

# EMPOWERING EDGE MINING ON SMARTPHONES WITH RECONFIGURABLE FABRICS

Zeyu Yan, Xiaowei Xu, Guangyu Yu and Hu Yu

School of Optical and Electronic Information, HUST, Wuhan, 430074, China

\*Corresponding Author's Email: bryanhu@hust.edu.cn

## ABSTRACT

With the prevalence of sensors in smartphones, a large volume of time series is being produced, which contains useful information about personal lifestyles and habitats. Considering privacy and security, the mining of such data is best done on the edge. However, the tight energy constraint imposed by batteries makes this task challenging. Recently many studies focus on edge mining of time series on low-power co-processors for high energy efficiency. However, none of them has explored the opportunity of reconfigurable fabrics, which are well-known energy-efficient and low-cost solutions. In this paper, we propose an efficient smartphone architecture to empower edge mining on smartphones. A configuration lib is associated with the fabrics which can be configured according to the current bottleneck of edge mining tasks. Two widely-adopted algorithms, artificial neural networks (ANNs) and Dynamic Time Warping (DTW) for edge mining, are evaluated. The proposed architecture is implemented on the off-the-shelf smartphones and low-power FPGAs. Experimental results show that compared with smartphones, the proposed architecture can achieve a speedup of 7.8x-21.1x and an energy efficiency improvement of 10.6x-27.1x.

## INTRODUCTION

Smartphone has been playing an indispensable role in every aspect of human life. Today's smartphone can have about one dozen of sensors in it. Thus, tens of billions of user-specific data records (or time series) are generated by these smartphones. This huge amount of time series makes it possible to create well-managed healthy lifestyles and can be exploited to discover hidden patterns including frequent activities, classification of physiological data, and clusters of mobile trajectories. Therefore, there is a growing need to mine the value of these personal time series.

Currently, digital signal processor (DSP) and graphical processing unit (GPU) have also been adopted to deal with the time series for better energy efficiency. Georgiev et al. [1] implemented a more complicated audio sensing task on off-the-shelf smartphones. This audio sensing task is well designed and optimized for a low-power DSP co-processor. Meanwhile, a few studies pay attention to efficient data mining on heterogeneous mobile architecture. Prakash et al. [2] achieved high energy efficiency with a static partitioning strategy to

execute application kernels across CPU and GPU cores.

However, more efficient reconfigurable fabrics such as Field Programmable Gate Array (FPGA) have not been well explored for smartphones. Reconfigurable fabrics can be configured to a variety of algorithms with high performance, which is very resource-efficient. Furthermore, the energy efficiency of reconfigurable fabrics is much higher than GPUs or DSPs.

In this paper, we propose an efficient smartphone architecture to empower edge mining on smartphones. An algorithm configuration library is associated with the fabrics which can be configured according to the recent bottleneck of edge mining algorithms. Particularly, we implement the widely-used dynamic time warping (DTW) [3] and Artificial Neural Networks (ANNs) [4] for edge mining of time series in the algorithm configuration lib. The results show that compared with smartphones, the proposed architecture has a speedup of 7.8x-21.1x and an energy efficiency improvement of 10.6x-27.1x.

## HARDWARE IMPLEMENTATION

### Architecture Overview

As shown in Figure 1, the proposed architecture consists of a commercially available smartphone, an interface and reconfigurable fabrics. The smartphone gets data from inner sensors or external networks, which will be processed by edge mining tasks of time series. The bottleneck computation module analyzes the current edge mining tasks and distributes the computations load to CPU and reconfigurable fabrics. The bottleneck computation module will send the data of computational task to the reconfigurable fabrics and the processing of the data will be accelerated by the accelerator configured according to the configuration lib.

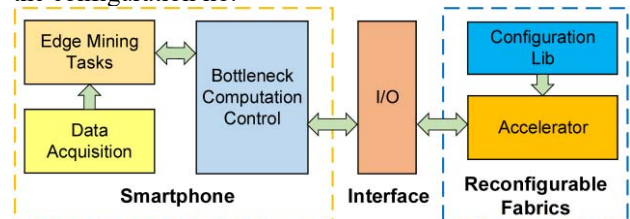


Figure 1: architecture with smartphone and reconfigurable fabrics

### ANN Accelerator

As shown as Figure 2, we implement a pipelined ANN accelerator including a Read/Write (R/W) controller, an ANN controller, a processing element (PE) array, a

sigmoid memory and a weight memory. The sigmoid module is responsible for the calculation of sigmoid function, and the weight array is used to store the parameters of ANNs. The PE array performs computations in ANNs. The R/W controller handles data reading and writing through the interface.

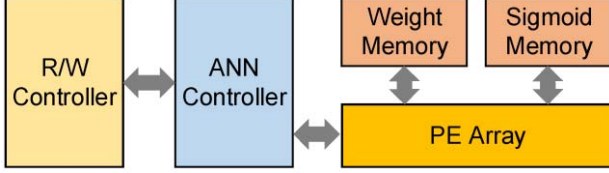


Figure 2: Overall structure of ANN accelerator on reconfigurable fabrics

The PE array module consists of several PEs, and the computing process of PE array is shown in Figure 3. As multiplication is very intense for ANNs and the number of multiplier (MUL) is often limited in FPGAs, a time-division multiplexing strategy for the multipliers is used to compute ANNs with the minimum resource.

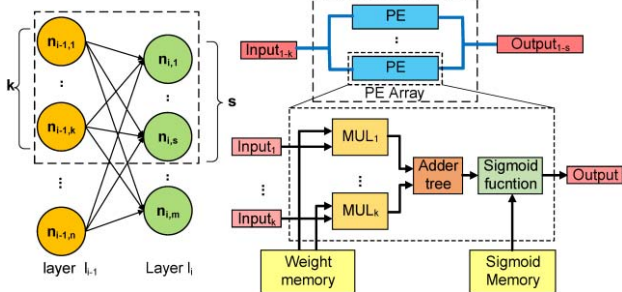


Figure 3: PE structure and its computation processes of ANN

### DTW Accelerator

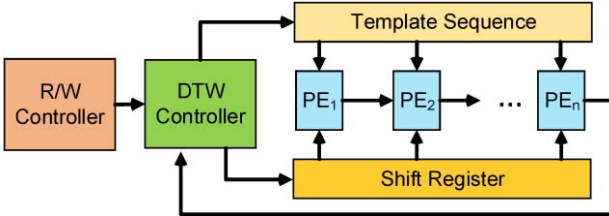


Figure 4: Overall structure of DTW accelerator

As shown in Figure 4, the DTW accelerator processes data streams with no pauses. The length of template sequence is equal to the number of PEs. The Manhattan distance is selected to be the distance function between two data elements. The DTW controller saves the template sequence in registers and there is a one-to-one correspondence between the data in template sequence and PE. The test data streams are sent to a shift register which will be fetched by PEs sequentially.

## EXPERIMENTS

### Experiment Setup

We select a smartphone, Meizu MX4 for the proposed

TABLE I. HARDWARE IMPLEMENTATIONS FOR ANNS AND DTW

FPGA	Algorithm	Resource Utilization (LEs)	Max frequency
Cyclone IV (EP4CE22)	ANN	7012/22320	56.05 MHz
IGLOO nano (AGLN020)	DTW	5896/6144	27.87 MHz

architecture. The USB2.0 chip FX2LP and two low-power FPGAs: EP4CE22 and AGLN020 are adopted for the experiment. The tested ANN is a  $4 \times 4 \times 3$  neural network, while the template length and the number of PEs in DTW test are 10, and the resource usage is shown in Table I.

The dataset used for ANNs and DTW are from UCI machine learning repository [5] and UCR time series classification archive [6].

### Experiment Result

As shown in Table II, it can be discovered that the power consumption of the smartphone (0.821W-1.066W) is slightly larger than that of our architecture (0.528W-0.776W). However, our architecture has a better performance which is around 8x to that of the smartphone implementation. Thus, our architecture has an overall energy efficient improvement of 10.57x-14.8x compared with the smartphone.

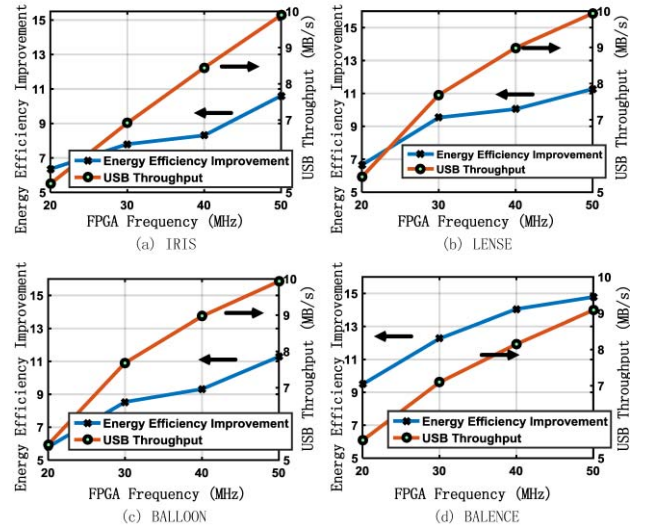


Figure 5: Energy efficient improvement and USB throughput of ANN.

We also discuss the frequency influence of FPGA on power consumptions and USB throughputs. As shown in Figure 5, the throughput of USB and energy efficiency improvement grow as the frequency increases in general. The maximum USB throughput is about 10 MB/s which is below the limit of USB transport capability of the smartphone.

TABLE II. SPEEDUP AND ENERGY CONSUMPTION OF ANN

Dataset	Performance (million items/second)		Energy (Watt)		Speedup	Energy Efficiency Improvement
	Smartphone	Our Architecture	Smartphone	Our Architecture		
IRIS	0.171	1.414	0.821	0.642	8.3x	10.6x
LENSE	0.173	1.420	1.066	0.776	8.2x	11.3x
BALANCE	0.167	1.298	1.006	0.528	7.8x	14.8x
BALLONS	0.167	1.419	0.842	0.635	8.5x	11.3x

TABLE III. SPEEDUP AND ENERGY CONSUMPTION OF DTW

Dataset	Performance (million items/second)		Energy (Watt)		Speedup	Energy Efficiency Improvement
	Smartphone	Our Architecture	Smartphone	Our Architecture		
50words	0.087	1.839	0.627	0.490	21.1x	27.1x

As to DTW accelerator, we first perform dimension reduction to preprocess the dataset with the widely-used piecewise constant models [7]. With the model, the template length in the dataset is normalized to 10, as it is equal to the number of PEs in the accelerator. Note that the template length can be changed for specific applications. The performance and energy efficiency are shown in Table III. Just like the results for ANNs, the power consumption of the smartphone (0.627W) is slightly larger than that of our architecture (0.490W), and the speedup and energy efficiency improvement are 21.14x and 27.05x, respectively.

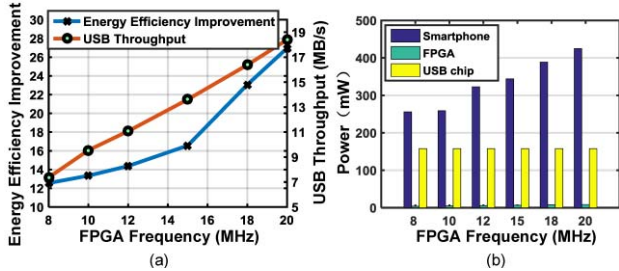


Figure 6: (a) Energy-efficient improvement and USB throughput, and (b) power consumption component of DTW.

Energy efficient improvement and USB throughput with the FPGA frequency are discussed as shown in Figure 6. We can notice that energy efficient improvement and USB throughput are almost positively correlated with the FPGA frequency. However, the throughput of USB is about 18.4MB/s which is close to the USB transport capability (20MB/s), which means we can not get a higher energy efficient improvement by simply increasing the FPGA frequency in our architecture. Meanwhile, the power consumption of FPGA is only a small portion of the whole power consumption of this architecture.

## CONCLUSIONS

In this paper, we propose a smartphone architecture, TIGER by embedding a low-power FPGA to existing

smartphone architecture for efficient edge mining of time series on smartphones. The results show that compared with smartphones, the smartphone architecture has a speedup of 7.8x-8.5x and an energy efficiency improvement of 10.6x-14.8x for accelerating ANNs, while 21.1x and 27.1x for accelerating DTW algorithm. Meanwhile, experiments show that a better performance can be achieved by adopting a more efficient I/O interface instead of USB

## REFERENCES

- [1] P. Georgiev, N. D. Lane, K. K. Rachuri, and C. Mascolo. Dsp. ear: Leveraging coprocessor support for continuous audio sensing on smartphones. In Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, pages 295–309. ACM, 2014.
- [2] Prakash, Alok, et al. "Energy-efficient execution of data-parallel applications on heterogeneous mobile platforms." Computer Design (ICCD), 2015 33rd IEEE International Conference on. IEEE, 2015.
- [3] G. Cormode. Fundamentals of analyzing and mining data streams. In Tutorial at Workshop on Data Stream Analysis, Caserta, Italy, 2007.
- [4] Ronao C A, Cho S B. Human activity recognition with smartphone sensors using deep learning neural networks[J]. Expert Systems with Applications, 2016, 59: 235-244.
- [5] A. Asuncion and D. Newman. Uci machine learning repository, 2007.
- [6] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The ucr time series classification archive. URL [www.cs.ucr.edu/~eamonn/time\\_series\\_data](http://www.cs.ucr.edu/~eamonn/time_series_data), 2015.
- [7] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. ACM Sigmod Record, 30(2):151–162, 2001.