ECG Authentication Neural Network Hardware Design with Collective Optimization of Low Precision and Structured Compression

Sai Kiran Cherupally, Gaurav Srivastava, Shihui Yin, Deepak Kadetotad, Chisung Bae†, Sang Joon Kim†, and Jae-sun Seo
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, USA
†Samsung Advanced Institute of Technology, Suwon, Korea

***Abstract*—For wearable devices that monitor personal health, secure access to private medical data becomes a crucial feature. Nowadays, device authentication based on biometrics such as fingerprint or iris has become increasingly popular. In this work, we investigate using electrocardiogram (ECG) signals as the biometric modality for device authentication, and we present accurate and low-power ECG-based authentication hardware. Deep neural networks (DNNs) have been employed with a cost function that maximizes inter-individual distance and minimizes intra-individual distance over time. During DNN training, we also introduce joint optimization of low-precision and structured sparsity, so that the real-time authentication hardware can consume minimal energy and area. Experimental results of custom hardware designed in 65nm LP CMOS technology exhibit low power consumption of 59.4 µW for real-time ECG authentication with a low equal error rate of 1.002% for a large 741-subject in-house ECG database.**

Keywords—ECG; authentication; deep neural network; structural sparsity; low-power hardware

I. INTRODUCTION

An increasingly large number of wearable devices are being introduced to the commercial market and being used by individuals during daily activities. Many wearable devices feature health-monitoring functionalities by integrating various physiological sensors, including electrocardiogram (ECG), photoplethysmogram (PPG), bio-impedance, etc. For instance, a number of recent wearable devices [1-2] embed single-lead ECG sensors for continuous cardiac monitoring capabilities.

Since a number of wearables already possess ECG sensors, researchers have been exploring to use ECG as a new modality for biometric authentication [3] towards enhanced security. Since the health-monitoring wearables contain private medical data, it becomes highly important to ensure secure access to such devices, so that adversaries cannot access private data.

Compared to other conventional biometric modalities (e.g. fingerprint, iris, face, voice, etc.), ECG-based authentication is advantageous with regards to inherent detection of liveness and the difficulty to be easily spoofed. Several prior works have presented low-power hardware designs for ECG authentication [4-7]. The authors of [4] implemented a deep neural network (DNN) based ECG authentication algorithm on FPGA, but consumed 1 MB of memory and 256 mW of power. In [5], a cross-correlation based ECG authentication algorithm was implemented on the ARM Cortex-M microcontroller in a wearable watch. However, both works evaluated the authentication accuracy only on relatively small databases (90 subjects for [4] and 28 subjects for [5]). The ECG authentication ASIC hardware presented in [6] was benchmarked on a large database of 645 subjects, but the reported equal error rate (EER) was 1.7%, which is quite higher than the EER values of recent fingerprint-based (0.8% [7]) and iris-based (0.82% [8]) authentication algorithms. In [9], an improved EER of 0.85% was reported for ECG authentication by using a DNN algorithm that maximizes the distance of ECG features for different individuals, but only software implementation was reported.

In this paper, we investigate a low-power design of ECG-based authentication hardware implementation adopting the DNN algorithm in [9]. To fit in costly fully-connected DNN within the power and area envelopes of wearable devices, we further incorporated both low-precision quantization and structured sparsity optimization in the overall DNN training process. Then, we implemented the ECG signal processing and the compressed and low-precision DNN in 65nm LP CMOS. For real-time ECG authentication, 59.4 µW power consumption at 1.2V was measured from simulation, and 1.002% equal error rate (EER) was achieved for an in-house 741-subject large database.

II. ECG SIGNAL PROCESSING AND DNN DESIGN

The raw digitized ECG signals go through signal processing and DNN tasks, which extracts the optimal ECG features that are used for ECG-based authentication. Fig. 1 shows the top-level diagram of the this work.

A. ECG Signal Processing

Signal processing of raw ECG beats sequentially goes through the steps of frequency domain filtering, detection of R-

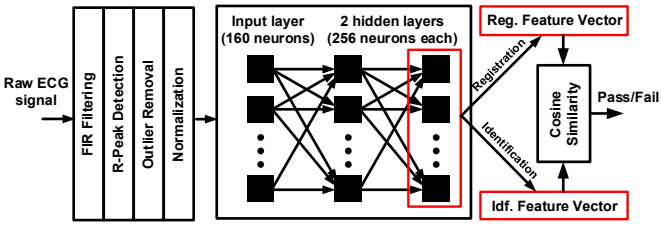

Fig. 1. Signal processing and DNN based ECG feature extraction for ECG-based authentication.

peak, outlier detection/removal, and normalization. The system can be programmed to perform authentication with either 8 beats for faster operation and 30 beats for more stable operation.

Frequency domain filtering: A single-lead sensor acquires raw ECG signals, which are digitized at 250 Hz sampling rate. The raw ECG signal goes through a 40-tap high pass filter, a 42-tap band pass filter, a differentiator and an 11-tap low pass filter.

R-peak detection: The outputs of aforementioned frequency domain filters are buffered in four successive 64-ECG-sample windows. The final low pass filter output is compared against a dynamic threshold [10], such that only valid R-peak points of ECG signals will be detected. With valid R-peak detection, 160 ECG samples that are aligned at the detected R-peak are saved.

Outlier detection/removal: As the valid ECG segments are collected, outlier ECG beats are detected using techniques reported in [11]. Outlier ECG beats that occur either due to temporal variability of ECG signals, abnormal sensor contact, or abrupt movement will be detected and removed, so that only similar ECG beats will be averaged and form a representative ECG beat for each individual [6]. The maximum, minimum values and the cosine distance of every extracted beat is compared with the mean maximum, minimum values and cosine distance of the collected beats. A beat with at least 50% variation in any of these comparisons is detected as an outlier and removed.

Normalization: The filtered and segmented ECG data is normalized before it is conveyed to the ensuing DNN. Every ECG segment is normalized by its mean and standard deviation, which is then additionally normalized by a global mean and a global standard deviation of all data from the ECG database.

B. DNN Training with Authentication-Specific Cost Function

Typically, DNN training uses one-hot coding for labeled outputs, where only the specific neuron output that correlates to the given input is labeled as "1" and all other neuron outputs are labeled as "0". As was done in [9], the extracted features are obtained at the last hidden layer, instead of the output layer, and this will be used for our ECG authentication. We compute and evaluate the cosine similarity between the registered ECG features and the identification ECG features, and wearable device access will be granted only when the cosine similarity is higher than a threshold value. We denote cosine similarity of two ECG feature vectors FV_1 and FV_2 as:

$$sim_{\cos} = \frac{FV_1 \cdot FV_2}{\|FV_1\|_2 \|FV_2\|_2} . \quad (1)$$

Cosine distance (CD) is defined as:

$$d_{\cos} = 1 - sim_{cos} . \quad (2)$$

In order to achieve low authentication error, the overlap between the intra-subject and inter-subject CD must be minimized. The one-hot labels typically used for DNNs are not most suitable for this specific purpose, due to the unawareness of how the extracted ECG features obtained from the hidden layers will be employed. For that purpose, we adopt the DNN algorithm with authentication-specific cost function in [9]. The authentication-specific cost function is:

$$cost = -\frac{\mu_{intra} - \mu_{inter}}{\sigma_{intra} + \sigma_{inter}} , \quad (3)$$

where μ_{intra} / μ_{inter} are means of intra-/inter-subject cosine similarity distributions, and σ_{intra} / σ_{inter} are standard deviations of intra-/inter-subject cosine similarity distributions.

If the threshold for authentication is set at $\theta = \frac{\mu_{inter}\sigma_{intra} + \mu_{intra}\sigma_{inter}}{\sigma_{inter} + \sigma_{intra}}$, it has been shown that larger relative distance will result in smaller EER [9]. Therefore, by minimizing the cost function, we can maximize the relative distance and minimize the EER.

We trained a DNN with two hidden layers of 256 neurons, where a simple network is judiciously chosen to fit under the ultra-low-power budget of wearable devices. Rectified linear unit (ReLU) activation function is used after both hidden layers. During each batch of the DNN training, μ_{inter}, μ_{intra}, σ_{inter} and σ_{intra} values are estimated. A relatively large batch size of 2,000 is used to have a sufficient number of pairs of intra-class and inter-class examples. We employed dropout in the first hidden layer with 0.1 dropout ratio. The 256-element vector output of the DNN is fed to the cosine similarity loss block. The cosine similarity block in Fig. 1 computes the cost function in Eq. (3), which is used to train the DNN using back-propagation with stochastic gradient descent.

C. DNN Training with Low-Precision and Structured Sparsity

Together with the aforementioned cost function, DNN training in this work also collectively optimizes structured compression and low-precision representation of weights.

Coarse-grain sparsity (CGS) [12] is a technique to generate structured sparsity by randomly dropping blocks of DNN throughout training. CGS block size and the compression ratio determine the level of sparsity in the trained DNN. Dropping blocks of weights (instead of pruning individual weights) has the advantage of minimizing index storage overhead and allowing the compressed DNN weights to be efficiently mapped onto regular SRAM arrays.

On the low-precision aspect, prior works have shown that low precision DNNs can substantially reduce the storage and communication while maintaining the accuracy. BinaryConnect [13] introduced DNN training techniques that can binarize the weights without affecting accuracy, while other works reported that low-precision weights such as 2-bit or 4-bit can lead to the optimal trade-offs in energy and accuracy [14-15].

In this work, we have jointly optimized CGS sparsity together with low-precision weight quantization during DNN training, for the ECG authentication task. Weight blocks are randomly dropped before training and throughout the training process. Various CGS blocks sizes (4×4, 8×8 and 16×16) and

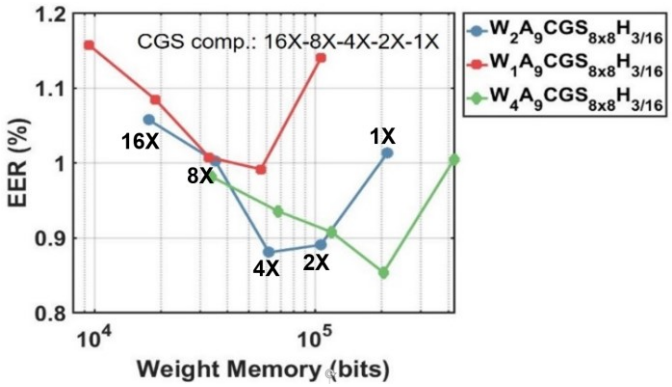

Fig. 2. Joint optimization of structured sparsity and low precision for the DNN for ECG authentication. CGS ratios for the points in each line are 16X-8X-4X-2X-1X from left to right.

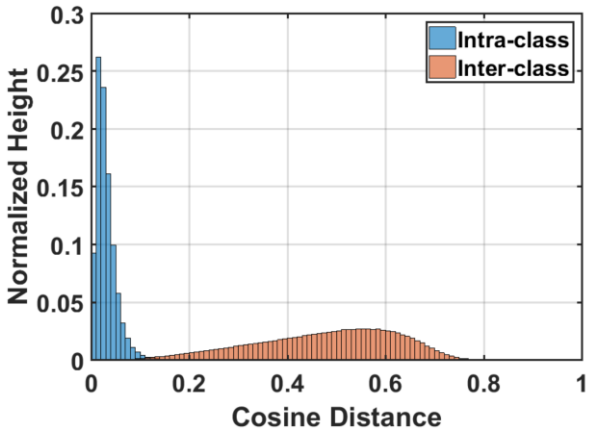

Fig. 3. Cosine distance distribution of intra-subject and inter-subject feature vectors, for 2-bit weight precision and 8X compression.

compression ratios (16X-8X-4X-2X-1X) are evaluated to determine optimal CGS configuration. Compression ratios 16X-8X-4X-2X-1X correspond to 10%-20%-30%-40%-100% and 6.25%-12.5%-25%-50%-100% sparsity in weight blocks (100% means no compression) for first hidden layer weight matrix and second layer weight matrix, respectively.

CGS-based DNNs are trained using back propagation, also with low-precision representation for DNN weights. Using the BinaryConnect technique [13], during the forward phase of training, we quantize high-precision weights and activations. During the backward phase, gradients of cost function are computed from output to input layer, and straight-through estimator [15] is used to estimate the gradient for quantized activations. During the weight update phase, the high precision weights are updated only for CGS blocks of non-zero weights.

Normalized DNN input is quantized to 6-bit precision and activations are quantized to 9-bit precision without EER degradation. DNN accuracy is further analyzed for different weight precisions of 1, 2, 4, 8 and 32 bits.

III. Software Results on Benchmarks

The in-house ECG database we used for benchmarking this work includes 741 subjects. Single-lead ECG acquisition procedure was followed for collecting the raw ECG data in this database, since our focus was on wearable devices. The single-channel (right arm cathode to left arm anode) ECG data for each subject has been acquired by analog front end (AFE) chip ADS1292R by TI at 250 Hz with 15-bit resolution. We trained DNNs with the authentication-specific cost function and joint CGS/low-precision, as described in Section II. We have separated the training and testing datasets. The training dataset consists of 18,306 beats (15-30 beats per subject) and the testing dataset consists of 52,849 beats (38-88 beats per subject).

Starting from floating-point precision, we swept a number of low-precision representations as well as structured sparsity schemes from dense to sparse designs. Fig. 2 shows the corresponding EER values with low-precision weights and CGS compression. Data point $W_2A_9CGS_{8x8}H_{3/16}$ represents 2-bit weight (W_2) and 9-bit activation (A_9) precision with CGS block size of 8x8. H is the range of weights used for the network. $H_{3/16}$ means that quantized weights are in the range [-3/16, +3/16]. It can be seen that CGS-compressed network with 2-bit weight precision leads to similar EER values the as 4-bit weight network for most of the CGS ratio settings, with half of the weight memory. Reducing weight precision to 1 bit hurts the EER significantly. The DNN with 2-bit weights results in ~1% EER even with CGS compression of 8X.

Using the trained DNN, in the testing phase, the feature vectors (FVs) are obtained from the final hidden layer output. We evaluate the feature extraction performance by examining the CD distributions for inter-subject and intra-subject FVs. To mitigate the time-variant nature of the ECG beats, we average eight extracted FVs obtained from the DNN with consecutive ECG beats to obtain a single representative FV. Fig. 3 shows the CDs for the trained DNN with 2-b weight precision and CGS compression of 8X.

IV. Hardware Implementation Results

We designed custom ECG authentication hardware in 65nm LP CMOS, including signal processing modules and the compressed, low-precision DNN. Supply voltage of 1.2V is used and the overall design was synthesized at 10 kHz for real-

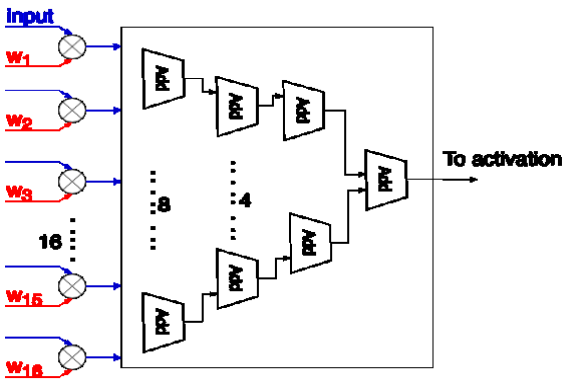

Fig. 4. Hardware implementation of a single neuron using fixed-point arithmetic units.

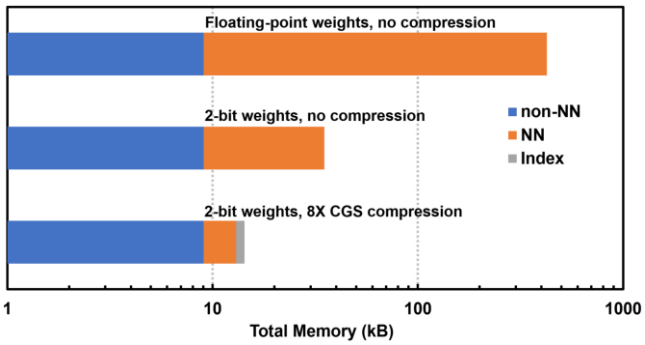

Fig. 5. Neuron network memory reduction aided by joint optimization of low-precision and structured compression (x-axis in log-scale).

time ECG authentication with extensive clock gating. DNN weights are stored in SRAM arrays, which are generated from a commercial memory compiler. EER and latency results are obtained from post-synthesis simulation using the in-house 741-subject ECG dataset. Power results are obtained from Synopsys Primetime PX using data activity of post-synthesis simulation.

A. DNN Hardware Implementation

The CGS-based sparse DNN that we implemented employs a pipelined datapath with synchronous clocking. We selected 2-bit weight precision for the DNN based on the results from Section III, which incurs negligible EER degradation compared to higher precision schemes. Due to the structured sparsity of the compressed DNN, the number of multiplications and additions are fixed to 32 per neuron, as shown in Fig. 4. A sparsely connected DNN containing 160 input neurons and two hidden layers with 256 neurons (ReLU activation) was implemented in hardware. With the CGS structure, the connectivity matrix of each layer is partitioned into blocks of 8×8 and four non-zero blocks are selected in each column and row. Thus, we only store 32 weights per neuron and the index of selected blocks.

In the DNN hardware design, one neuron in the first hidden layer is evaluated in each clock cycle, and all the neurons in the second hidden layer are computed simultaneously in each clock cycle. Thus, by the time all the neurons in the first hidden layer are evaluated, only one additional clock cycle is required to obtain the final output of the network. The same fixed-point arithmetic modules were used to sequentially evaluate the neurons in the first hidden layer. The latency of the DNN for generating one feature vector is 262 clock cycles.

B. EER, Area and Power Results

TABLE I. ECG PROCESSOR POWER/AREA BREAKDOWN

Module	Power (µW)	Area (mm^2)
Pre-processing + Cosine Similarity	17.0	0.5
NN Logic	0.5	0.03
SRAMs (Pre-processing / NN)	28.0 / 13.9	0.04 / 0.02
Total	**59.4**	**0.59**

The total ECG processor area is 0.59 mm^2. The power and area breakdown is summarized in Table I. Compression and

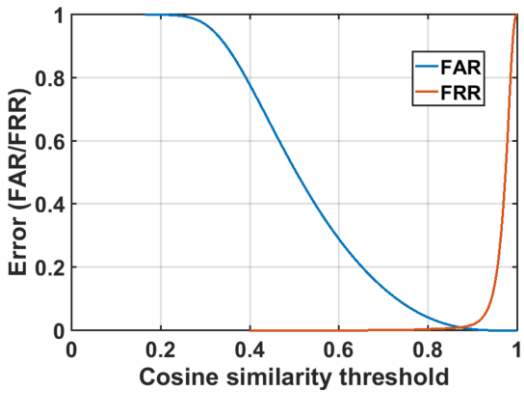

Fig. 6. False acceptance rate (FAR) and false rejection rate (FRR) plots are shown. EER is the error value when FAR equals FRR.

low-precision optimization substantially reduced the area and power consumption of the SRAM (39.6 µW out of 41.9 µW is leakage power due to low frequency) and neural network logic. Fig. 5 shows the total memory reduction aided by low precision (2-b) weights (16X) and structured sparsity optimization (~6X). Compared to the uncompressed fully-connected DNN used in [9] with 32-bit floating-point precision, the total DNN memory is reduced by 104X, with minimal EER degradation of 0.15%.

TABLE II. COMPARISON TO PRIOR WORK

Work	Power (µW)	Memory (kB)	# of subjects in database	EER (%)
[6]	50.4 (@0.8V, 10kHz)	64	645	1.7
This work	**59.4 (@1.2V, 10kHz)**	**14.25**	**741**	**1.002**

Fig. 6 shows the false acceptance rate (FAR) and false rejection rate (FRR) plots for the proposed ECG authentication hardware. Low EER (when FAR equals FRR) of 1.002% is achieved. Table II shows a comparison to previous ECG authentication ASIC hardware [8] for EER, power and memory. The proposed ECG processor design with a new DNN and joint-optimization of precision and compression considerably improves both memory footprint and EER for a larger database of 741 subjects, while consuming similar power at nominal supply voltage. Further power reduction is possible with dynamic voltage scaling.

V. CONCLUSION

In this paper, we investigated ECG-based authentication hardware employing a new cost function and collective optimization of low-precision and compression during DNN training. The corresponding hardware was implemented in 65nm LP CMOS, demonstrating low EER of 1.002% and low power of 59.4 µW for real-time ECG authentication.

ACKNOWLEDGEMENT

This work is in part supported by NSF grant 1652866, Samsung Advanced Institute of Technology, and C-BRIC, one of six centers in JUMP, a SRC program sponsored by DARPA.

REFERENCES

[1] AliveCor, "KardiaBand," https://store.alivecor.com.

[2] Samsung, "Simband," https://www.simband.io.

[3] S. A. Israel *et al.*, "ECG to identify individuals," *Pattern Recognition*, vol. 38, no. 1, pp. 133–142, 2005.

[4] A. Page *et al.*, "Utilizing deep neural nets for an embedded ECG-based biometric authentication system," *IEEE BioCAS*, 2015.

[5] S. J. Kang *et al.*, "ECG authentication system design based on signal analysis in mobile and wearable devices," *IEEE Signal Prcessing Letters*, vol. 23, no. 6, pp. 805-808, June 2016.

[6] S. Yin *et al.*, "A 1.06 µW smart ECG processor in 65 nm CMOS for real-time biometric authentication and personal cardiac monitoring," *IEEE Symp. on VLSI Circuits*, 2017.

[7] K. K. M. Shreyas, S. Rajeev, K. Panetta, and S. S. Agaian, "Fingerprint authentication using geometric features," *IEEE International Symposium on Technologies for Homeland Security (HST)*, April 2017, pp.1–7.

[8] A. Uka, A. Roçi, and O. Koç, "Improved segmentation algorithm and further optimization for iris recognition," *IEEE EUROCON - 17th International Conference on Smart Technologies*, July 2017, pp. 85–88.

[9] S. Yin *et al.*, "Designing ECG-based physical unclonable function for security of wearable devices," *IEEE EMBC*, 2017.

[10] C. Choi, *et al.*, "A PD control-based QRS detection algorithm for wearable ECG applications," *IEEE EMBC*, 2012.

[11] A. Lourenco, *et al.*, "Outlier detection in non-intrusive ECG biometric system," *Int. Conf. on Image Analysis and Recognition*, 2013.

[12] D. Kadetotad *et al.*, "Efficient memory compression in deep neural networks using coarse-grain sparsification for speech applications," *IEEE/ACM ICCAD*, 2016.

[13] M. Courbariaux *et al.*, "BinaryConnect: Training deep neural networks with binary weights during propagations," *NIPS*, 2015.

[14] B. Moons *et al.*, "Minimum Energy Quantized Neural Networks," *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2017.

[15] S. Yin *et al.*, "Minimizing Area and Energy of Deep Learning Hardware Design Using Binarization and Structured Compression," *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2017.