Post-Silicon Heat-Source Identification and Machine-Learning-Based Thermal Modeling Using Infrared Thermal Imaging

Sheriff Sadiqbatcha, *Graduate Student Member, IEEE*, Jinwei Zhang, *Student Member, IEEE*, Hengyang Zhao, *Student Member, IEEE*, Hussam Amrouch, *Member, IEEE*, Jörg Henkel, *Fellow, IEEE*, and Sheldon X.-D. Tan, *Senior Member, IEEE*

***Abstract*—In this article, we present a novel post-silicon approach to locating the dominant heat sources on commercial multicore processors using heatmaps measured via an infrared (IR) thermal imaging setup. To locate the heat sources, 2-D spatial Laplacian transformation is performed on the heatmaps followed by *K*-means clustering to find the dominant power/heat-source clusters. This is an exclusively post-silicon approach that does not require any knowledge of the underlying design of the commercial chips other than the information that is publicly available. Since the identified clusters are the thermally vulnerable areas on the die, we then propose a machine-learning-based framework to deriving a thermal model capable of estimating their temperatures during online use. Our approach involves collecting transient temperature data of the aforementioned heat sources and synchronized high-level performance metrics from the chip, and training a long-short-term-memory (LSTM) neural network (NN) that uses the performance metrics as inputs to estimate the temperatures of the identified heat sources in real time. Since the model is meant for real-time use, we explore methods of reducing the performance overhead and inference time of the model. This includes a novel power correlation-based approach to identifying the thermally irrelevant performance metrics and eliminating them in order to reduce the input dimensionality of the model, and an analysis on network sizing to determine the ideal NN configuration for the problem at hand. The model is trained and tested exclusively using measured thermal data from commercial multicore processors. The experimental results from two Intel multicore processors (i5-3337U and i7-8650U) show that the proposed approach achieves very high accuracy (root-mean-square error: 0.55 °C–0.93 °C) in estimating the temperatures of all the identified heat sources on the chip.**

***Index Terms*—Deep learning, infrared imaging, machine learning, microprocessors, temperature measurement, thermal model, thermal sensors.**

Manuscript received June 27, 2019; revised November 11, 2019 and March 4, 2020; accepted June 9, 2020. Date of publication July 7, 2020; date of current version March 19, 2021. This work was supported in part by NSF under Grant CCF-1741961, Grant CCF-2007135, and Grant OISE-1854276. This article was recommended by Associate Editor S. Reda. *(Corresponding author: Sheriff Sadiqbatcha.)*

Sheriff Sadiqbatcha, Jinwei Zhang, Hengyang Zhao, and Sheldon X.-D. Tan are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: ssadi003@ucr.edu).

Hussam Amrouch and Jörg Henkel are with the Department of Computer Science, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.

Digital Object Identifier 10.1109/TCAD.2020.3007541

I. Introduction

The temperature has a profound impact on all the major long-term reliability effects, such as electromigration (EM) for interconnects, and bias temperature instability (BTI) and hot carrier injection (HCI) for CMOS devices [1]–[3]. To enhance reliability, many system-level thermal/power regulation techniques, such as clock gating, power gating, dynamic voltage and frequency scaling (DVFS), and task migration have been proposed in the past [4]–[7]. One critical aspect of the aforementioned algorithms is correctly estimating the full-chip temperature profile to properly guide the online thermal management schemes [8], [9]. However, accurate thermal estimation is a difficult task, especially for commercial off-the-shelf multicore processors. Some of the existing methods depend on the on-chip temperature sensors. However, very few physical sensors are typically available, and they may not be located in close proximity to the true hotspots on the chip, consequently misleading the temperature regulation decision [10]. Hence, the more popular solution is to supplement the on-chip sensor readings with estimated temperatures of all the prominent heat sources or hotspots on the chip via thermal models based on estimated power traces. These methods offer higher spatial resolution as they allow for the temperature of all the heat sources on the chip to be monitored in the real time [11]–[13].

Existing approaches consist of several bottom-up numerical methods, such as hotspot [11]-based simplified finite difference methods, finite-element methods [14], equivalent thermal RC networks [15], and the recently proposed top-down behavioral thermal models based on the matrix pencil method [16] and the subspace identification method [17], [18]. However, the existing methods suffer from several drawbacks. First, most of the compact thermal models need accurate power traces as inputs; but estimating the power of each functional unit (FU) of a real microprocessor is not a trivial task, if not infeasible [19], [20]. On the other hand, from the system-level thermal or power management perspective, the parameters that can be easily accessed are the frequency, voltage, and many other performance metrics natively supported by most commercial processors. Thermal models which are functions of those parameters will be more desirable and practical. Second, calibration of the compact models against the actual chip

0278-0070 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

temperature under different workloads and thermal boundary conditions is very difficult. The reason being, measuring the temperature profile of a working chip that is under load without the heat sink is a difficult task. Finally, there is still a lack of an exclusively post-silicon approach to locating and estimating the temperatures of dominant heat sources on the chip. Such a method would enable the development and deployment of more robust thermal control schemes for current and older generations of processors.

Hence, in this work, we aim to address all the aforementioned issues. Our novel contributions are summarized as follows.

1) We establish a lucid infrared (IR) thermal imaging setup with an advanced thermoelectric-based rear-mounted cooling technique. This system allows us to obtain accurate online thermal maps of commercial multicore processors while they are under load.
2) We propose a novel post-silicon approach to locating the prominent heat sources on commercial microprocessors without any proprietary information about the chip's design. Our approach involves 2-D discrete cosine transformation (DCT) for noise reduction, and Laplacian transformation followed by K-means clustering for heat-source identification.
3) We propose the use of high-level performance metrics provided by tools such as Intel's Performance Counter Monitor (IPCM), instead of low-level performance counters, as the inputs to our thermal models as they provide a comprehensive view of the processor's utilization in real time. Moreover, they are easily accessible as most commercial processors are natively supported.
4) We apply long-short-term-memory (LSTM) networks to build the system-level hybrid thermal model that is capable of highly accurate online temperature estimation. The proposed model is parameterized with IPCM metrics, such as chip frequency, instruction counts, etc., and is trained and tested exclusively using thermal data measured directly from the processors under test.
5) Since the model is meant to be deployed for real-time use, we explore methods of reducing the performance overhead and inference time of the model. This includes a novel power correlation-based approach to identifying the thermally irrelevant IPCM metrics and eliminating them in order to reduce the input dimensionality of the model, and an analysis on network sizing to determine the ideal neural network (NN) configuration that offers sufficient tradeoff between accuracy and inference time.
6) We have structured the proposed framework such that it does not require any design changes and moreover does not need any information on the chip's architecture or floorplan. Hence, it can just as easily be applied by the original manufacturer as well as a third party for current and older generations of commercial processors.

The experimental results from two Intel multicore processors (i5-3337U and i7-8650U) show that the proposed thermal model achieves very high accuracy (RMSE: 0.55 °C–0.93 °C) in estimating the temperatures of all the identified heat sources on the chip. For i5-3337U, the maximum root-mean-square error (RMSE) is 0.76 °C or 1.79%, and for i7-8650U, the maximum RMSE is 0.93 °C or 1.35%.

The remainder of this article is organized as follows. Section II reviews some of the existing relevant works. Section III introduces the proposed algorithm flow and describes the IR setup that will be used for data acquisition. Section IV presents the new post-silicon heat-source identification method. Section V presents the runtime data that will be used to train the machine learning-based model. Sections VI and VII describe methods of reducing the computational overhead of the model by means of input dimensionality reduction and network sizing, respectively. Section VIII presents the implementation of the proposed approach on two commercial multicore processors and compares the accuracy of the model with measured thermal data. Section IX concludes this article.

II. Review of Relevant Work

Performance counter-based power-consumption estimation methods for both high performance and mobile/embedded processors have been developed in the past [21]–[23]. These methods offer a software-based solution to runtime fine-grain power estimation rather than requiring component-wise power sensors which incur significant design overheads and are prone to sensing and process-based noise similar to embedded temperature sensors [24]. Additionally, performance counters along with the temperature readings from the embedded temperature sensors have also been used to predict the future readings from the embedded sensors [24]–[26]. However, as previously mentioned, the number of embedded sensors on the chip is very limited due to their high area and power overheads and they may not always be placed in close proximity to the hotspots on the chip.

To supplement the temperature readings from the embedded sensors, it is imperative to develop thermal models that can either estimate the temperature profile of the entire chip, or all the thermally vulnerable areas on the chip. To this end, interpolation-based methods have been proposed to compute the full-chip thermal map from the sensor readings [27]. Since the number of sensors and their placement have a significant impact on the accuracy of the aforementioned interpolation, smart sensor placement algorithms have been proposed that can be used during design time to find the optimal placement for the given budget of embedded thermal sensors [28]–[30]. It has been shown that adapting the aforementioned sensor placement algorithms significantly improves the accuracy of soft sensing or interpolation-based methods that can be used to estimate the temperature of any arbitrary location on the die including the hotspots. However, these methods are not suitable for chips that are not designed with the aforementioned smart sensor placement algorithm. There is still a lack of an exclusively post-silicon approach that requires no changes to the design of the chip.

Hence, in this work, we propose a novel machine-learning-based framework to post-silicon temperature estimation for commercial multicore processors using high-level performance metrics. Here, the correlation between the utilization behavior of the processor shown by high-level performance monitors

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

and the temperature response of the chip is automatically learned. With data being of utmost importance with any machine learning-based approach, in this work, we present a thorough and systematic method to measure first-hand thermal and utilization data from commercial microprocessors. The overarching goal of this work is to propose an exclusively post-silicon method to identify all the thermally vulnerable areas of the die and build a thermal model that can be used to estimate the temperatures of these areas during runtime.

III. Proposed Thermal Modeling Framework

A. New Thermal Modeling and Characterization Overview

The proposed thermal modeling approach involves several critical steps. First and foremost, it requires an advanced IR thermography setup that is capable of recording lucid thermal maps of the processor under test while it is executing real workloads. This setup will be discussed in detail in the next section. The heatmaps acquired using this system will then be used to objectively determine the locations of prominent heat sources (or power sources) on the commercial processor. The located heat sources will then be clustered together into dominant heat-source clusters, which are the thermally vulnerable spatial locations on the chip (hotspots). Our novel approach to locating these dominant heat sources will be discussed in Section IV. Once the heat sources are located, the IR setup will once again be used to record time-series temperature data of all the identified heat sources while the processor is subjected to a variety of realistic workloads. At the same time, a suite of high-level performance metrics will be recorded in synchronous with the capture rate of the IR camera. After sufficient data are acquired, a variant of recurrent neural networks (RNNs) called the LSTM network will be employed to train the thermal model. Once trained, the thermal model will be able to use the performance metrics as inputs to estimate the temperatures of all the identified heat sources in real time.

B. Our IR Thermography Setup With Rear-Mounted Cooling

One important aspect of the proposed approach is the acquisition of spatial and temporal heatmaps from the processor under test while it is running a practical load. First-hand acquisition of thermal data has been shown to be superior, as opposed to using simulators or other golden models [31], [32] for this task. To this end, we have built a specialized IR thermography setup inspired by the recently proposed RAMA thermal imaging system [10]. This state-of-the-art setup features a thermoelectric-based rear-mounted heat extractor which offers a precisely controllable cooling solution where the processor is cooled from underneath, leaving the front side completely exposed to the IR camera. This introduces minimum interference with the IR emissions from the chip which is in stark contrast with the existing cool-liquid or oil-based front-cooling techniques where some sort of compensation or de-embedding is needed [20]. This allows us to capture lucid thermal images of the chip, while maintaining a safe operating temperature comparable to the traditional front-mounted heat sink-based cooling solutions.

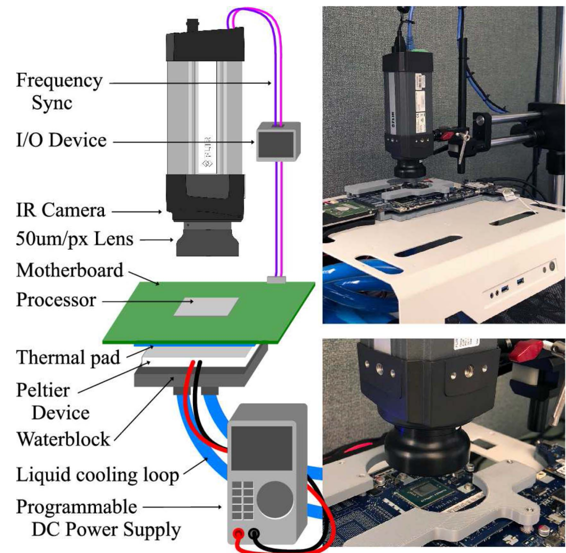


Fig. 1. Illustration of our IR thermography setup.

The IR setup that we have built (Fig. 1) has a slightly different configuration than what was presented in [10], as it is adapted specifically for the application presented in this study. The new setup consists of an FLIR A325sc IR camera with an image resolution of 16-bit 320×240 pixels (px) and operating frequency of 60 Hz [33]. It can measure the temperature range from 0 °C to 328 °C. Its spectral range (observable wavelength of electromagnetic radiation) is from 7.5 to 13 μm. It has a microscope lens attachment which provides a spatial resolution of 50 μm/px. The IR camera has an internal waveform generator that generates a square waveform in synchronous with the capture rate of the camera. An I/O device is used to interface the waveform generator to the processor under test, so that the performance metrics (recorded on the processor) can be synchronized with the thermal data captured by the camera. The processors under test are the Intel i5-3337U (2 cores, 4 threads, released in 2012) and the Intel i7-8650U (4 cores, 8 threads, released in 2017). Mounted on the PCB directly underneath the processor is the thermoelectric-based cooling system which includes a Peltier device powered by a programmable dc power source. A liquid cooling system is used to cool the hot side of the Peltier device.

While the aforementioned configuration of the RAMA system was used in this study, few key caveats must be considered while building such a setup [10]. First, the Peltier device used in our setup is the TEC1-12710 which has a maximum power rating of 110 W [34]. It should be noted that the cooling potential of the Peltier device must be significantly larger than the thermal design power (TDP) of the processors under test. This is due to the fact that the rear-mounted cooler has to overcome the low-thermal conductivity of the PCB which is now in-between the processor and the cooler. We found that the Peltier device we used was able to easily match the cooling potential of the stock heat sink of the two processors we tested. However, more powerful Peltier devices may be required for higher end processors. If the Peltier device is not able to match the stock cooling system of the chip, but

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

is able to maintain the chip's temperature under its thermal limits, then a method such as the one presented in [20] can be used to scale the captured thermal maps to the amplitude that would have been observed under the stock cooling system. Second, IR camera systems are factory calibrated on materials with a high emissivity coefficient (ϵ). If the heat spreader covering the processor's die has a low ϵ, then the temperature readings from the camera will not be accurate. One way to address this issue is to recalibrate the camera to the given heat spreader using readings from the processor's integrated thermal sensors. However, this method is not recommended as the internal sensors have an accuracy of ±5 °C [35]. Another method, which we prefer, is to improve ϵ of the heat spreader by covering it with a thin layer of a better emitter. As suggested by Amrouch and Henkel [10], one simple option is masking tape ($\epsilon \approx 0.92$).

IV. Heat-Source Identification

One important aspect of building a thermal model for a processor is identifying the dominant heat/power-source clusters. These are the critical areas of the chip for many online or dynamic thermal/power management schemes. Locating the heat-source clusters or hotspots during design time is trivial as it can be done through power/thermal simulation tools. However, post-silicon identification of these locations with no knowledge of the chip's proprietary design is not trivial. As a result, locating these heat-source clusters without the floorplan and layout information becomes an important problem. In this section, we will present our novel approach to locating these heat sources on commercial processors exclusively using measured thermal data.

A. Laplacian Operation for Heat-Source Identification

We start with the general thermal diffusion equation shown in [36]

$$\rho C_p \frac{\partial T}{\partial t} - \nabla(\kappa \nabla T) = g \quad (1)$$

where T is the temperature (K), ρ is the mass density of the material (kg·m^{-3}), C_p is the mass heat capacity (J·kg^{-1}·K^{-1}), κ is the thermal conductivity, and g is the spatial heat energy generation (W · m^{-3}).

Since, in this step, we are only concerned with the spatial information that we can extract from a single time step, we can ignore the transient terms in (1). We then get the 2-D steady-state thermal equation (assuming homogeneous material with location independent κ)

$$-\kappa \nabla^2 T(x, y) = g_T(x, y) \quad (2)$$

where ∇^2 is the Laplace operator. From the simplified heat equation (2), we can see that the negative spatial Laplacian of the temperature distribution across the die is equal to the spatial heat generation. Therefore, we can perform the 2-D spatial Laplacian on a given thermal map to locate the underlining heat sources $g_T(x, y)$. The exact amplitude of $g_T(x, y)$ remains unknown since we do not know the value of κ; however, this is not important since we are only interested in the spatial locations of the heat sources, hence relative amplitude will suffice. This method also works even if there is a thin heat-spreader layer with a conductive surface (e.g., a die with heat spreader and package). This is due to the fact that although the heat spreader will distribute the heat across its surface and dissipate it, the spacial locations of the underlying heat sources do not change.

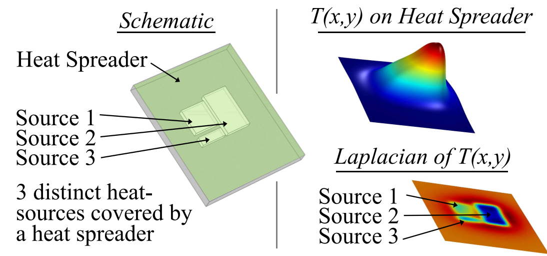


Fig. 2. COMSOL validation of the heat-source identification method.

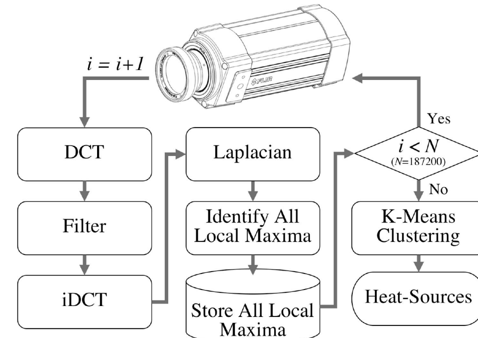


Fig. 3. Illustration of our novel heat-source identification flow.

To illustrate this idea, we simulate a simple structure in COMSOL multiphysics where three distinct heat sources are placed underneath a thin heat spreader with a conductive boundary in-between. The simulation results (Fig. 2) show that by applying 2-D Laplacian transformation on the temperature distribution, $T(x, y)$, observed on the heat spreader, the three distinct heat sources located underneath the heat spreader can be easily identified.

B. Full Heat-Source Identification Flow

In this section, based on the aforementioned principles, we present our approach to identifying the major heat sources using measured thermal maps captured from the commercial processor under test. First, the raw thermal map is preprocessed to remove the inherent noise present in measured IR data. After this, 2-D spatial Laplacian is applied to locate the active heat sources in 2-D space. This process is repeated on the ten-of-thousands of heatmaps captured at different time instances under different workloads. Finally, a K-means-based clustering algorithm is invoked to find the dominant heat-source clusters with high densities of heat sources. The proposed method is illustrated in Fig. 3. For clarity, we will demonstrate the algorithm using the following example.

1) Preprocessing for Noise Reduction via DCT: We start with a thermal map (i.e., Fig. 4) captured from the dual-core Intel i5-3337U processor under test.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

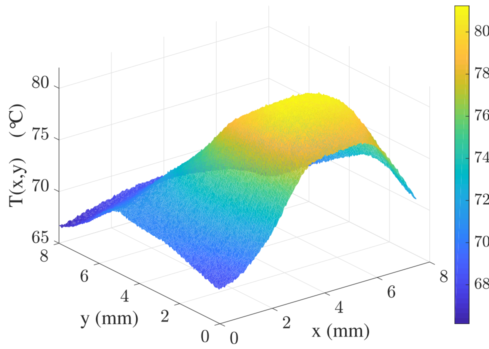


Fig. 4. Heatmap of the Intel i5-3337U captured using our IR system.

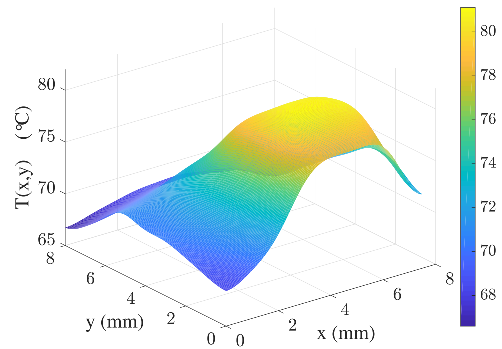


Fig. 5. Noise-reduced heatmap of the Intel i5-3337U.

The raw thermal map may contain noise, which must be removed as a preprocessing step. This step is crucial because the 2-D discrete Laplacian

$$\nabla^2 f(x,y) = f(x-1,y) + f(x+1,y) + f(x,y-1) + f(x,y+1) - 4f(x,y) \quad (3)$$

is very sensitive to local difference of adjacent pixels.[1] 2-D DCT filter is an effective method for eliminating high-frequency noise, by transforming the heatmap into spatial frequency domain, masking the high-frequency components, and then transforming back to the original space domain. A 2-D DCT consists of two separate 1-D DCT operations, which can be denoted as

$$f_k = \frac{a_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{i=1}^{N-1} a_i \cos\frac{(2i+1)k\pi}{2N}, \; 0 \le k < N \quad (4)$$

where vector $\{a_i\}$ is the original data, and $\{f_k\}$ is the result of 1-D DCT. A 2-D DCT is completed by applying 1-D DCT on each column and then on each row of the matrix. With the heatmap $T(x, y)$ transformed to 2-D frequency domain $F(x, y)$, a filtered frequency map $\mathcal{F}(x, y)$ can be obtained by applying a mask

$$\mathcal{F}(x,y) = F(x,y)m(x,y) \quad (5)$$

where $m(x, y)$ is the mask map valued 0 at high frequencies and 1 at low frequencies. The filtered heatmap $\mathcal{T}(x, y)$ is then obtained by taking the inverse 2-D DCT on the filtered frequency map $\mathcal{F}(x, y)$. Similar to its forward counterpart, the inverse 2-D DCT consists of two separate inverse 1-D DCT steps on the rows and columns, respectively. The inverse 1-D transformation of (4) is

$$a_i = \frac{f_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} f_k \cos\frac{(2i+1)k\pi}{2N}, 0 \le i < N. \quad (6)$$

This operation performed on the noisy heatmap previously shown in Fig. 4 results in the filtered heatmap shown in Fig. 5.

[1] For a heatmap with 177×166 px, with temperature ranging from 65 °C to 80 °C, the Laplacian range is approximately at ± 0.025 °C/px^2. While with noise introduced, the Laplacian can easily go up to ± 1.0 °C/px^2, which is much higher than the useful Laplacian component.

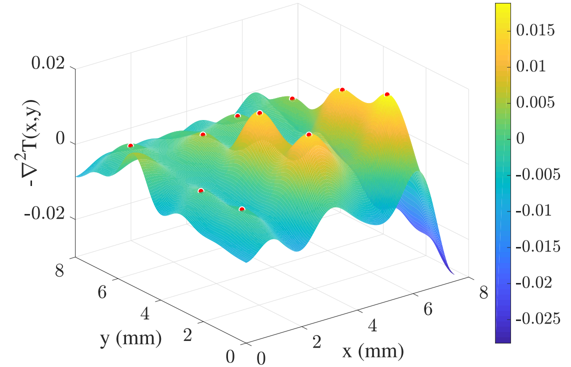


Fig. 6. Negative Laplacian of the heatmap with all the heat sources (red) identified.

2) Temperature Laplacian for Heat-Source Identification: The Laplacian operation in (2) can now be applied to the noise-less heatmap, which reveals the locations of the internal heat sources that were active during the time this particular heatmap was captured. These heat sources can be identified by locating all the local maxima on the negative Laplacian of the temperature distribution as shown in Fig. 6.

3) K-Means Clustering for Dominant Heat-Source Localization: While the above step can be used to identify the heat sources that were active during the time the heatmap shown in Fig. 4 was recorded, there is no guarantee that all the prominent heat sources within the chip were active during that time. In fact, many of the heat sources are disabled at any given time due to the extreme power and clock gating used in modern processors. In order to ensure that most, if not all, of the prominent heat sources on the chip are identified, we repeat the aforementioned heat-source identification process on many ($N \sim 2 \times 10^5$) heatmaps that were collected while the processor is subjected to a multitude of different workloads with varying execution patterns. This process increases the chances of activating all the prominent heat sources on the chip, at least once, so that their thermal signature can be recorded. The aggregate of the local maxima identified using this method is shown as clusters of red dots in Fig. 7.

This method results in dense clusters of heat sources. However, it is not possible to track the temperature of each point in the cluster. Instead, we use the K-means clustering algorithm (using the "elbow criterion" to determine the

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

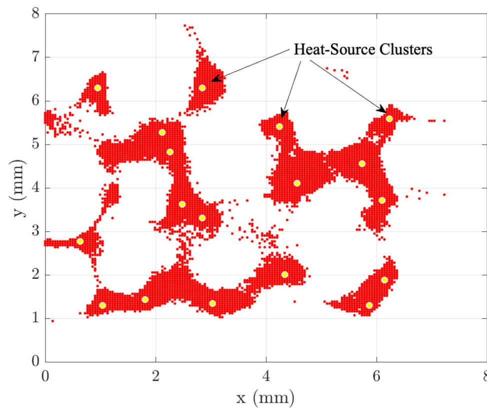


Fig. 7. Distinct heat sources (red) extracted from 187 200 heatmaps of the Intel i5-3337U and dominant heat-source clusters (yellow) identified using k-means.

TABLE I
IPCM METRICS FOR THE INTEL I5-3337U

Pkg.	Pkg.	Core1.1	Core1.2	Core2.1	Core2.2
exec	inst nom	exec	exec	exec	exec
IPC	inst nom%	IPC	IPC	IPC	IPC
freq	C2res%	freq	freq	freq	freq
afreq	C3res%	afreq	afreq	afreq	afreq
L3 miss	C6res%	L3 miss	L3 miss	L3 miss	L3 miss
L2 miss	C7res%	L2 miss	L2 miss	L2 miss	L2 miss
L3 hit	energy (J)	L3 hit	L3 hit	L3 hit	L3 hit
L2 hit	temp	L2 hit	L2 hit	L2 hit	L2 hit
L3 MPI		L3 MPI	L3 MPI	L3 MPI	L3 MPI
L2 MPI		L2 MPI	L2 MPI	L2 MPI	L2 MPI
read rate		C0res%	C0res%	C0res%	C0res%
write rate		C1res%	C1res%	C1res%	C1res%
inst count		C3res%		C3res%	
ACYC		C6res%		C6res%	
physIPC		C7res%		C7res%	
physIPC%		temp		temp	

value of k) to identify the centroids of the clusters (shown as yellow dots in Fig. 7). We will, from this point, refer to these centroids as our distinct heat-source clusters or simply as heat sources. In total, we were able to identify 18 prominent heat-source clusters on the dual-core Intel i5-3337U processor which has only two on-chip temperature sensors. Likewise, as we will show in Section VIII, 20 heat sources were identified on the quad-core Intel i7-8650U which has only four on-chip temperature sensors.

With all the heat sources on the chip identified, our next goal is to derive a model that can be used to estimate their temperatures in real time. The derivation of this model will be detailed in the subsequent sections. For clarity of the presentation, the discussions in Sections V and VII will only consider the Intel i5-3337U. These exact steps will also be implemented on the Intel i7-8650U. Section VIII will show the implementation of the proposed methodology and experimental results for both chips.

V. Machine Learning-Based Thermal Modeling: Datasets

A. Runtime Temperature Measurement

The heat-source identification method discussed in Section IV allowed us to locate 18 distinct heat-source clusters on the Intel i5-3337U dual-core processor under test. With the thermal model, our goal is to accurately estimate the temperatures of these heat sources during online operation. Hence, in order to train the regression model, we need time-series temperature data of these 18 heat sources. With the IR thermography setup (Fig. 1), we can directly record the temperatures of the identified heat sources while the processor is subjected to a variety of workloads. This temperature data measured directly from the processor gives us a significant advantage in developing an accurate thermal model, as opposed to relying on another previously established model or simulator to acquire the required datasets. In this study, we strictly use first-hand measured data for training, and later for testing the accuracy of the trained model.

B. Runtime Performance Metrics

While the IR setup allows us to capture the temperature of the processor externally, the other major part of the dataset comes from monitoring the utilization of the processor while the temperature data are being recorded. One way to monitor the processor's utilization is through online performance monitoring software, which is supported by most, if not all, major manufacturers of commercial microprocessors. In this work, we use high-level performance metrics provided by tools such as IPCM [37]. These provide a comprehensive high-level view of the processor's utilization with system-level metrics, such as the current frequency of the cores, instruction counts, cache hit/miss rates, sleep-state residency, temperature from the internal sensors, etc. In total, IPCM provides 80 performance metrics (I_1–I_{80}) for the Intel i5-3337U. The complete list of these performance metrics is given in Table I. Since these performance metrics are a good representation of the processor's utilization, we can train a model which can accurately estimate the temperature of the hotspots using these metrics as inputs. Note that it is important to ensure that these performance metrics are captured in synchronous with the thermal data captured by the IR camera. Hence, as previously mentioned, the IR camera's internal waveform generator, along with an I/O device, is used to synchronize the capture rate of the camera and the performance metrics recorded on the test system. This setup ensures that, at the frequency of 60 Hz, one set of IPCM data is recorded in tandem with each set of temperature data captured by the IR camera.

Thermal models based on runtime performance metrics have been demonstrated in the past [38]–[40]. However, the existing methods are not practical to implement in modern commercial processors for several reasons. First, for each FU on the chip, the low-level performance counters that have a significant correlation with the power draw of the given FU must be manually identified. However, this is under the assumption that microbenchmarks can be used to target a single FU in isolation so that this correlation can be determined, this is not feasible in modern processors. Second, even if these correlations can be found, the number of low-level performance counters that can be recorded in parallel is limited by the number of free

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

programmable registers available in the processor. In the case of the Intel i5-3337U, only 11 registers were available. Since more than one metric is typically needed to model the temperature of a single FU, it is not possible to track the temperature of all the FUs on the chip in parallel. Alternatively, in this study, we use high-level performance metrics offered by performance monitors such as IPCM. The correlation between the transient behavior of the high-level performance metrics and the thermal response of the previously identified hotspots are automatically learned through training. This makes the proposed method more practical for modern commercial processors with advanced microarchitectures. Moreover,the proposed approach does not require any information regarding the chip's architecture or floorplan, which is typically necessary for an FU-wise temperature estimation.

VI. Machine Learning-Based Thermal Modeling: IPCM Input Reduction

From the machine learning perspective, the larger the number of inputs and outputs, the more complex the model will be. Currently, we have 80 inputs and 18 outputs. However, not all the inputs are important and relevant from the thermal/power perspective. To eliminate the irrelevant inputs, we need to identify the IPCM metrics that have little correlation with the thermal response of the heat sources. If these metrics are removed, we can reduce the number of inputs to our model while maintaining its accuracy. This step is especially crucial for real-time applications as it leads to a more compact model with lower inference latency.

To determine if a given IPCM metric is relevant to any of the heat sources, we view the IPCM metrics as heat-source stimulants. Hence, our goal will be to identify the IPCM metrics that are effective stimulators for the heat sources on our chip. The IPCM metrics that fail to stimulate any of the heat sources can be deemed thermally irrelevant and can therefore be removed. To this end, we first apply a DVFS heuristic algorithm to simulate spatial power (W$\cdot m^{-}$3) on a given heat source induced by a targeted IPCM metric. We then set up the heat partial differential equation (PDE) for the given heat source. The thermal coefficients for the PDE will be obtained by using the least-squares method. After this, we can compute the estimated input power from the measured temperature at the given heat source. If the targeted IPCM metric is relevant to the given heat source, then the estimated power should agree with the IPCM metric activities. In the following sections, we give a detailed explanation of the proposed method.

A. Transient Power Estimation From Measured Thermal Map

For our problem, we first need to estimate the transient power density at the heat sources from transient heatmaps. Then, we will calculate the correlation in the time domain. The total power of a CMOS digital chip is typically determined by clock frequency, supply voltage, and capacitance of transistors. Today's DVFS techniques couple frequency and voltage (called power states), which makes frequency the only variable needed to determine energy consumption. Official specifications released by Intel [41], [42] show that the active power of the CPU is linearly related to the operating frequency and squarely related to supply voltage, while the static power remains almost constant. In addition, [41] states that frequency and power are linearly related; the official data of power states in [42] illustrate this linear relationship. Thus, the total chip power is cubically related to frequency, which can be expressed by

$$P = CV^2 f + P^{(S)} = \eta f^{\alpha} + P^{(S)} \tag{7}$$

where P denotes the total power, C is the capacitance, V is the supply voltage, f is the operating frequency, which is known, $\alpha = 3$ is a constant, η is the coefficient for frequency scaling, and $P^{(S)}$ is the static power consumption.

As mentioned previously, we simulate the spatial power density by applying the DVFS heuristic algorithm to IPCM metrics. The power density g_n at heat source #n located at (x_n, y_n), can be computed by considering all the IPCM metrics together

$$\begin{aligned} g_n(t) &\propto \sum_{m=1}^{M} \eta_m (I_m(t) + \beta_m) f^{\alpha} + g_n^{(s)} \\ n &\in [1, 2, 3, \ldots, N] \\ m &\in [1, 2, 3, \ldots, M] \end{aligned} \tag{8}$$

where $g_n^{(s)}$ denotes the static power density at heat source #n. M and N denote the total number of IPCM metrics and heat sources, respectively. $I_m(t)$ is the value of the mth IPCM metric at time step t, η_m is a coefficient, and β_m serves as a constant bias for the mth IPCM metric.

Let us consider the case where the power density at heat source #n is very sensitive to one of the IPCM metrics. In this case, the power density at heat source #n must have high correlation with this particular metric. Hence, power density can be estimated using just the targeted IPCM metric

$$g_n(t) \approx \eta_k f^{\alpha} (I_k(t) + \beta_k) + g_n^{(s)} \tag{9}$$

where η_k is a constant coefficient for the targeted kth IPCM metric. In (9), higher sensitivity will lead to higher equality.

In Section IV, a simplified steady-state thermal diffusion equation was used to convert the heatmaps to scaled powermaps. But, in this section, we have to consider the transient effects so we start with the heat PDE in the time domain

$$\rho C_P \frac{\partial T_n(t)}{\partial t} - \kappa \nabla^2 T_n(t) = \eta_k f^{\alpha} (I_k(t) + \beta_k) + g_n^{(s)}. \tag{10}$$

In this equation, we assume that the heat-source temperature T_n is mainly stimulated by the kth IPCM metric. However, this may not always be true. When (10) approximately holds, then the kth IPCM must be highly correlated with this heat source, which is what we are looking for. If (10) does not hold, the weak correlation will manifest in the form of weak cross-correlation (CC) coefficient and spurious thermal constants which will be discussed in Section VI-B. In Section VI-B, we will refine the correlations and extract relevant IPCM metrics. Before this analysis, let us rewrite (10) as

$$\frac{\rho C_P}{\eta_k} \frac{\partial T_n(t)}{\partial t} - \frac{\kappa}{\eta_k} \nabla^2 T_n(t) = f^{\alpha} (I_k(t) + \beta_k) + \frac{g_n^{(s)}}{\eta_k} \tag{11}$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

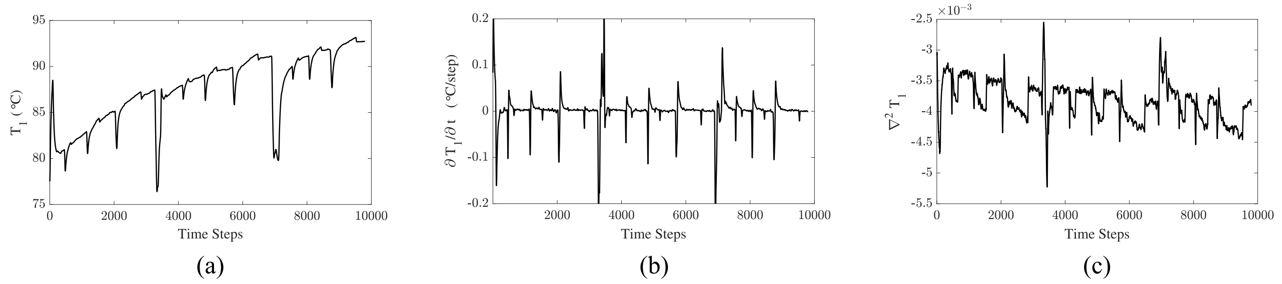


Fig. 8. Measured data. (a) Transient temperature at heat source #1 over time. (b) Time derivative of temperature. (c) Laplacian of temperature at heat source #1.

where $T_n(t)$ is the temperature at heat source #n at time step t acquired from measured thermal data. $(\rho C_P/\eta_k)$ and (κ/η_k) are thermal constants scaled by η_k. Suppose we have sufficient amount of time steps, by stacking all the data along time vertically, (11) can be rewritten as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{12}$$

where $\boldsymbol{A}$ is a matrix with four columns, $\boldsymbol{x}$ is an unknown vector with four elements, and $\boldsymbol{b}$ is a vector of known IPCM data, i.e.,

$$\begin{aligned} \boldsymbol{A}_t &= \begin{bmatrix} \frac{\partial T_n(t)}{\partial t} & -\nabla^2 T_n(t) & -f(t)^\alpha & -1 \end{bmatrix} \\ \boldsymbol{x} &= \begin{bmatrix} \frac{\rho C_P}{\eta_k} & \frac{\kappa}{\eta_k} & \beta_k & \frac{g_n^{(s)}}{\eta_k} \end{bmatrix}^T \\ \boldsymbol{b}_t &= f(t)^\alpha I_k(t). \end{aligned} \tag{13}$$

Here, the optimal solution for $\boldsymbol{x}$ can be obtained by applying the least-squares method

$$\begin{aligned} \boldsymbol{x}^* &= \left[\left(\frac{\rho C_P}{\eta_k}\right)^* \ \left(\frac{\kappa}{\eta_k}\right)^* \ (\beta_k)^* \ \left(\frac{g_n^{(s)}}{\eta_k}\right)^* \right]^T \\ &= (\boldsymbol{A}^T \cdot \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}. \end{aligned} \tag{14}$$

Then, we obtain the two power densities for the given heat source. The estimated power density $g_{n,\text{est}}(t)$ by substituting x^* back to (11)

$$g_{n,\text{est}}(t) = \left(\frac{\rho C_P}{\eta_k}\right)^* \frac{\partial T_n(t)}{\partial t} - \left(\frac{\kappa}{\eta_k}\right)^* \nabla^2 T_n(t) \tag{15}$$

and the IPCM activity-related power density $g_{n,I_k}(t)$

$$g_{n,I_k}(t) = f(t)^\alpha \big(I_k(t) + (\beta_k)^*\big) + \left(\frac{g_n^{(s)}}{\eta_k}\right)^*. \tag{16}$$

The transient power density estimation $g_{n,\text{est}}(t)$ contains two contributions: the time derivative for the first term and the Laplacian of temperature for the second term. Fig. 8 shows the two components for one heat-source location (heat source #1). As we can see, both components are quite significant and should be considered. Now, we are ready to compute the correlation by looking at those two power densities over time.

B. IPCM Correlation Analysis and Refinement

We now compute the power density correlation between the heat source #n and the targeted kth IPCM metric. We use the CC definition [43] of two deterministic and discretized digital signals, which measures the degree of similarity between the two time-series signals

$$\begin{aligned} X_{n,k}[t_i] &= \big(g_{n,\text{est}} \otimes g_{n,I_k}\big)[t_i] \\ &= \sum_{j=-\infty}^{\infty} g_{n,\text{est}}(j) \cdot g_{n,I_k}(t_i + j) \end{aligned} \tag{17}$$

where t_i is the discretized time point that we are interested in and $\otimes$ indicates the convolution operation. Then, we use the normalized maximum absolute value of $X_{n,k}$, between [0, 1], as the CC measurement of the two power signals.

In the following discussions, we take heat source #1 (measured thermal data in Fig. 8) with the 11th IPCM metric, i.e., *read rate* [Fig. 9(a)], as an example to illustrate a strong correlation. We also analyze this same heat source with the 31st IPCM metric, i.e., *L3 hit* [Fig. 9(b)], as a counter example to show weak correlation. The estimated power density and IPCM activity-related power density for the two IPCM metrics are compared in Fig. 9(c) and (d), respectively. As we can see, a strong correlation [0.88, Fig. 9(e)] is observed for the 11th IPCM metric, while a weak correlation [0.28, Fig. 9(f)] is observed for the 31st IPCM metric. Although, heat source #1 is weakly correlated with the 31st IPCM metric, it is possible that this metric is better correlated with some other heat source. It is also possible that more than one IPCM metrics are highly correlated with heat source #1. In Fig. 10(a), we show the CC coefficient of heat source #1 with respect to all 80 IPCM metrics. As we can see, it is highly correlated with more than one metric. Note that CC coefficient of 24th, 40th, and 68th IPCM metrics are set to zero since they are temperature sensor data which are not power stimulants.

Our detailed study shows that some correlations are still spurious. We observe that the thermal constants ratio $(\rho C_P)/\kappa$ obtained from optimal solution x^* (14) is not constant for different IPCM metrics as shown in Fig. 10(b). This indicates that the related correlations have some error and noise. In order to address this problem, we need to find the real ratio of thermal constants. A simple technique we apply is to extract the IPCM metrics that have high correlations (e.g., > 0.8), and then take an average of their ratios $(\rho C_P/\kappa)$ as the real ratio. Then, we update the correlations by fixing this ratio for (10). By using this technique, we identify seven relevant IPCM metrics for heat source #1; their associated thermal constants ratios are shown in Fig. 10(c). After this, we proceed to compute the average ratio $(\rho C_P)/\kappa$, which is 0.0053. In this case study,

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

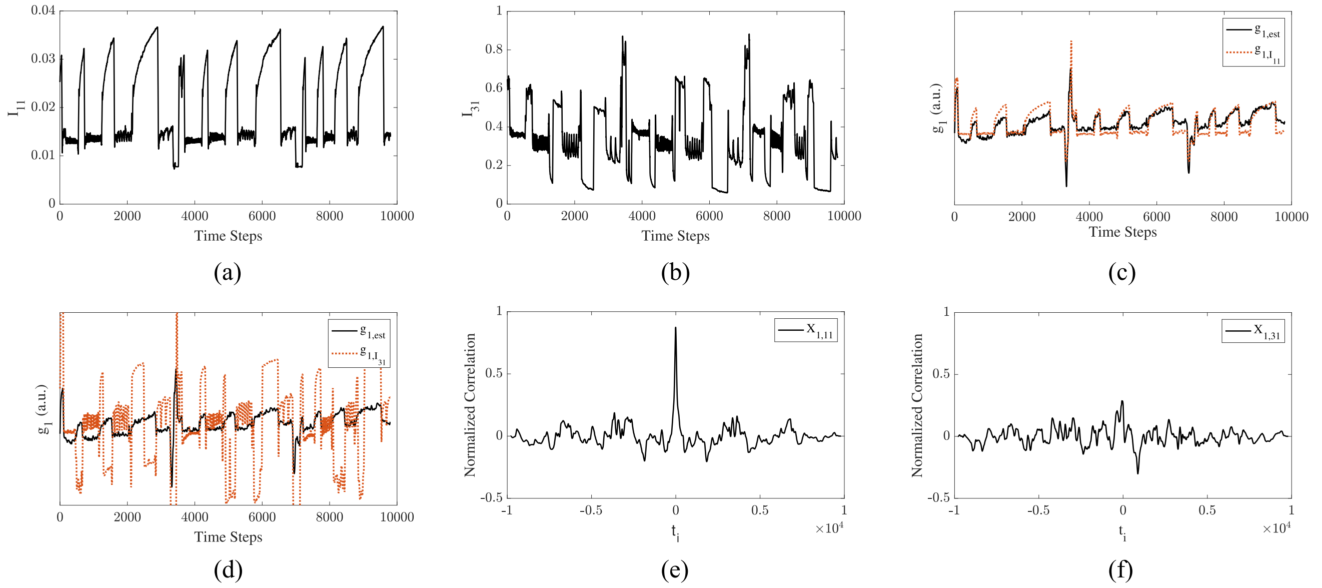


Fig. 9. (a) 11th IPCM data I_{11}. (b) 31st IPCM data I_{31}. (c) Estimated power density at heat source #1 compared to the 11th IPCM-related power density. (d) Estimated power density at heat source #1 compared to 31st IPCM-related power density. (e) CC between heat source #1 and 11th IPCM with coefficient 0.88. (f) CC between heat source #1 and 31st IPCM with coefficient 0.28.

TABLE II
REDUCED PERFORMANCE METRICS (INTEL PCM)

Pkg.	Pkg.	Core1.1	Core1.2	Core2.1	Core2.2
exec	write rate	IPC	exec	exec	afreq
IPC	inst count	freq	IPC	IPC	L2 miss
freq	ACYC	L2 miss	freq	freq	
afreq	physIPC	L2 hit	afreq	L3 miss	
L3 miss	physIPC%	L2 MPI	L3 miss	L3 hit	
L2 miss	inst nom	C0res%	L2 miss	L3 MPI	
L3 hit	inst nom%	C7res%	L3 hit	L2 MPI	
L3 MPI	energy (J)	temp	L3 MPI	C0res%	
L2 MPI	temp		L2 MPI	temp	
read rate			C0res%		

the updated nonspurious CC coefficients are shown in red in Fig. 10(a).

We performed this analysis for all 18 heat sources which yielded an average ratio $(\rho C_P)/\kappa = 0.0048$. We then used this ratio for (10) to identify all the relevant IPCM metrics, which are listed in Table II. In total, we were able to identify 47 IPCM metrics that are highly correlated with the heat sources on the Intel i5-3337U. This is a significant reduction from the original 80 IPCM metrics previously shown in Table I.

VII. Machine Learning-Based Thermal Modeling: Network Architecture

Since online temperature estimation of a microprocessor is very much a time-series problem, we need a method that is naturally suited for modeling such a system. One option would be to use a statistical analysis-based approach, such as autoregressive moving average (ARMA) or even simple least-squares regression models. The proposed heat-source identification, data acquisition, and preparation methods discussed previously can also be applied to fit these models as well. However, in this work, we will instead be utilizing deep learning as the recent advancements in this area have shown promising results in time-series estimation and pattern recognition tasks [44], [45]. RNNs are the classical architecture designed for such tasks. In this work, we will utilize a specialized subset of RNNs, called the LSTM network, which uses gated internal states making it ideal for problems that require substantial temporal resolution. For brevity, we refer the readers to [44] for detailed discussions and analysis of LSTM networks.

While the NN architecture of choice was obvious, there is no standard method of determining the size and depth of the network that is optimal for the problem at hand. In most cases, some experimentation is necessary to determine the smallest network that is robust enough to accurately model the given problem. The network size is especially crucial for online/real-time applications, like that one explored in this study, where the model should be lightweight enough for inference at moderate to high frequencies with low computational overhead. In this work, our goal is to do nearly real-time temperature estimation. Meaning, we want to estimate the temperature at time t by time $t + t_{\text{inference}}$, where $t_{\text{inference}}$ is the time taken for each inference. Hence, in order to be as close to real time as possible, it is imperative to reduce $t_{\text{inference}}$ as much as possible without deteriorating accuracy. The IPCM reduction method from the previous section aids in reducing $t_{\text{inference}}$ by reducing the input dimensionality of the model, however, another factor that affects $t_{\text{inference}}$ is the network architecture itself. To this end, in this section, we explore various network depths and layer sizes to determine the optimal configuration. Table III shows the estimation error and inference times for

[1] In network configurations, $\text{LayerType}_n\{\mathbf{m}\}$ is a condensed representation describing the structure of the NN. Here, LayerType refers to the type of hidden layer (i.e., LSTM or Dense), subscript n refers to the number of the aforementioned layers in the network, and the $1 \times n$ vector $\mathbf{m}$ refers to the number of nodes in each of the respective layers (i.e., $\text{LSTM}_3\{100, 70, 40\}$ + $\text{Dense}_1\{18\}$ refers to a network with three LSTM layers, with 100, 70, and 40 nodes per layer, respectively, and one dense layer with 18 nodes).

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE COMPARISONS BETWEEN VARIOUS NN CONFIGURATIONS

Network Configuration[2]	All IPCM		Reduced IPCM	
	RMS Error (°C)	Inference Time (sec.)	RMS Error (°C)	Inference Time (sec.)
$\text{LSTM}_1\{10\}$ + $\text{Dense}_1\{18\}$	1.203	0.461×10^{-3}	0.980	0.436×10^{-3}
$\text{LSTM}_1\{18\}$ + $\text{Dense}_1\{18\}$	0.821	0.482×10^{-3}	0.927	0.459×10^{-3}
$\text{LSTM}_1\{30\}$ + $\text{Dense}_1\{18\}$	0.785	0.543×10^{-3}	0.851	0.515×10^{-3}
$\text{LSTM}_1\{50\}$ + $\text{Dense}_1\{18\}$	0.791	0.668×10^{-3}	0.679	0.581×10^{-3}
$\text{LSTM}_1\{100\}$ + $\text{Dense}_1\{18\}$	0.748	1.082×10^{-3}	0.684	1.047×10^{-3}
$\text{LSTM}_2\{100,40\}$ + $\text{Dense}_1\{18\}$	0.827	1.748×10^{-3}	0.717	1.687×10^{-3}
$\text{LSTM}_3\{100,70,40\}$ + $\text{Dense}_1\{18\}$	0.881	2.662×10^{-3}	0.699	2.544×10^{-3}

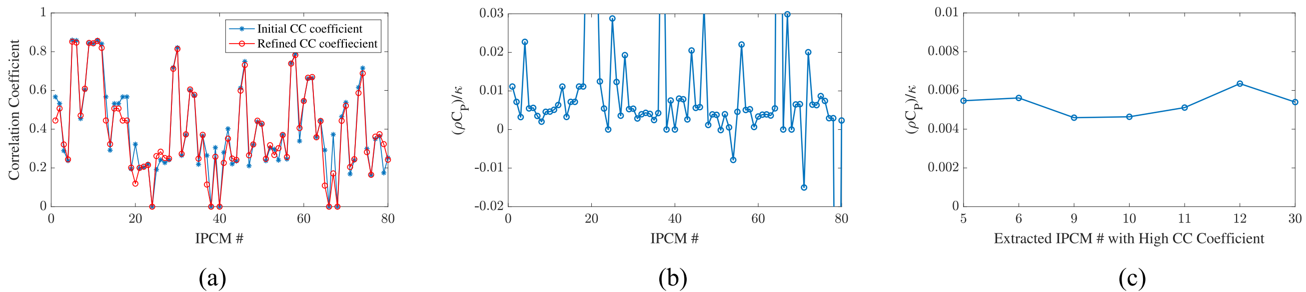


Fig. 10. (a) CC coefficient between heat source #1 and 80 IPCM metrics. Blue dot-marked trace includes the spurious CC coefficient, while the red circle-marked trace is for after refinement. (b) Ratio of thermal constants $(\rho C_P)/\kappa$ associated with 80 IPCM metrics. (c) Ratio of thermal constants for relevant IPCM metrics that have high CC coefficients.

various networks. Each network was trained twice, first with all 80 IPCM metrics as inputs, and a second time with only the thermally relevant 47 IPCM metrics as inputs. The results for both cases are shown in Table III.

From the analysis in Table III, it is clear that as the network size grows, so does the inference time and consequently the performance overhead of the model. The thermal time constant (τ) for semiconductor devices is typically in the order of 10^{-3} s (milliseconds); hence, it is important to ensure that the inference time of the model is equal to or less than τ. To be conservative, we will aim for an inference time less than 1 millisecond (ms). In addition to minimizing inference time, the model must also yield usable accuracy. The analysis in Table III shows that the model with the fastest inference time also has the worst accuracy, hence there must be a tradeoff between the two. However, we found that accuracy does not continue to improve as the network size grows; after a certain threshold, accuracy saturates or, in some cases, even declines. This is due to the fact that larger networks are generally more difficult to train, often requiring meticulous tuning. While such tuning can yield higher accuracy, a larger network performing worse than a smaller one is usually a sign that the network has grown larger than necessary for modeling the given problem.

Hence, based on the aforementioned observations, we chose to proceed with $\text{LSTM}_1\{50\}$ + $\text{Dense}_1\{18\}$, which has one LSTM layer with 50 nodes and one dense layer with 18 nodes as shown in Fig. 11. This network offers a good tradeoff between accuracy and inference time. When all 80 IPCM metrics are used as inputs, the network yields an overall RMS error of 0.79 °C and an inference time of 0.67 ms. Utilizing the input reduction method discussed in Section VI, the same network should maintain its accuracy while yielding a faster inference time. This is indeed the case, when trained with only the thermally relevant 47 IPCM metrics as inputs, the network performs even better with an overall RMS error of 0.68 °C and an inference time of 0.58 ms. Here, as the input dimensionality is decreased, the inference time also decreases. Additionally, the accuracy of the model improves as well since the IPCM metrics with little or no correlation with the thermal response of the chip are removed, resulting in less noise at the input. Further details of the experimental setup and performance analysis will be presented in the next section.

VIII. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the experimental results from the proposed system-level thermal modeling approach. The proposed method has been implemented on two Intel processors: 1) the Intel Core i5-3337U with 2 cores/4 threads, which is the chip that was used in all of the previous discussions and 2) the Intel Core i7-8650U with 4 cores/8 threads. It should be noted that, in this work, only one i5-3337U and one i7-8650U chip were used to collect all of the training datasets. However, it is generally recommended to use multiple sample chips of the same type to gather the datasets in order to account for the statistical variations between the chips. Additionally, it should be noted that the proposed heat-source identification method and the proposed data collection and preparation methods can be used to fit any regression-based model, not just train an NN-based machine learning model. To this end, the same datasets

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RMSE FOR EACH HEAT SOURCE (i5-3337U)

Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg
HS#1	0.694°C	2.528°C	HS#6	0.737°C	3.091°C	HS#11	0.547°C	2.036°C	HS#16	0.651°C	2.701°C
HS#2	0.762°C	2.861°C	HS#7	0.584°C	2.031°C	HS#12	0.678°C	2.244°C	HS#17	0.734°C	3.084°C
HS#3	0.696°C	2.921°C	HS#8	0.615°C	2.383°C	HS#13	0.710°C	2.809°C	HS#18	0.686°C	2.393°C
HS#4	0.697°C	1.992°C	HS#9	0.686°C	2.312°C	HS#14	0.623°C	2.276°C			
HS#5	0.736°C	2.796°C	HS#10	0.731°C	3.013°C	HS#15	0.686°C	2.609°C			

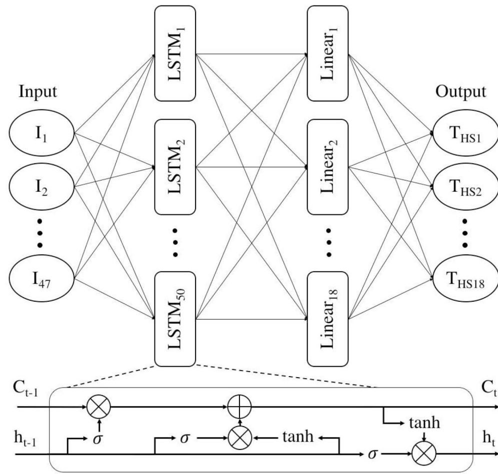


Fig. 11. LSTM network architecture.

used to train the LSTM network were used to fit a simple least-squares linear regression model so comparisons can be made between the estimation accuracy and inference times of the two approaches.

A. Results From Intel Core i5-3337U

From the Core i5-3337U, a total of 187 200 data points were collected which, considering the capture rate of 60 Hz, constitutes to 52 min of continuous runtime. Each data point consists of 80 IPCM metrics captured internally on the test system, and the temperatures of the previously identified 18 heat sources captured via the thermal imaging setup. During this time, the processor was subjected to a variety of realistic workloads. These range from lightweight workloads like idling and word processing, to intensive workloads like data compression. Some workloads were primarily compute-intensive tasks while others were memory intensive. The idea is to utilize all the different subsystems in the processor during the course of data acquisition so that their thermal response can be recorded. The amount of data that is required for the proposed approach will depend on the processor and its application. As a rule of thumb, with any machine-learning-based approach, more data will generally always lead to a better model. Additionally, it is important to ensure that the workloads used during data acquisition are as diverse as possible. One option to ensure diversity is to use a benchmark suite, such a Phoronix test suite, that offers workloads that vary in hardware utilization and intensity [46].

Once all the data are acquired, the method discussed in Section VI was used to select the IPCM metrics that are highly correlated with the previously identified heat sources. Only using these metrics to train the model allowed us to reduce the input dimensionality of the model from 80 to 47. After this step, the network shown in Fig. 11 was trained for a total of 50 epochs with 60 time steps used for the LSTM layers. Out of the 187 200 data points collected for this study, the first 60% were used for training, the next 15% were used for validation, and the remaining 25% were used for testing. Similarly, for the least-squares regression model, the first 75% of the dataset were used for fitting and the remaining 25% were used for testing. The training and testing datasets were kept completely isolated from each other in order to ensure that no testing data are used in the training process.

Formal testing and validation carried out on the final LSTM model shows that it performs exceptionally well. The results presented in Fig. 12(a) show the model estimating the run-time temperature of heat source #1 for a duration of about 8 min. The measured temperature from the IR camera is overlaid on top of the estimation for comparison. For brevity, we only show this plot for one heat source; however, the RMSE computed for all 18 heat sources is presented in Table IV. In summary, the highest RMSE was 0.76 °C (HS #10) while the lowest was 0.55 °C (HS #2). Considering the observed dynamic range of 58.73 °C–101.35 °C, these constitute a relative RMSE of 1.79% and 1.28%, respectively. The same tests run on the least-squares linear regression-based model shows that this approach produces lower estimation accuracy with the highest RMSE of 3.09 °C (HS #6) and the lowest RMSE of 1.99 °C (HS #4). These constitute a relative RMSE of 7.25% and 4.67%, respectively. RMSE computed for all 18 heat sources is shown in Table IV. While accuracy suffers with the simple regression model, it does have the advantage of extremely fast-fitting time and inference time. While the LSTM model took more than a day to train, the least-squares model took a few seconds to fit after the data had been prepared. Similarly, while the inference time of the LSTM model is in the order of milliseconds ($\sim$ 0.58 ms), as discussed in Section VII, the inference time of the regression model is in the order of microseconds ($\sim$ 0.62 μs). Hence, there is a tradeoff between the two black-box modeling approaches.

B. Results From Intel Core i7-8650U

To further validate the proposed approach, it was also implemented on the Intel Core i7-8650U. Using the heat-source

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

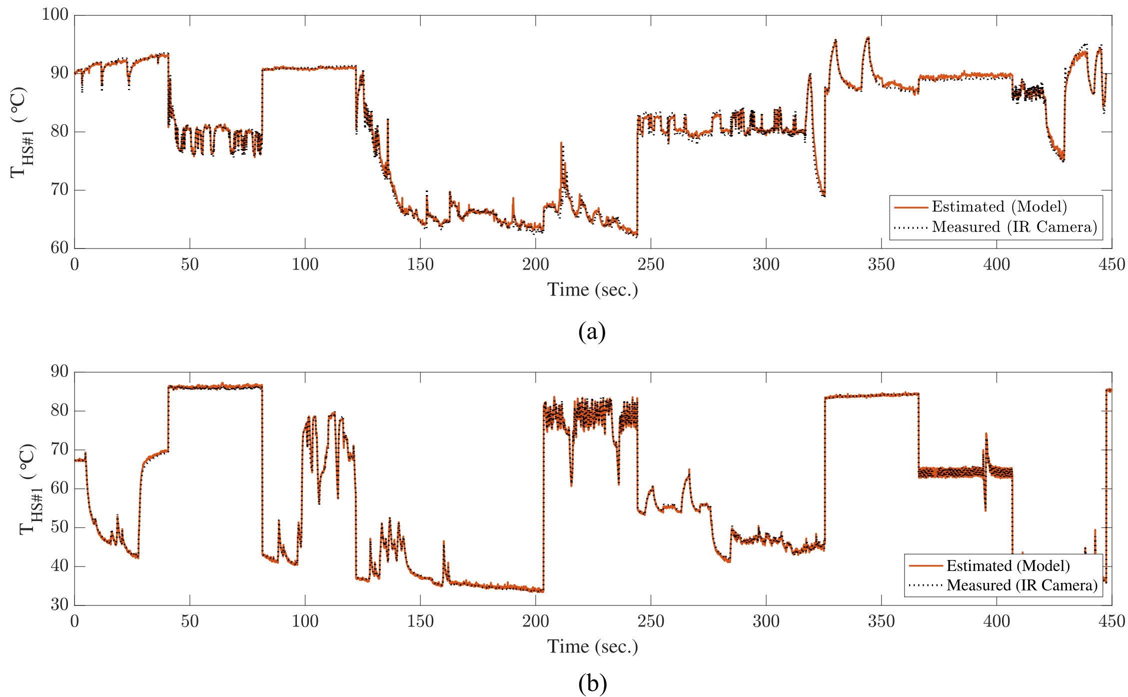


Fig. 12. Estimated versus measured runtime temperature of heat source #1. (a) i5-3337U. (b) i7-8650U.

TABLE V
RMSE FOR EACH HEAT SOURCE (I7-8650U)

Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg
HS#1	0.737°C	1.830°C	HS#6	0.690°C	1.841°C	HS#11	0.852°C	2.775°C	HS#16	0.804°C	2.045°C
HS#2	0.802°C	3.181°C	HS#7	0.654°C	2.089°C	HS#12	0.691°C	2.208°C	HS#17	0.694°C	2.480°C
HS#3	0.898°C	2.799°C	HS#8	0.679°C	2.559°C	HS#13	0.859°C	2.837°C	HS#18	0.783°C	3.198°C
HS#4	0.934°C	2.396°C	HS#9	0.734°C	2.146°C	HS#14	0.617°C	2.106°C	HS#19	0.717°C	2.365°C
HS#5	0.869°C	2.741°C	HS#10	0.747°C	2.056°C	HS#15	0.710°C	2.002°C	HS#20	0.815°C	1.865°C

identification method presented in Section IV, a total of 20 heat sources were detected on the Intel i7-8650U. Then, 288 000 time steps of runtime temperature data of the 20 heat sources were recorded along with synchronized runtime IPCM data. In total, IPCM provides 170 metrics for the Intel i7-8650U. With the method discussed in Section VI, 72 thermally relevant metrics were selected to be used as inputs to the model. With this reduction, the final model will have an input dimensionality of 72 (for the 72 IPCM metrics) and an output dimensionality of 20 (for the temperatures of the 20 heat sources). With this in mind, the analysis presented in Section VII was then preformed to find a network that offers a reasonable tradeoff between accuracy and inference time. The network that was selected is very similar to the one shown in Fig. 11 but with 75 LSTM nodes in the first layer and 20 dense nodes in the second layer. Once the network was selected, it was then trained for a total of 50 epochs with 65% of the data used for training, 15% used for validation, and the remaining 25% used for testing. The same data were also used to fit a simple least-squares linear regression model, where the first 75% of the data were used to fitting the model and the remaining 25% of the data were used for testing. Using the same training and testing data on two different black-box modeling approaches allows us to compare and contrast the advantages and disadvantages of the two. The results presented in Fig. 12(b) show the LSTM model estimating the runtime temperature of heat source #1 for a duration of about 8 min. The RMSE for each of the 20 heat sources is given in Table V for both the LSTM model and the least-squares regression model.

The results for the Intel i7-8650U are very comparable to what was achieved with the Intel i5-3337U. Here, the LSTM model yielded the highest RMSE of 0.93 °C (HS#4) and the lowest of 0.62 °C (HS#14). Considering the observed dynamic range of 28.9 °C–97.9 °C, this constitutes to a relative RMSE of 1.35% and 0.89%, respectively. While the least-squares linear regression model yielded the highest RMSE of 3.19 °C (HS#2) and the lowest RMSE of 1.83 °C (HS#14). These constitute a relative RMSE of 4.62% and 2.65%, respectively.

IX. CONCLUSION

In this article, we have presented a novel method of systematically identifying all prominent heat sources on commercial

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

processors and deriving a dynamic thermal model to estimate the temperatures of the identified heat sources during online use. Unlike many existing studies, this work exclusively utilizes measured data gathered directly from commercial off-the-shelf processors. Additionally, the proposed approach inherently avoids all the major obstacles faced by traditional methods that currently exist in the literature, allowing it to be easily deployed by chip manufacturers and third-parties alike. The experimental results on two Intel multicore CPUs show that the proposed thermal model achieves very high accuracy (RMSE: 0.55 °C–0.76 °C on the Intel i5-3337U and 0.62 °C–0.93 °C on the Intel i7-8650U) in estimating the temperatures of all the identified heat sources on the two chips. These results make the proposed approach very desirable for dynamic thermal management schemes which now rely heavily on the temperature data from just the on-chip temperature sensors alone. The high spatial resolution yielded by the proposed approach can help greatly in supplementing the temperature data from the on-chip sensors, allowing for the development of more robust and smarter online thermal/power control schemes.

REFERENCES

[1] *Critical Reliability Challenges for the International Technology Roadmap for Semiconductors (ITRS)*, document 03024377A, Int. Sematech Technol. Transfer, New York, NY, USA, 2003.

[2] H. Amrouch, V. M. van Santen, T. Ebi, V. Wenzel, and J. Henkel, “Towards interdependencies of aging mechanisms,” in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2014, pp. 478–485.

[3] S. Sadiqbatcha, Z. Sun, and S. X.-D. Tan, “Accelerating electromigration aging: Fast failure detection for nanometer ICs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 4, pp. 885–894, Apr. 2020.

[4] D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Jan. 2001, pp. 171–182.

[5] V. Hanumaiah and S. Vrudhula, “Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling,” *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 349–360, Feb. 2014.

[6] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, “Task migrations for distributed thermal management considering transient effects,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 397–401, Feb. 2015.

[7] H. Wang *et al.*, “Hierarchical dynamic thermal management method for high-performance many-core microprocessors,” *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, pp. 1–21, Jul. 2016.

[8] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” in *Proc. Int. Symp. Comput. Architect.*, 2003, pp. 2–13.

[9] J. Kong, S. W. Chung, and K. Skadron, “Recent thermal management techniques for microprocessors,” *ACM Comput. Surveys*, vol. 44, pp. 1–42, Jun. 2012.

[10] H. Amrouch and J. Henkel, “Lucid infrared thermography of thermally-constrained processors,” in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2015, pp. 347–352.

[11] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, “HotSpot: A compact thermal modeling methodology for early-stage VLSI design,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[12] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, “ISAC: Integrated space and time adaptive chip-package thermal analysis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 1, pp. 86–99, Jan. 2007.

[13] H. Wang, S. X.-D. Tan, G. Liao, R. Quintanilla, and A. Gupta, “Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management,” in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2011, pp. 716–723.

[14] S. P. Gurrum, Y. K. Joshi, W. P. King, K. Ramakrishna, and M. Gall, “A compact approach to on-chip interconnect heat conduction modeling using the finite element method,” *J. Electron. Packag.*, vol. 130, pp. 1–8, Sep. 2008.

[15] Y. C. Gerstenmaier and G. Wachutka, “Rigorous model and network for transient thermal problems,” *Microelectron. J.*, vol. 33, pp. 719–725, Sep. 2002.

[16] D. Li, S. X.-D. Tan, E. H. Pacheco, and M. Tirumala, “Parameterized architecture-level dynamic thermal models for multicore microprocessors,” *ACM Trans. Design Autom. Electron. Syst.*, vol. 15, no. 2, pp. 1–22, 2010.

[17] T. Eguia *et al.*, “General parameterized thermal modeling for high-performance microprocessor design,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 211–224, Feb. 2012.

[18] Z. Liu, S. X.-D. Tan, H. Wang, Y. Hua, and A. Gupta, “Compact thermal modeling for packaged microprocessor design with practical power maps,” *Integr. VLSI J.*, vol. 47, no. 1, pp. 71–85, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167926013000412

[19] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, “Efficient power modeling and software thermal sensing for runtime temperature monitoring,” *ACM Trans. Design Autom. Electron. Syst.*, vol. 12, no. 3, pp. 1–29, 2007.

[20] K. Dev, A. N. Nowroz, and S. Reda, “Power mapping and modeling of multi-core processors,” in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Sep. 2013, pp. 39–44.

[21] M. Witkowski, A. Oleksiak, T. Piontek, and J. Weglarz, “Practical power consumption estimation for real life HPC applications,” *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 208–217, 2013.

[22] M. J. Walker *et al.*, “Accurate and stable run-time power modeling for mobile and embedded CPUs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 1, pp. 106–119, Jan. 2017.

[23] F. Pittino, F. Beneventi, A. Bartolini, and L. Benini, “A scalable framework for online power modeling of high-performance computing nodes in production,” in *Proc. Int. Conf. High Perform. Comput. Simulat. (HPCS)*, Jul. 2018, pp. 300–307.

[24] R. Diversi, A. Tilli, A. Bartolini, F. Beneventi, and L. Benini, “Bias-compensated least squares identification of distributed thermal models for many-core systems-on-chip,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 9, pp. 2663–2676, Sep. 2014.

[25] R. Cochran and S. Reda, “Consistent runtime thermal prediction and control through workload phase detection,” in *Proc. Design Autom. Conf. (DAC)*, 2010, pp. 62–67.

[26] A. Bartolini, R. Diversi, D. Cesarini, and F. Beneventi, “Self-aware thermal management for high-performance computing processors,” *IEEE Des. Test*, vol. 35, no. 5, pp. 28–35, Oct. 2018.

[27] F. Beneventi, A. Bartolini, P. Vivet, and L. Benini, “Thermal analysis and interpolation techniques for a logic + WideIO stacked DRAM test chip,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 4, pp. 623–636, Apr. 2016.

[28] R. Cochran and S. Reda, “Spectral techniques for high-resolution thermal characterization with limited sensor data,” in *Proc. Design Autom. Conf. (DAC)*, 2009, pp. 478–483.

[29] S. Reda, R. Cochran, and A. N. Nowroz, “Improved thermal tracking for processors using hard and soft sensor allocation techniques,” *IEEE Trans. Comput.*, vol. 60, no. 6, pp. 841–851, Jun. 2011.

[30] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli, “Eigenmaps: Algorithms for optimal thermal maps extraction and sensor placement on multicore processors,” in *Proc. ACM 49th Annu. Design Autom. Conf. (DAC)*, 2012, pp. 636–641.

[31] R. Cochran, A. Nowroz, and S. Reda, “Post-silicon power characterization using thermal infrared emissions,” in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2010, pp. 331–336.

[32] S. Reda, K. Dev, and A. Belouchrani, “Blind identification of thermal models and power sources from thermal measurements,” *IEEE Sens. J.*, vol. 18, no. pp. 680–691, Jan. 2018.

[33] FLIR. *FLIR A325SC*. Accessed: Jan. 9, 2019. [Online]. Available: https://www.flir.com/products/a325sc/

[34] Thermonamic. *Thermoelectric Module TEC1–12710*. Accessed: Jan. 9, 2019. [Online]. Available: http://www.thermonamic.com/TEC1-12710-English.pdf

[35] Intel. *Technical Resources: Intel Core Processors*. Accessed: Jan. 9, 2019. [Online]. Available: https://www.intel.com/content/www/us/en/products/docs/processors/core/core-technical-resources.html

[36] F. P. Incropera and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 5th ed. New York, NY, USA: Wiley, 2002.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.

[37] Intel. *Intel Performance Counter Monitor (PCM)*. Accessed: Jan. 9, 2019. [Online]. Available: https://software.intel.com/en-us/articles/intel-performance-counter-monitor

[38] K. Lee and K. Skadron, "Using performance counters for runtime temperature sensing in high-performance processors," in *Proc. 19th IEEE Int. Parallel Distrib. Process. Symp.*, Apr. 2005, p. 8.

[39] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware DVFs," *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 127–133, Jan. 2010.

[40] H. Wang, S. X.-D. Tan, S. Swarup, and X. Liu, "A power-driven thermal sensor placement algorithm for dynamic thermal management," in *Proc. Design Autom. Test Europe Conf. (DATE)*, Mar. 2013, pp. 1215–1220.

[41] M. Dixon, P. Hammarlund, S. Jourdan, and R. Singhal, *The Next-Generation Intel Core Microarchitecture*, vol. 14, Intel, San Jose, CA, USA 2010.

[42] E. Intel, "Speedstep technology for the Intel Pentium M processor," Intel, San Jose, CA, USA, White Paper, 2004.

[43] R. Bracewell, *Pentagram Notation for Cross Correlation. The Fourier Transform and Its Applications*, vol. 46. New York, NY, USA: McGraw-Hill, 1965.

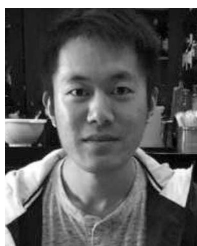
[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[46] Phoronix. *Open-Source, Automated Benchmarking*. Accessed: Jan. 9, 2019. [Online]. Available: https://www.phoronix-test-suite.com/

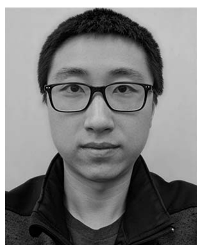
Sheriff Sadiqbatcha (Graduate Student Member, IEEE) received the B.S. degree (most outstanding graduate of the year) in computer engineering from California State University, Bakersfield, CA, USA, in 2016, and the M.S. degree (*magna cum laude*) in electrical engineering from the University of California at Riverside, Riverside, CA, USA, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering.

He is also an Associate Instructor with the Department of Electrical and Computer Engineering, University of California at Riverside. His research interests include presilicon reliability (electromigration), and applied machine learning in the area of electronic design automation, post-silicon thermal, power, and reliability modeling and control.



Jinwei Zhang (Student Member, IEEE) received the B.S. degree in electrical and control engineering from the Beijing Institute of Technology, Beijing, China, in 2014, and the M.S. degree in electrical engineering from Washington University in Saint Louis, USA, in 2016. He is currently pursuing the Ph.D. degree with VSCLAB, Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA.

His research interests are in electronic design automation, system-level thermal and power modeling for VLSI, and optimization with modern machine learning techniques.



Hengyang Zhao (Student Member, IEEE) received the B.S. degree in computer science and the M.S. degree in metering and instrumentation engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from the University of California at Riverside, Riverside, CA, USA, in 2018.

He is currently a Software Engineer with Google, Inc, Mountain View, CA, USA. His current research interests include VLSI reliability modeling, smart building energy optimization, finite-element method-based simulation, and machine learning applications.

Hussam Amrouch (Member, IEEE) received the Ph.D. degree (*summa cum laude*) from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2015.

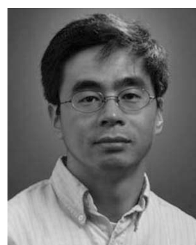
He is currently a Research Group Leader with the Chair for Embedded Systems, KIT. He is also leading the Dependable Hardware Research Group. His main research interests are emerging technologies, design for reliability from physics to system level, and machine learning.

Dr. Amrouch holds seven HiPEAC Paper Awards. He has three best paper nominations at DAC'16, DAC'17, and DATE'17 for his work on reliability. He has served on the technical program committees of Design Automation Conference, International Conference on Computer Aided Design, and Asia and South Pacific Design Automation Conference. He currently serves as an Associate Editor for *Integration* and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He also severs as a reviewer of several journals, such as the IEEE TRANSACTIONS ON ELECTRON DEVICES, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.

Jörg Henkel (Fellow, IEEE) received the Diploma degree and the Ph.D. degree (*summa cum laude*) from the Technical University of Braunschweig, Braunschweig, Germany, in 1996.

Before that, he was a Research Staff Member with NEC Laboratories, Princeton, NJ, USA. He is currently the Chair Professor of embedded systems with the Karlsruhe Institute of Technology, Karlsruhe, Germany. His research interests include co-design for embedded hardware/software systems with respect to power, thermal, and reliability aspects.

Dr. Henkel has received six best paper awards throughout his career from, among others, ICCAD, ESWeek, and DATE. For two consecutive terms, he served as the Editor-in-Chief for the *ACM Transactions on Embedded Computing Systems*. He is also the Editor-in-Chief of the *IEEE Design & Test Magazine*. He is/has been an Associate Editor for major ACM and IEEE journals. He has led several conferences as a General Chair, including ICCAD and ESWeek. He has serves as a Steering Committee Chair/Member for leading conferences and journals for embedded and cyber-physical systems. He coordinates the DFG program SPP 1500 Dependable Embedded Systems. He is a Site Coordinator of the DFG TR89 Collaborative Research Center on Invasive Computing. He is the Chairman of the IEEE Computer Society, Germany Chapter.



Sheldon X.-D. Tan (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, IA, USA, in 1999.

He is a Professor with the Department of Electrical Engineering, University of California at Riverside, Riverside, CA, USA, where he is also a Cooperative Faculty Member with the Department of Computer Science and Engineering. He was a Visiting Professor with Kyoto University, Kyoto, Japan, as a JSPS Fellow from December 2017 to January 2018. His research interests include machine and deep learning for VLSI reliability modeling and optimization at circuit and system levels, machine learning for circuit and thermal simulation, thermal modeling, optimization and dynamic thermal management for many-core processors, and parallel computing and adiabatic and ising computing based on GPU and multicore systems. He has published more than 300 technical papers and has coauthored six books on those areas.

Prof. Tan received the NSF CAREER Award in 2004. He also received three Best Paper Awards from ICSICT'18, ASICON'17, ICCD'07, and DAC'09. He also received the Honorable Mention Best Paper Award from SMACD'18. He is serving as the TPC Chair for ASPDAC 2021, and the TPC Vice Chair for ASPDAC 2020. He is serving or served as the Editor-in-Chief for *Integration* (Elsevier) and *VLSI Journal*, and the Associate Editor for three journals, such as the *IEEE Transaction on VLSI Systems (TVLSI)*, the *ACM Transaction on Design Automation of Electronic Systems*, and *Microelectronics Reliability* (Elsevier).

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on June 24,2026 at 00:44:44 UTC from IEEE Xplore. Restrictions apply.