# Demo Abstract: PRINCE: Device Energy Estimation with a Single Photo

Farooq Dar
University of Tartu
farooq.dar@ut.ee

Mohan Liyanage
University of Tartu
mohan.liyanage@ut.ee

Mayowa Olapade
University of Tartu
mayowa.olapade@ut.ee

Zhigang Yin
University of Tartu
zhigang.yin@ut.ee

Abdul-Rasheed Ottun
University of Tartu
rasheed.ottun@ut.ee

Adeyinka Akintola
University of Tartu
adeyinka.akintola@ut.ee

Francisco Airton Silva
Federal University of Piauí
faps@ufpi.edu.br

Huber Flores*
University of Tartu
huber.flores@ut.ee

## ABSTRACT

We contribute PRINCE, an innovative sensing solution capable of accurately estimating the energy consumption of applications executing on a wider range of smart and IoT devices, including smartwatches, wearables and autonomous drones, without the need for direct instrumentation of the device. Modern devices lack detachable batteries or are sealed, making it challenging to profile their energy consumption. In this demo, we showcase PRINCE, a proof-of-concept prototype that provides precise energy consumption measurements of applications running in devices with a single (thermal) photo. PRINCE harnesses the thermal radiation (heat) generated by the processing units of the device, which is released through the device casing. This allows PRINCE to derive accurate energy estimations of application execution. Extensive benchmarks that compare PRINCE with traditional solutions, such as Monsoon power monitor, demonstrate that PRINCE provide similar performance levels but does not require any instrumentation, facilitating the profiling of the energy consumption of devices.

## KEYWORDS

Battery efficiency, Thermal Imaging, IoT, Temperature

**ACM Reference Format:**
Farooq Dar, Mohan Liyanage, Mayowa Olapade, Zhigang Yin, Abdul-Rasheed Ottun, Adeyinka Akintola, Francisco Airton Silva, and Huber Flores. 2024. Demo Abstract: PRINCE: Device Energy Estimation with a Single Photo. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 BACKGROUND

Modern devices come in a wide variety, from personal devices to autonomous ones like drones, wearables, and IoT devices. Many of these devices are sealed or do not have detachable batteries. This makes it possible to optimize their design and usability; and protect them from environmental factors once they are deployed. In turn, lack of access to batteries makes it difficult to estimate the energy consumption of applications running in these devices. Energy consumption estimation is critical to support development tasks, such as software optimizing, code troubleshooting and partitioning of functionality.

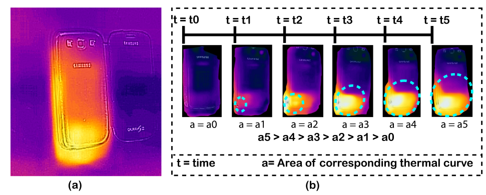


**Figure 1: Heat released from the device casing as an application is executed, a) Single (thermal) photo taken from the back of the phone, b) Heat based on processing load.**

Existing solutions for measuring energy consumption are inadequate due to their reliance on specialized hardware or lack of generality. Hardware-based solutions, in particular, necessitate modifying the device by intercepting connections between the battery and other components. Likewise, software-based methods for estimating power consumption relies on various execution metrics, such as static code analysis or collecting performance metrics of applications during run-time. Moreover, these approaches are susceptible to errors due to differing execution environments, which can lead to significant variations caused by factors like input parameters, concurrent applications, and network connectivity issues. Besides this, external factors like ambient environment, resource availability (e.g., network connectivity), and battery capacity also influence energy drain.

In this demo, we showcase PRINCE as a proof-of-concept prototype that estimates energy consumption of a device just by taking a (thermal) picture. As shown in Figure 1, PRINCE harnesses the generated heat (aka thermal footprint [2]) resulting from application induced processing over the resources of the device. Indeed, released heat correlates with the processing effort required for executing an application [2].

## 2 PRINCE METHODOLOGY IN A NUTSHELL

PRINCE estimates energy consumption from thermal pictures using a three-stage pipeline. Figure 2b) presents a conceptual overview of the PRINCE. We next summarize each stage.

*Corresponding author


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

**(1) Thermal footprint extraction:** removes noise from images (i.e.internal re-calibration noise from camera) and extracts the thermal footprint from the image. Thus, images are cropped to a set size and turned into two-dimensional thermal arrays, where each pixel values holds temperature data estimates. After converting the arrays to grayscale, Regions of Interest (ROI) are found to pinpoint thermal footprints.

**(2) Parameter estimation:** tracks and analyzes thermal changes over time to detect processing variations. Here, three parameters are derived from the images to characterize the temperature distribution within the Region of Interest (ROI). Initially, the ROI's thermal intensity distribution is calculated to capture the spatial probability distribution of different thermal values within the ROI. This distribution is summarized by computing the cumulative distribution function and fitting a linear regression line. The resulting parameters, namely the slope (m), intercept (c), and area under the distribution curve (a), represent the temporal changes in thermal image effectively.

**(3) PRINCE energy estimation:** PRINCE prediction models are built using classical regression and classification machine learning algorithms. The model is trained from extracted parameters using data from applications used in our experiments (See Section 3), each with distinct input data and processing requirements.

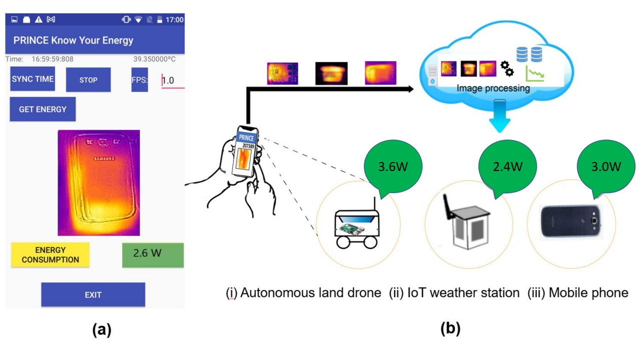


**Figure 2: PRINCE, a) Application b) Estimation process.**

## 3 PRINCE PROTOTYPE

**Prototype deployment:** Our prototype uses a mobile phone (CAT S60) with an integrated FLIR thermal camera (Client) and a Cloud /Edge service (Server) hosting the regression model for energy prediction. As a baseline, we use Monsoon Power Monitor for all of our experiments.

**PRINCE application:** We develop an Android application (Figure 2(a)) to take a thermal image and extract its thermal heat values (as arrays). These arrays depict temperature profile matrices in which a temperature value is associated with each pixel. Then, this data (thermal heat values) with the time stamp is uploaded to the server to get the prediction of the energy consumption (overall process is shown in Figure 2(b)).

**Demo procedure:** As a target device from which energy is estimated, we rely on a Samsung Galaxy S3. This phone has a detachable battery and we use it to verify our estimates against Monsoon power meter values. Besides this, it has been demonstrated that this type of hardware can be re-purposed to enable new applications, e.g., portable cloudlets [1]. We execute three distinct applications in the device (BobBall, Droidfish and Gobandroid) individually to measure their respective energy consumption. These applications were chosen due to their diverse complexity, for instance, Droidfish has the highest lines of code count while Gobandroid has the highest amount of classes. While the different applications were running on the phone, we captured thermal images of the back cover of the device using our PRINCE. To do this, once the PRINCE application started, the first task is to synchronize the time by pressing the "SYNC TIME" button. After, we could initiate the capturing process by pressing the "START" button. By pressing the "GET ENERGY" button, the energy consumption estimate is obtained.

## 4 DEMONSTRATION RESULTS

**Energy estimation and comparison to baseline:** Table 1 shows true power (monsoon power monitor) values and corresponding thermal values, preserving the relative order of application energy consumption: Droidfish ranks highest, followed by Bobball and Gobandroid. We also perform code analysis over the application to quantify the lines of code (LoC), classes (Classes) and cyclomatic complexity (CC). While it is possible to observe relative relations between these metrics and the thermal values, code execution changes during runtime, making it difficult to correlate thermal values to those accurately. For example, although Gobandroid has more classes than Bobball, it requires less computational processing. In contrast, it is possible to observe that overall estimates from the power monitor matches relatively to the estimates derived from the thermal images, suggesting that heat is a better indicator to estimate energy consumption. Figure 3 shows the predicted values from thermal images in real-time, demonstrating that PRINCE can be used to produce accurate and fast energy estimation of applications.

| Application | LoC | CC | Classes | Thermal Values | Predicted Power (mW) | True Power (mW) |
|---|---|---|---|---|---|---|
| *Bobball* | 1966 | 786 | 35 | 29533 | 1701 | 1433 |
| *Droidfish* | 30476 | 10375 | 161 | 29760 | 2610 | 2110 |
| *Gobandroid* | 605 | 205 | 37 | 29499 | 1200 | 1100 |

**Table 1: Estimated code metrics, thermal values, and power**

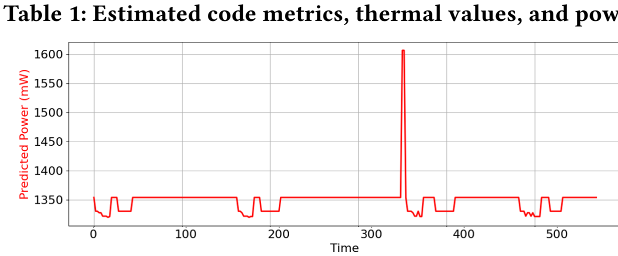


**Figure 3: Real-time power estimation performance**

**PRINCE estimation performance:** The regression performance in estimating energy show R-squared values around 0.81 and low root mean square error (RMSE) values of approximately 0.13. Bayesian Ridge, Bayesian ARD and Ridge show almost similar RMSE 0f 0.10.

## REFERENCES

[1] Farooq Dar et al. 2023. Upscaling Fog Computing in Oceans for Underwater Pervasive Data Science using Low-Cost Micro-Clouds. *ACM Transactions on Internet of Things* 4, 2 (2023), 1–29.

[2] Huber Flores et al. 2020. Estimating energy footprint using thermal imaging. *GetMobile: Mobile Computing and Communications* 23, 3 (2020), 5–8.