A Novel Easy-to-construct Power Model for Embedded and Mobile Systems
Using Recursive Neural Nets to Estimate Power Consumption of ARM-based Embedded Systems and Mobile Devices

Oussama Djedidi, Mohand A. Djeziri, Nacer K. M'Sirdi and Aziz Naamane
Aix-Marseille University, Université de Toulon, CNRS, LIS, SASV, Marseille, France

Keywords: Data Fitting, Embedded Systems, Modeling, NARX, Neural Nets, Power Consumption, Smartphone.

Abstract: This paper features a novel modeling scheme for power consumption in embedded and mobile devices. The model hereafter presented is built thought data fitting techniques using a NARX nonlinear neural net. It showcases the advantages of using a nonlinear model to estimate power consumption over the widely used linear regression models, where The NARX neural net is simpler, easier to implement, and more importantly more suitable as power changes are not always linear. Finally, experimental results validate the model with one with an accuracy of 97.1% on a smartphone.

1 INTRODUCTION

Power consumption in mobile and embedded devices occupies a lead role in research since these devices either come with limited resources (batteries) or have to be optimized for minimal consumption as is the case for the internet of things (IoT). The high interest in power consumption and estimation is justified by a multitude of reasons. First and foremost, power models allow the developers to estimate the remaining battery life (Kim et al., 2016) and estimate the energy consumption of their applications (Mittal et al., 2012). They also show developers how user and application actions and interactions translate into energy loss (Guo et al., 2017), allow for the said-developers to debug power consumption in the device (Hoque et al., 2015), and even identify energy hogs in the code of their applications (Banerjee et al., 2014). Moreover, by studying power consumption, researchers are able to create energy-friendly algorithms (Huang et al., 2017) and even more innovatively detect security issues and threats (Caviglione et al., 2016).

Researchers have been building models to estimate power consumption for over fifteen years, with each model focusing on either accurate power estimation (Gordon et al., 2011), or granularity (Google Inc., 2017), or the simplicity of implementation (Kim et al., 2012; Kim et al., 2014). Most of these models are repotorted in an extended litterature review (Hoque et al., 2015), in which Hoque et al. proposed a taxonomy for the recorded power profilers and models according to the measurement mechanisms, different inputs used to each type of model (utilization, event-driven and code), modeling philosophies (white-box vs black-box), profiling schemes (On or off-device), and granularity. The most prominent (and still available) among the studied are those built by system providers like Google's *Android Power Profiler* (Google Inc., 2017), *Trepn* and *Snapdragon Profiler* (Qualcomm Innovation Center,), and research published in the literature such as *PowerBooter* (Zhang et al., 2010), *Sesame* (Dong and Zhong, 2011), *DevScope* (Jung et al., 2012), *Eprof* (Pathak et al., 2012), and the models presented in (Banerjee et al., 2014) and (Shye et al., 2009).

Ahmad et al. also established a survey and essentially mentioned the same works. Nevertheless, it looked at the profilers through the lens of their implementation (software/hardware-based) (Ahmad et al., 2015).

Besides the works cited in the surveys, Kim et al. also proposed a polynomial model for power consumption modeling, built using linear regression (Kim et al., 2012). The authors later continued to enhance their model with better power estimation for the GPU (Kim and Chung, 2013), and the display (Kim et al., 2015). More recently, several new models were

Djedidi, O., Djeziri, M., M'Sirdi, N. and Naamane, A.
A Novel Easy-to-construct Power Model for Embedded and Mobile Systems - Using Recursive Neural Nets to Estimate Power Consumption of ARM-based Embedded Systems and Mobile Devices.
DOI: 10.5220/0006915805410545
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1*, pages 541-545
ISBN: 978-989-758-321-6

also published, either for the system as a whole, as it was the case for (Di Nucci et al., 2017a), (Shukla et al., 2016) and (Chowdhury and Hindle, 2016), or just the Central Processing Unit (CPU) like the models presented in (Yoon et al., 2017) and (Walker et al., 2017). Some of them even offered better accuracy like (Yoon et al., 2017) and proved that software profiling is practically as accurate and reliable as hardware profiling (Di Nucci et al., 2017b).

In a previous work, Djedidi et al. proposed a neural net model for power consumption (Djedidi et al., 2017). The model is a subsystem part of a general model to estimate several characteristics of a whole embedded system on chip (SoC). In their work, the authors focused on building a model with little overhead and capable of delivering estimations online with little lag. Although the model was quite accurate—With an accuracy of 95%—it still presented a lot of noise that didn't correspond to estimations (Djedidi et al., 2017).

In this work, we propose a new Artificial Neural Network (ANN) model to power consumption estimation that offers better results and keeps the estimation speed and the low overhead. The model is a nonlinear autoregressive model with exogenous inputs (NARX) neural net. As it will be showcased by the experimental results later in this work, the proposed model also manages to overcome the traditional shortcomings of the regression-based models which fall short when the power change is not linear.

The remainder of this work is structured as follows. In section 2, the model construction process is described alongside the choice of the type of the model and the inputs. Section 3 is used to describe the experimental setup, followed by the experimental results and a comparative study in section 4. Finally, the work ends with a conclusion of obtained results and future works.

2 MODEL CONSTRUCTION

Data-fitting techniques are quite useful when not all the dynamics are known in a system or are too complex to be modeled vis-à-vis the constraints at hand—computational cost, for instance. These techniques train the model so that its estimations would match the measured outputs. A lot of the models reported in the literature rely on black-box techniques, more precisely linear regression (Hoque et al., 2015), since white-box techniques can become quite complex (Pathak et al., 2012).

Regression-based models, as is the case for *PowerBooter* (Zhang et al., 2010), the models by Dong et al., (Dong and Zhong, 2011), Kim et al. (Kim et al., 2012), Kim et al. (Kim et al., 2015), Shukla et al. (Shukla et al., 2016), and Xu et al. (Xu et al., 2013) amongst others, are quite popular. However, these models assume—by definition—that the variations in power consumption are linear causing an increased estimation error when it is not the case (Hoque et al., 2015). In this work, we avoid the pitfalls of using a linear model by using a nonlinear model capable of accommodating nonlinear dynamics; The model we use is a NARX neural network (Xie et al., 2009).

Nonetheless, data-based models—Whether constructed through regression techniques or using machine learning—require no formal relation of interaction between the inputs and the estimated outputs (Zhang et al., 2010). They rely often upon observations and experimental analyses in various operating conditions for the choice of model inputs (Hoque et al., 2015), which—along with the ease of construction and implementation—explains their popularity.

2.1 Granularity and Input Selection

One of the major differences between the different power models in embedded and mobile systems is their granularity, which dictates the lowest level at which the model can estimate power consumption (Hoque et al., 2015). It is generally defined by the purpose behind the building of the model. For instance, in the models built to investigate the influence of elements of the code on power consumption, the granularity should be as fine as possible, at function level or even more (line of code). The model constructed in the work is an evolution and an improvement upon that constructed in (Djedidi et al., 2017). Hence, it aims to estimate and monitor power consumption of system components mainly those found in the SoC.

As shown in fig 1, the total amount of power consumed by the device can be broken down into the

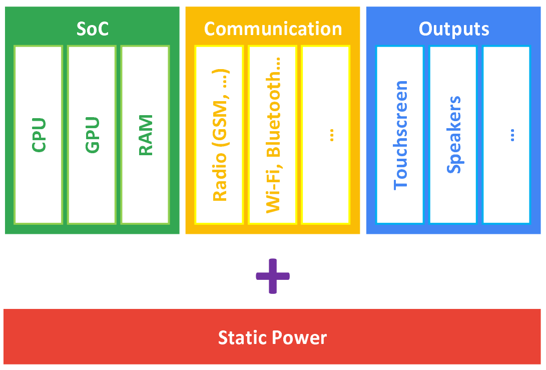

Figure 1: Power consumption distribution in a typical mobile or embedded system.

sum of powers consumed by each individual component (i.e. CPU, or Wi-Fi, ...) plus a minimum static amount of power that is always consumed by the device (Hoque et al., 2015) :

$$P_{Total} = P_{Static} + \sum_{i=1}^{n} P_i \tag{1}$$

i denotes the component number (CPU, GPU, RAM...), and n is the total number of these components. The power consumed each of these components can be correlated with one of its characteristics. These characteristics will be used as inputs to the power model.

To keep the model as simple as possible—and consequently as computationally optimized as possible, we chose to include only the most characterizing and directly influential inputs.

2.1.1 The SoC

In most of the embedded system, the SoC is composed of a CPU, the Random Access Memory (RAM), and a Graphics Processing Unit (GPU) when needed. CPU power consumption is a function of its frequency and voltage (Altamimi and Naik, 2015). Moreover, the voltage in processors with dynamic voltage and frequency scaling (DVFS) is also a function of frequency (Djedidi et al., 2017). Thus, the input used to estimate power consumption is the frequency f of each CPU core. Most works in the literature also use the load, but our experimental results showed no improvement with its inclusion, and we chose for optimization purposes to omit it. The same input is used for the GPU.

To account for the power used by the RAM, in the first trials we opted for the use of the value of the occupied RAM. However, that value on its own does not factor the maximum and the minimum possible values of the RAM on the system. Henceforth, we opt for a ratio of the occupied RAM over its maximum value and call it Memory occupation Rate (MOR).

2.1.2 Communication Peripherals

The communication peripherals are the phone's GSM chip, Wi-Fi (and Bluetooth), and the GPS. For the GSM chip, the inputs are the On-call status (ON/OFF), the signal strength, and the data transfer rate (download and upload). Similarly, for the Wi-Fi and Bluetooth, the inputs are the status and the data rate (download and upload). Finally, for the GPS, we used the only status.

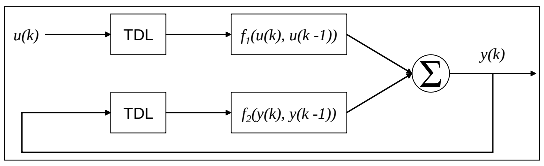

Figure 2: The general structure of the neural network model.

2.1.3 Output Peripherals

The output peripherals considered in this study is only the screen as a case study of the model. For the touchscreen, the status of the screen (ON/OFF) and its brightness are used as inputs. We consider that the power consumed by the touch layer of the screen is negligible (Kim et al., 2012).

2.2 The NARX Neural Net

NARX neural networkss are discrete-time nonlinear systems that are used to predict the output of time series (Xie et al., 2009). Mathematically, they are represented as follows :

$$\begin{aligned} y(k+1) = f\,[&y(k),\ldots,y(k-d_y+1) \\ &u(k-n),u(k-n-1),\ldots,u(k-n-d_u+1)] \end{aligned} \tag{2}$$

$y(k)$ and $u(k)$ are the output to be predicted and the input a the time step k, respectively. The term n is the process dead-time. While d_y and d_u are the orders of time delays for the input and output, also called input and output-memory. Since the power changes reacts pretty quickly with changes in the characteristics (inputs), d_y, d_u, and n were all set to the value of 1. In the previous paragraph, the input u was set to be:

$$\begin{aligned} u = [&f_1,\ldots,f_j,f_{GPU},MOR, \\ &GSM_{OnCall},GSM_{SignalStrength},GSM_{DataRate}, \\ &Wi-Fi_{Status},Wi-Fi_{DataRate},GPS_{Status}, \\ &Screen_{Status},Screen_{Brightness}] \end{aligned} \tag{3}$$

Where j is the total number of CPU cores in the system. The neural network is composed of two layers. The hidden layer contains n neurons whose activation functions are sigmoids, where n is the length of the input vector. The output layer contains a single neuron with a linear transfer function. Fig. 2 shows the general structure of the NARX model with the output feedback and the time delay line (TDL).

3 EXPERIMENTAL SETUPS

For running tests and undertaking experimental validation, we used an Android™ smartphone. These devices are very popular and developed application can be effortlessly transferred from one device to another. Moreover, the availability of the source code for their operating systems makes some of the needed parameters accessible for reading and even modification. Additionally, they are equipped with the all necessary sensors for the measurement of all the variables mentioned in paragraph 2.1.

The smartphone we used in this study equipped with an octa-core processor arranged in a big.LITTLE configuration (ARM Inc,), running at frequencies up to 2.3 GHz, a state of the art GPU, and 4 GB of RAM.

4 EXPERIMENTAL RESULTS

In figure 3, the measured power consumption of the Android™ phone is drawn against values estimated by the NARX neural network model. The estimations made by the model follow the measured values accurately with minimal estimation errors. It has an accuracy of 97.12%, and the recorded Mean Absolute Error (MAE) 0.0168 W, whereas the Mean Squared Error (MSE) is 7.04×10^{-4}. The model is hence validated with encouraginf results.

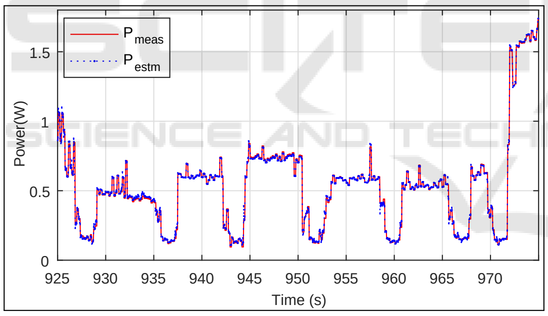

Figure 3: Power estimation by the neural network model against system measurements.

Going a step further, we compared our the accuracy of our results with established and recent power models. In our tests, the measured accuracies of *Trepn* and *Snapdragon Profiler* were 94.5% and 95% respectively. *PowerBooter* has a reported accuracy of 96% (Zhang et al., 2010), and *PETrA*'s accuracy is also reported to be the same(Di Nucci et al., 2017a), as is the case for *Eprof* (Hoque et al., 2015).

We also compared the power overhead caused by our modeling and monitoring program to the profilers we could test; *PowerBooter*, *Trepn*, and *Snapdragon Profiler*. The first caused an increase of 9% in power consumption during a 15 minutes test. While the second caused an increase of 8%, in a same scenario test. The *Snapdragon Profiler* caused an increase of 5%. Compared to the three aforementioned profilers, our modeling and monitoring scheme caused an increase of only 3% in power consumption, during the same test.

5 CONCLUSION

In this work, we have presented a novel power model for embedded and mobile devices. The new model is very simple to construct and train. It causes less overhead than existing ones and still performs at the speeds necessary to follow the power changes in the system. Finally and more importantly, it improves upon existing models in terms of accuracy with a mean absolute percentage error of 2.8%, one of the smallest in the reported literature.

This model will be incorporated in the in the incremental modeling scheme previously built by (Djedidi et al., 2017) to improve the estimation results. It will also be used to monitor power consumption in the device in future studies aiming to study the health status and operating state of the device.

ACKNOWLEDGEMENTS

This paper is a part of the ”FUI 19” project which is supported and funded by the Banque Publique d'Investissement (BPI), to whom we address our thanks.

REFERENCES

Ahmad, R. W., Gani, A., Hamid, S. H. A., Xia, F., and Shiraz, M. (2015). A Review on mobile application energy profiling: Taxonomy, state-of-the-art, and open research issues. *J. Netw. Comput. Appl.*, 58:42–59.

Altamimi, M. L. and Naik, K. (2015). A Computing Profiling Procedure for Mobile Developers to Estimate Energy Cost. In *Proc. 18th ACM Int. Conf. Model. Anal. Simul. Wirel. Mob. Syst. - MSWiM '15*, pages 301–305, New York, New York, USA. ACM Press.

ARM Inc. Technologies — big.LITTLE – Arm Developer.

Banerjee, A., Chong, L. K., Chattopadhyay, S., and Roychoudhury, A. (2014). Detecting energy bugs and hotspots in mobile apps. In *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng. - FSE 2014*, volume 3, pages 588–598, New York, New York, USA. ACM Press.

Caviglione, L., Gaggero, M., Lalande, J. F., Mazurczyk, W., and Urbański, M. (2016). Seeing the unseen: Revealing mobile malware hidden communications via energy consumption and artificial intelligence. *IEEE Trans. Inf. Forensics Secur.*, 11(4):799–810.

Chowdhury, S. A. and Hindle, A. (2016). GreenOracle: Estimating Software Energy Consumption with Energy Measurement Corpora. In *Proc. 13th Int. Work. Min. Softw. Repos. - MSR '16*, pages 49–60, New York, New York, USA. ACM Press.

Di Nucci, D., Palomba, F., Prota, A., Panichella, A., Zaidman, A., and De Lucia, A. (2017a). PETrA: A Software-Based Tool for Estimating the Energy Profile of Android Applications. In *2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion*, pages 3–6. IEEE.

Di Nucci, D., Palomba, F., Prota, A., Panichella, A., Zaidman, A., and De Lucia, A. (2017b). Software-based energy profiling of Android apps: Simple, efficient and reliable? In *SANER 2017 - 24th IEEE Int. Conf. Softw. Anal. Evol. Reengineering*, volume 15, pages 103–114. IEEE.

Djedidi, O., Djeziri, M. A., M'Sirdi, N. K., and Naamane, A. (2017). Modular Modelling of an Embedded Mobile CPU-GPU Chip for Feature Estimation. In *Proc. 14th Int. Conf. Informatics Control. Autom. Robot.*, volume 1, pages 338–345, Mardrid, Spain. SCITEPRESS - Science and Technology Publications.

Dong, M. and Zhong, L. (2011). Self-constructive high-rate system energy modeling for battery-powered mobile systems. In *MobiSys*, page 335, New York, New York, USA. ACM Press.

Google Inc. (2017). Measuring Power Values — Android Open Source Project.

Gordon, M., Zhang, L., and Tiwana, B. (2011). PowerTutor.

Guo, Y., Wang, C., and Chen, X. (2017). Understanding application-battery interactions on smartphones: A large-scale empirical study. *IEEE Access*, 5:13387–13400.

Hoque, M. A., Siekkinen, M., Khan, K. N., Xiao, Y., and Tarkoma, S. (2015). Modeling, Profiling, and Debugging the Energy Consumption of Mobile Devices. *ACM Comput. Surv.*, 48(3):1–40.

Huang, J., Li, R., An, J., Ntalasha, D., Yang, F., and Li, K. (2017). Energy-Efficient Resource Utilization for Heterogeneous Embedded Computing Systems. *IEEE Trans. Comput.*, 66(9):1518–1531.

Jung, W., Kang, C., Yoon, C., Kim, D. D., and Cha, H. (2012). DevScope: A Nonintrusive and Online Power Analysis Tool for Smartphone Hardware Components. *Proc. eighth IEEE/ACM/IFIP Int. Conf. Hardware/software codesign Syst. Synth. - CODES+ISSS '12*, page 353.

Kim, D., Chon, Y., Jung, W., Kim, Y., and Cha, H. (2016). Accurate Prediction of Available Battery Time for Mobile Applications. *ACM Trans. Embed. Comput. Syst.*, 15(3):1–17.

Kim, K., Shin, D., Xie, Q., Wang, Y., Pedram, M., and Chang, N. (2014). FEPMA: Fine-Grained Event-Driven Power Meter for Android Smartphones Based on Device Driver Layer Event Monitoring. In *Des. Autom. Test Eur. Conf. Exhib.*, pages 1–6, New Jersey. IEEE Conference Publications.

Kim, M. and Chung, S. W. (2013). Accurate GPU power estimation for mobile device power profiling. *Dig. Tech. Pap. - IEEE Int. Conf. Consum. Electron.*, pages 183–184.

Kim, M., Kong, J., and Chung, S. W. (2012). Enhancing online power estimation accuracy for smartphones. *IEEE Trans. Consum. Electron.*, 58(2):333–339.

Kim, Y. G., Kim, M., Kim, J. M., Sung, M., and Chung, S. W. (2015). A novel GPU power model for accurate smartphone power breakdown. *ETRI J.*, 37(1):157–164.

Mittal, R., Kansal, A., and Chandra, R. (2012). Empowering developers to estimate app energy consumption. In *Proc. 18th Annu. Int. Conf. Mob. Comput. Netw. - Mobicom '12*, page 317, New York, New York, USA. ACM Press.

Pathak, A., Hu, Y. C., and Zhang, M. (2012). Where is the Energy Spent Inside My App?: Fine Grained Energy Accounting on Smartphones with Eprof. In *Proc. 7th ACM Eur. Conf. Comput. Syst.*, EuroSys '12, pages 29–42, New York, NY, USA. ACM.

Qualcomm Innovation Center, I. Trepn Profiler - Android Apps on Google Play.

Shukla, N. K., Pila, R., and Rawat, S. (2016). Utilization-based power consumption profiling in smartphones. In *Proc. 2016 2nd Int. Conf. Contemp. Comput. Informatics, IC3I 2016*, pages 881–886. IEEE.

Shye, A., Scholbrock, B., and Memik, G. (2009). Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures. *Micro*, pages 168–178.

Walker, M. J., Diestelhorst, S., Hansson, A., Das, A. K., Yang, S., Al-Hashimi, B. M., and Merrett, G. V. (2017). Accurate and Stable Run-Time Power Modeling for Mobile and Embedded CPUs. *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, 36(1):106–119.

Xie, H., Tang, H., and Liao, Y. H. (2009). Time series prediction based on narx neural networks: An advanced approach. In *Proc. 2009 Int. Conf. Mach. Learn. Cybern.*, volume 3, pages 1275–1279. IEEE.

Xu, F., Liu, Y., Li, Q., and Zhang, Y. (2013). V-edge: fast self-constructive power modeling of smartphones based on battery voltage dynamics. In *nsdi'13 Proc. 10th USENIX Conf. Networked Syst. Des. Implement.*, pages 43–55. USENIX Association.

Yoon, C., Lee, S., Choi, Y., Ha, R., and Cha, H. (2017). Accurate power modeling of modern mobile application processors. *J. Syst. Archit.*, 81:17–31.

Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., Mao, Z. M., and Yang, L. (2010). Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proc. eighth IEEE/ACM/IFIP Int. Conf. Hardware/software codesign Syst. Synth. - CODES/ISSS '10*, page 105, New York, New York, USA. ACM Press.