A DNN was trained to predict the activity factor of each core without requiring any floorplan knowledge and achieved a Mean Average Error ranging from 3% to 5% for the highest performing core on a variety of system type. In addition, the DNNs were also evaluated on a variety of other cores, characterizing the factors that dictate the accuracy of the estimated core activity factor when using the thermal side-channel.

Lastly, a methodology for detecting malware through analysis of the thermal side-channel using a statistical model was described. The model was evaluated using the power-density estimates directly, as well as using the activity factor estimates produced by the developed DNN model. The effects of using diagonalization as a form of regularization were evaluated and shown to improve the AuROC score of the model at times, but not in all cases, motivating either model selection or the use of more advanced regularization techniques.

References

[1] A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "EDDIE: EM-based detection of deviations in program execution," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, June 2017, pp. 333–346. [Online]. Available: http://doi.acm.org/10.1145/3079856.3080223

[2] B. Mao, W. Hu, A. Althoff, J. Matai, J. Oberg, D. Mu, T. Sherwood, and R. Kastner, "Quantifying timing-based information flow in cryptographic hardware," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2015, pp. 552–559.

[3] A. Singh, M. Kar, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "Improved power side channel attack resistance of a 128-bit AES engine with random fast voltage dithering," in *Proceedings of the IEEE European Solid State Circuits Conference*, Sep. 2017, pp. 51–54.

[4] D. Strobel, F. Bache, D. Oswald, F. Schellenberg, and C. Paar, "SCANDALee: A Side-ChANnel-based DisAssembLer using local electromagnetic emanations," in *Proceedings of the Design, Automation Test in Europe Conference Exhibition*, Mar. 2015, pp. 139–144.

[5] F. Shahzad, M. Alhabeb, C. B. Hatter, B. Anasori, S. M. Hong, C. M. Koo, and Y. Gogotsi, "Electromagnetic interference shielding with 2d transition metal carbides (MXenes)," *Science*, vol. 353, no. 6304, pp. 1137–1140, Sep. 2016. [Online]. Available: http://science.sciencemag.org/content/353/6304/1137

[6] M. Hutter and J.-M. Schmidt, "The temperature side-channel and heating fault attacks," in *Proceedings of the Smart Card Research and Advanced Applications International Conference*, ser. Lecture Notes in Computer Science, 2013.

[7] R. J. Masti, D. Rai, A. Ranganathan, C. Mller, L. Thiele, and S. Capkun, "Thermal covert channels on multi-core platforms," in *Proceedings of the USENIX Security Symposium*, 2015, pp. 865–880. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/masti

[8] J. Brouchier, N. Dabbous, T. Kean, C. Marsh, and D. Naccache, "Thermocommunication," Tech. Rep. 002, 2009. [Online]. Available: http://eprint.iacr.org/2009/002

[9] K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware Trojan detection using multimodal characterization," in *Proceedings of the Design, Automation Test in Europe Conference*, Mar. 2013, pp. 1271–1276.

[10] S. Reda, K. Dev, and A. Belouchrani, "Blind Identification of Thermal Models and Power Sources From Thermal Measurements," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 680–691, Jan. 2018.

[11] S. Reda and A. Belouchrani, "Blind identification of power sources in processors," in *Proceedings of the Design, Automation Test in Europe Conference*, Mar. 2017, pp. 1739–1744.

[12] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proceedings of the Annual International Symposium on Computer Architecture*, Jun. 2011, pp. 365–376.

[13] "Intel Quark SoC X1010 (16k cache, 400 MHz) product specifications." [Online]. Available: http://ark.intel.com/products/80901

[14] R. Cochran, A. N. Nowroz, and S. Reda, "Post-silicon power characterization using thermal infrared emissions," in *Proceedings of the ACM/IEEE International Symposium on Low-Power Electronics and Design*, Aug. 2010, pp. 331–336.

[15] A. K. Coskun, T. Simunic Rosing, K. Mihic, G. De Micheli, and Y. Leblebici, "Analysis and Optimization of MPSoC Reliability," *Journal of Low Power Electronics*, vol. 2, no. 1, pp. 56–69, 2006.

[16] B. C. Schafer and T. Kim, "Hotspots elimination and temperature flattening in VLSI circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 11, pp. 1475–1487, Nov. 2008.

[17] "Opencores." [Online]. Available: http://opencores.org/

[18] "Spiral dft/fft ip core generator." [Online]. Available: http://www.spiral.net/hardware/dftgen.html

[19] "HotSpot 6.0 Temperature Modeling Tool." [Online]. Available: http://lava.cs.virginia.edu/HotSpot/documentation.htm

[20] "AVR211 : Wafer level chip scale packages - doc42007.pdf." [Online]. Available: http://www.atmel.com/Images/doc42007.pdf

[21] "Understanding Flip-Chip and Chip-Scale package technologies and their applications - application note - Maxim." [Online]. Available: https://www.maximintegrated.com/en/app-notes/index.mvp/id/4002

[22] R. Cochran, A. N. Nowroz, and S. Reda, "Post-silicon power characterization using thermal infrared emissions," in *Proceedings of the ACM/IEEE International Symposium on Low-Power Electronics and Design*, Aug. 2010, pp. 331–336.

[23] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/mrm.21391/abstract

[24] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[25] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[26] D. Nikovski and K. Byadarhaly, "Regularized covariance matrix estimation with high dimensional data for supervised anomaly detection problems," in *Proceedings of the 2016 International Joint Conference on Neural Networks*, Jul. 2016, pp. 2811–2818.