Accelerating Dynamic Time Warping With Memristor-Based Customized Fabrics

Xiaowei Xu, *Student Member, IEEE*, Feng Lin, *Member, IEEE*, Aosen Wang, *Student Member, IEEE*, Xinwei Yao, *Member, IEEE*, Qing Lu, Wenyao Xu, *Member, IEEE*, Yiyu Shi, *Senior Member, IEEE*, and Yu Hu, *Member, IEEE*

***Abstract*—The rapid development of Internet of Things is yielding a huge volume of time series data, the real-time mining of which becomes a major load for data centers. The computation bottleneck in time series mining is the distance measure, in which dynamic time warping (DTW) is one of the most widely used distance measures. Recently, various software optimization and hardware acceleration techniques have been proposed for DTW acceleration. However, the throughput and energy efficiency of DTW are still big concerns considering the ever-increasing volume of times series. In this paper, we propose a high-throughput and efficient memristor-based DTW architecture for real-time time series mining on data centers. Specifically, memristors have been adopted for both computation and configuration of the computing architecture. The computation flow in this architecture is fully presented in a continuous and asynchronous manner. To improve the computation efficiency, we propose an early lower bound algorithm by exploiting the predictability in the circuit characteristic. Experiments are performed with module evaluation and end-to-end evaluation including three popular applications: 1) similarity search; 2) classification; and 3) anomaly detection. Experimental results indicate that, compared to existing approaches, the speedup and energy efficiency improvement are 12×–43× and 51×–287×, respectively.**

***Index Terms*—Data mining, dynamic time warping (DTW), energy efficiency, high throughput, memristor, time series.**

Manuscript received January 27, 2017; revised April 25, 2017 and June 18, 2017; accepted June 21, 2017. Date of publication July 19, 2017; date of current version March 29, 2018. This paper was recommended by Associate Editor S. Pasricha. *(Corresponding author: Yu Hu.)*

X. Xu is with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: xxu8@nd.edu).

F. Lin, A. Wang, and W. Xu are with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260-2500 USA (e-mail: flin28@buffalo.edu; aosenwang@buffalo.edu; wenyaoxu@buffalo.edu).

X. Yao is with the Department of Computer Science and Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xwyao@zjut.edu.cn).

Q. Lu and Y. Shi are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: qlu2@nd.edu; yshi4@nd.edu).

Y. Hu is with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: bryanhu@hust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2729344

I. Introduction

ENERGY efficiency of data centers has been a primary focus in the past a few years due to their excessive power consumption. On the other hand, the load on data centers keeps increasing with the explosion of information technologies. It has been predicted that a major portion of the load will come from Internet of Things (IoT), which will yield over 4.4 zettabytes (5.5×10^{21} Bytes) of time series data by 2020 [6]. These time series data are transmitted to data centers for real-time mining [20]. It is therefore of utmost interest to explore techniques that handle time series data in real-time with high energy efficiency.

Distance measure between time series plays an important role in time series mining, which is the foundation of higher data mining tasks, such as classification and similarity search. Dynamic time warping (DTW) is one of the best distance measures according to a recent comparison study of several distance measures with 44 datasets [13]. It is widely used in different fields, such as speech recognition, financial analysis, and network traffic monitoring [12]. However, DTW has a quadratic time complexity, which is computation-expensive for huge data processing in data centers.

DTW has been well optimized with software and hardware methods to solve the obstacle of computation complexity. Lower bound (LB) [14], [25] is a powerful optimization method, which can prune a lot of sequences in many tasks such as similarity search and classification. Early abandon [37] is also very effective. Rakthanmanon *et al.* [37] cascaded multiple stages of software optimizations, which is considered as the most powerful software implementation for similarity search to date. DTW in a streaming manner is proposed by Sakurai *et al.* [41], which achieves a linear time complexity, however, allows false dismissals. The majority of these techniques achieve speedup by reducing the number of the invoked times of DTW rather than accelerating DTW itself. However, DTW calculation still accounts for about 80% of the total executing time [48], [51].

Meanwhile, customized hardware is adopted for further acceleration. Sart *et al.* [42] proposed a highly pipelined architecture for DTW on field-programmable gate arrays (FPGAs), which can compute DTW in linear time. Wang *et al.* [48] implemented a high-throughput DTW framework for similarity search on FPGAs. The proposed structure of processing element (PE) ring exploits the fine-grained parallelism of DTW

and achieves a significant speedup. Hardware acceleration of DTW has also been implemented on graphic processing units (GPUs) [19], [42]. In order to improve energy efficiency, Lotfian and Jafari [32] implemented an application specific integrated circuit (ASIC) for DTW with a low performance for energy-sensitive medical applications. Some researchers have also implemented efficient DTW acceleration on embedded platforms [45], [52]. However, the throughput and energy efficiency of DTW are still big concerns considering the ever-increasing volume of times series for data centers.

In this paper, we introduce a high-throughput and energy efficient memristor-based DTW (mDTW) architecture for real-time time series mining on data centers. In this new architecture, we adopt memristors for both computation and configuration, and the computation flow is in a continuous and asynchronous manner. To further improve the throughput, we exploit the predictability in DTW computing process. Specifically, we develop an early LB (ELB) algorithm and an effective early termination algorithm for DTW calculation. Experiments are performed with module evaluation and end-to-end evaluation including three popular applications: 1) similarity search; 2) classification; and 3) anomaly detection. Experimental results show that compared to existing approaches, the proposed mDTW can achieve a speedup and an energy efficiency improvement of 12×–43× and 51×–287×, respectively.

The remainder of this paper is organized as follows. Section II describes the backgrounds of DTW algorithm and memristors. The proposed mDTW architecture is presented in Section III. The experiment is discussed in Section IV. Section V reviews the related works, and this paper concludes in Section VI.

II. Background and Preliminaries

A. Dynamic Time Warping

DTW is a robust distance measure for time series. Suppose there are two sequences (or time series) as shown in Fig. 1(a), a sequence P of length n as a candidate, and a sequence Q of length m as a training template, where

$$P = P_1, P_2, \ldots, P_i, \ldots, P_n, \quad Q = Q_1, Q_2, \ldots, Q_i, \ldots, Q_m. \tag{1}$$

Sequences must be normalized to make a meaningful comparisons [37]. Z-normalization is adapted in this paper to remove offsets and amplitudes as

$$\mu_T = \frac{1}{m}\sum_{k=1}^{m} P_k,\ \sigma_T^2 = \frac{1}{m}\sum_{k=1}^{m} P_k^2 - \mu_T^2,\ P'_k = \frac{P_k - \mu_T}{\sigma_T}. \tag{2}$$

To measure the similarity of these two sequences, DTW creates an n-by-m matrix MT. The value of the (ith, jth) element in MT represents the distance, $d(P'_i, Q'_j)$, between points P_i and Q_j as

$$\mathrm{MT}(i,j) = d\left(P'_i,\ Q'_j\right) \tag{3}$$

which is called *distance matrix calculation* as shown in Fig. 1(b). There are many effective distance metrics such as Manhattan distance and Euclidean distance for *distance matrix calculation*. We choose the widely used Manhattan distance as shown in (4), which is also adopted in recent FPGA implementations [42], [48].

$$d\left(P'_i, Q'_j\right) = \left|P'_i - Q'_j\right|. \tag{4}$$

With the distance matrix, the warping path can be derived. There are three well-known constraints for the warping path in DTW: 1) boundary conditions; 2) continuity condition; and 3) monotonic condition. *Boundary conditions* means that the first/last point of P must correspond to the first/last point of Q. *Continuity condition* means that each element of the warping path in the matrix MT must have two elements of the warping path around it except the first and the last points. *Monotonic condition* requires that the extending direction of the warping path is right or top or top-right. The shortest warping path through the matrix is derived [38]

$$\begin{gathered} W(i,j) = w_{i,j}\left|P_i - Q_j\right| + \min\left\{W_{i,j-1}, W_{i-1,j}, W_{i-1,j-1}\right\} \\ W_{0,0} = 0 \quad W_{0,j} = W_{i,0} = \infty 1 \le i \le n;\ 1 \le j \le m \\ \mathrm{DTW}(P,Q) = W_{n,m} \end{gathered} \tag{5}$$

where W is the cumulate distance in the warping path, m and n are the length of Q and P, respectively. $w_{i,j}$ is for weighted DTW [22], which is set to 1 for general DTW computation. This procedure is called *warping path calculation*. Usually the calculation combination of *distance matrix calculation* and *warping path calculation* are regarded as DTW matrix calculation. The time complexity of DTW is $O(n^2)$.

The Sakoe–Chiba [40] band is used as DTW constraint as shown in Fig. 1(b). The DTW constraint, R, can reduce the available DTW path thus achieve speedup, which is defined as the rate of the warping length over the whole sequence and varies from 0% to 100%. It can also avoid some unpractical matchings, e.g., the first point of one sequence matches the last point in another sequence. DTW constraint is effective for many applications, and the choice of R depends on specific applications and configurations.

LB method is one of the most powerful optimization methods for DTW computation, which can prune a lot of sequences. The widely used LB method, LB_Keogh [37], is adopted in this paper, and the definition is as follows:

$$P_{i,\mathrm{upper}} = \max\{P_{i-R}, P_{i-R+1}, \ldots, P_{i+R-1}, P_{i+R}\} \tag{6}$$

$$P_{i,\mathrm{lower}} = \min\{P_{i-R}, P_{i-R+1}, \ldots, P_{i+R-1}, P_{i+R}\} \tag{7}$$

$$D_i = \begin{cases} Q_i - P_{i,\mathrm{upper}}, & \text{if } P_{i,\mathrm{upper}} < Q_i \\ P_{i,\mathrm{lower}} - Q_i, & \text{if } P_{i,\mathrm{lower}} > Q_i \\ 0, & \text{else} \end{cases} \tag{8}$$

$$\mathrm{LB}(P,Q) = \sum D_i. \tag{9}$$

The magical power of LB method is that if $\mathrm{LB}(P,Q)$ is larger than the lowest DTW value so far (the most similar one) for similarity search application [37], the following DTW calculation can be aborted as the final DTW value must be larger than the lowest DTW value. LB is also very effective for other applications, e.g., classification and anomaly detection.

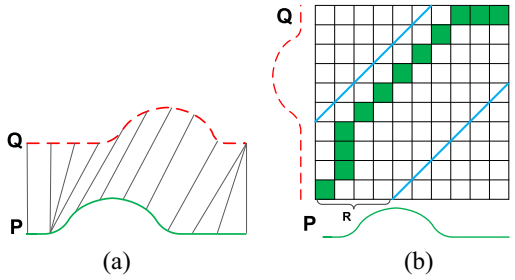


Fig. 1. (a) DTW matching indicated with lines. (b) DTW warping path based on a DTW distance matrix.

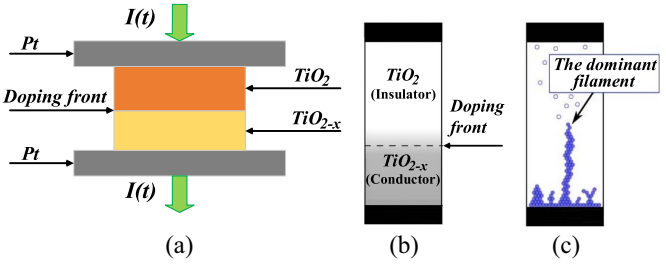


Fig. 2. Memristor structure and filament formulation. (a) Memristor structure. (b) Simple barrier memristor with nonlinear analog dynamics. (c) Filament formulation with nondeterministic digital dynamics (Reprinted from [7]).

For simplicity of presentation, other optimization methods (e.g., early abandon and reordering) are not presented here. Readers can refer to [37] for details.

B. Memristor

Memristor is a two-terminal resistive switching device which has nonlinear analog dynamics. The typical four-layer structure of memristors is shown in Fig. 2(a) [44]. Two of them are electrodes and the other two are TiO_2 and TiO_{2-x} layers, respectively. The resistance of the TiO_2 layer is high, while the resistance of the TiO_{2-x} layer is relatively low. Once current flows through the two layers, the doping front of the two layers gradually shifts, which makes the resistance of the device varies between that of high resistance state (HRS) and low resistance state (LRS) as shown in Fig. 2(b).

Recently studies show that memristors also have nondeterministic digital dynamics [49]. In the subthreshold voltage, it is probabilistic to form a single, dominant nanoscale filament as shown in Fig. 2(c). Memristors will also experience an abrupt resistance change. The nondeterminism will have a negative influence on analog computation, which will be considered in the context of DTW computation in this paper.

III. Memristor-Based DTW Architecture

In this section, we present the mDTW architecture. We adopt memristors for high efficient computation in LB modules and reconfigurability in DTW calculation modules. Based on the features of analog circuits, we propose a novel ELB algorithm and an effective early termination algorithm for DTW calculation. We also describe reconfigurability of DTW with memristors. For the sake of convenience, we use the same notation to present both a memristor and its resistance.

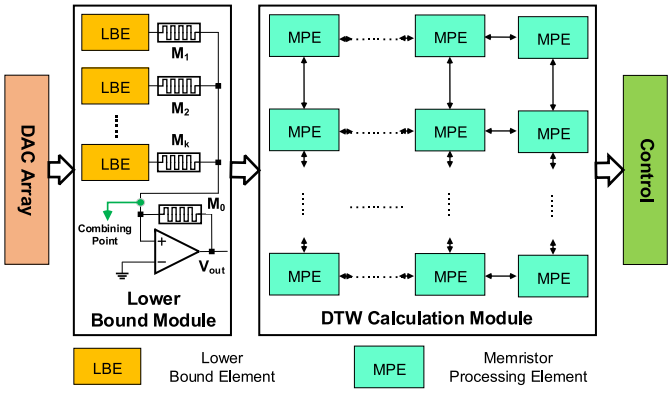


Fig. 3. Framework of the proposed DTW acceleration architecture.

Our design is to calculate the DTW distance in the analog domain. Thus, the computing procedure is in a continuous and asynchronous fashion. The DTW acceleration architecture comprises four modules: 1) a digital analog convertor (DAC) array; 2) an LB module; 3) a DTW calculation module; and 4) a control module as shown in Fig. 3.

In the architecture, the digital sequences are converted to analog signals through a DAC array first. To increase the computation efficiency, an LB module follows the DAC array to prune the input sequences. The LB module is made up of several LB elements (LBE) and one analog addition circuit. We propose an ELB algorithm, which enables a higher throughput.

The DTW calculation module is comprised of memristor PEs (MPEs), which performs calculation according to (5). The DTW calculation module maps DTW calculation of two sequences with length m and n to a $m \times n$ matrix of MPEs. The connections between MPEs are defined according to (5). The control module controls the data path among the above modules.

A. Architecture Overview

Note that, for DTW calculation modules, the convergence output is the DTW value. However, for LB modules, it mainly works in the uncovergence state.

In analog circuits, memristors are used for computation due to two reasons. First, using memristors as normal resistors enables the fine-tuning of memristance, which helps mitigate the impact of process variation and parasitic resistance. Second, by setting memristors to specific resistance, computation can be realized. For example, a widely used calculation of multiply accumulate operation with memristors is shown in the row structure in Fig. 3. V_{out} is the weighted sum of the output of each PE, and the weight is determined by the ratio of M_i $(1 \leq i \leq k)$ and M_0. For general computation of DTW, the ratio of 1 is adopted, and only the HRS and LRS of memristors are used. Recently, weighted DTW [22] has been widely adopted for a variety of applications. In this situation, different ratios between memristors are used, and memristors need to be set to specific resistance other than HRS or LRS. The calculation with memristors in the matrix structure follows the same principle. It should be pointed out that there is no LB optimization for weighted DTW. Thus, when weighted

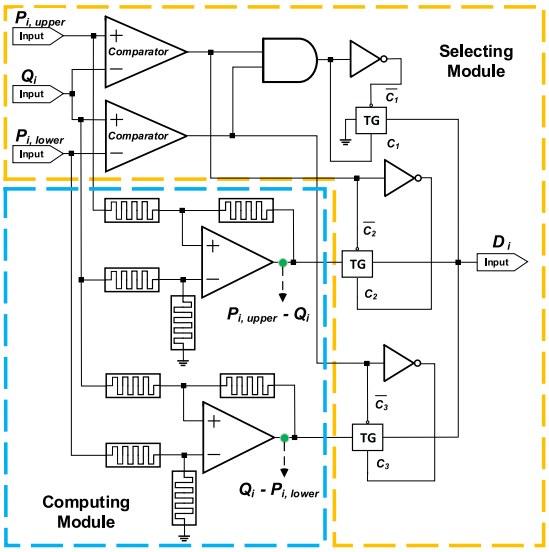


Fig. 4. Circuit structure (the resistance of memristors change according configurations) of LBE.

DTW is adopted, LB computation is not involved. Within analog circuits, the computation is conducted in a parallel manner. We discover that with identical circuit structures for inputs in the LB module, the relations of outputs in converage state and unconverage state are the same, which could be used for further optimization. The details of implementations are discussed in following sections.

Note that the nonlinear behavior of the memristor model is only used for resistance tuning. It is strictly avoided for accurate computation during normal operation [31], which is achieved with a low load voltage as discussed in Section IV-A. Thus, the polarity of memristors will not affect the performance, which is not indicated in all the figures in this paper.

B. Memristor-Based Lower Bound Module

1) Hardware Implementation: LBE realizes the function of (8) as shown in Fig. 4. By combining multiple LBEs, memristors, and one amplifier, the function in (9) is achieved.

LBE includes two modules: 1) a selecting module and 2) a computing module. The computing module has two analog subtractors, which are responsible for $Q_i - P_{i,\text{upper}}$ and $P_{i,\text{lower}} - Q_i$, respectively. The selecting module with comparators and transmission gates determines which result should be connected to the output port. With the obtained D_i from LBE, V_{out} in LB modules as shown in Fig. 3 can be formulated as follows:

$$V_{\text{out}} = M_0 \times \sum \frac{D_i}{M_i} = \sum \frac{M_0}{M_i} \times D_i. \tag{10}$$

Combining (9) and (10), it can be noted that by tuning memristors to $M_i = M_0$, V_{out} is exactly the expected output, LB(P, Q). Note that the tuning processing is presented only once for specific applications.

2) Algorithm Optimization: We propose an ELB algorithm to accelerate LB computation. In the LB module, each input has an equal position to each other, and the circuit structure

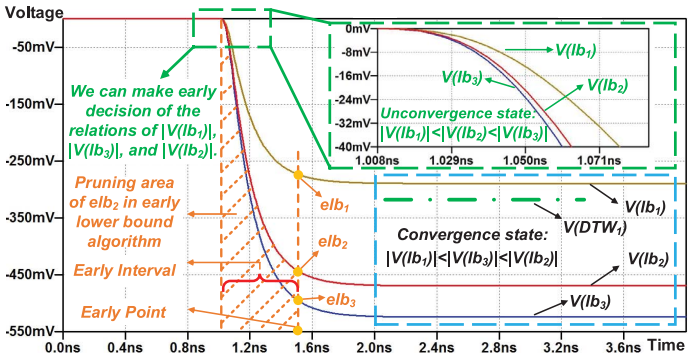


Fig. 5. Illustration of ELB algorithm. The relationship of the three LB outputs are the same in unconvergence and convergence states.

Algorithm 1 ELB Algorithm

Require: *Test*, *Template*[n];
Ensure: The minimum DTW distance;
1: $dtw_{bsf} \leftarrow$ INF;
2: $elb_{nu} \leftarrow$ INF;$elb_{bsf} \leftarrow$ INF;
3: $lb_{nu} \leftarrow$ INF;$lb_{bsf} \leftarrow$ INF;
4: **for** $i = 1 \rightarrow n$ **do**
5: $elb_i \leftarrow$FuncEarlyLowerBound(*Test*, *Template*[i]);
6: **if** $elb_i \leftarrow elb_{nu}$ **then**
7: $lb_i \leftarrow$FuncLowerBound(*Test*, *Template*[i]);
8: **if** $lb_i \leftarrow dtw_{bsf}$ **then**
9: $dtw_i \leftarrow$FuncDTW(*Test*, *Template*[i]);
10: **if** $dtw_i \leftarrow dtw_{bsf}$ **then**
11: **if** $lb_{bsf} \leftarrow dtw_i$ **then**
12: $elb_{nu} \leftarrow elb_{bsf}$; $lb_{nu} \leftarrow lb_{bsf}$;
13: **end if**
14: $dtw_{bsf} \leftarrow dtw_i$; $lb_{bsf} \leftarrow lb_i$; $elb_{bsf} \leftarrow elb_i$;
15: **end if**
16: **else**
17: $elb_{nu} \leftarrow elb_{bsf}$; $lb_{nu} \leftarrow lb_{bsf}$;
18: **end if**
19: **end if**
20: **end for**
21: Return dtw_{bsf};

for each input is identical. With this character, early decision in LB modules can be achieved, which means LB modules can process sequences with a shorter time rather than the convergence time. The detail is illustrated in Fig. 5. It can be noted that the relation of $|V(\text{lb}_1)|$, $|V(\text{lb}_2)|$, and $|V(\text{lb}_3)|$ in the unconvergence state and the convergence state are the same.

Based on this phenomenon, we propose a novel ELB algorithm shown in Algorithm 1. The general LB algorithm has to wait for the convergence state of the circuit to obtain LB, lb, for pruning. However, ELB algorithm prunes sequences with the ELB, elb, when the circuit is in unconvergence state. Thus, high throughput processing can be achieved. The sampling point is defined as Early Point, and the interval between the rising edge of the input and Early Point is defined as Early Interval. ELB algorithm maintains the nearest upper lb_{nu} to the best DTW, dtw_{bsf}, and its elb_{nu} is also stored.

We provide an example to explain how ELB algorithm works. As shown in Fig. 5, Early Point is set to 1.5 ns, and Early Interval is 0.5 ns. In the following discussion, all variables are the absolute values of their corresponding voltages

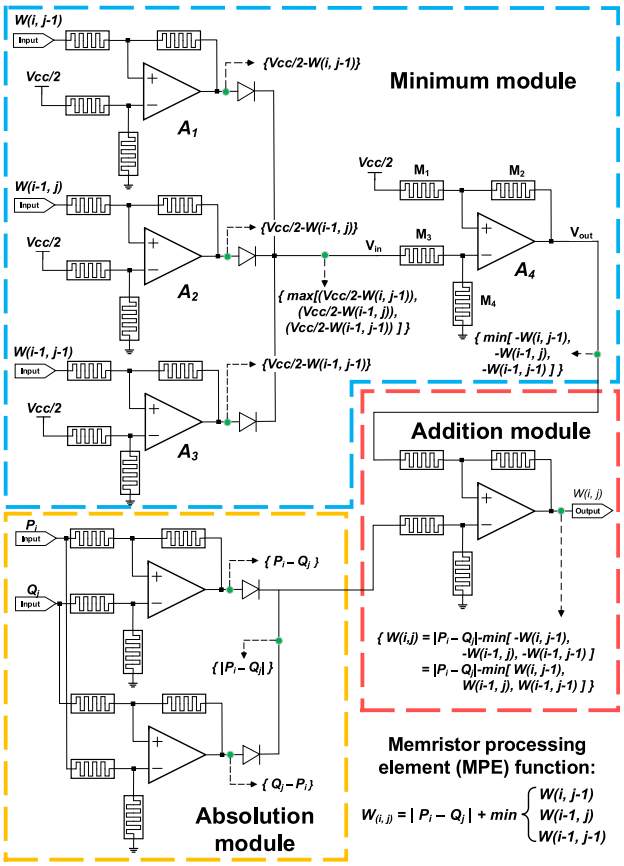


Fig. 6. Circuit structure of MPE (the resistance of memristors change according configurations).

for simplification. The first sequence, S_1, is calculated, and its LB lb_1 at 1.5 ns, elb_1, is also obtained. Thus, $\text{dtw}_{\text{bsf}} = \text{dtw}_1$, $\text{elb}_{\text{bsf}} = \text{elb}_1$, and $\text{lb}_{\text{bsf}} = \text{lb}_1$. By calculating the LB of sequence, S_2, it is pruned as its LB, lb_2, is larger than dtw_{bsf}. The absolute value of the LB lb_2 at 1.5 ns, elb_2, can be obtained. Then, $\text{elb}_{\text{nu}} = \text{elb}_2$ and $\text{lb}_{\text{nu}} = \text{lb}_2$. When calculating the LB of sequence, S_3, waiting for the convergence state is not needed. We can just obtain elb_3, and comparison is made between elb_3 and elb_{nu}. As elb_3 is larger than lb_{nu}, it can be predicted that lb_3 is larger than lb_2 in the convergence state. Thus, the LB and DTW calculation of S_3 can be aborted. Otherwise, elb_3 will be bypassed and DTW computation is needed. The nearest upper lb_{nu}, can be updated and it can be rather tight, which can prune a large number of sequences.

C. DTW Calculation Module

1) Hardware Implementation: The DTW calculation module is shown in Fig. 6, which includes three modules: 1) absolution module; 2) minimum module; and 3) addition module. The absolution module calculates the absolute value of $(P_i - Q_j)$. Two analog subtractors are used for calculating $(P_i - Q_j)$ and $(Q_j - P_i)$, respectively. Two diodes are to output the larger value of the two values. Thus, the output value is the positive value, which is the absolute value of $(P_i - Q_j)$. For conditions of $P_i = Q_j$, the output is also correct.

The minimum module obtains the minimum value of $W(i, j-1)$, $W(i-1, j)$, and $W(i-1, j-1)$. As diodes are perfect for maximum value calculation, we transform the minimum calculation to a maximum problem as shown in (11), where V_{cc} is the supply voltage. In step 1 of (11), the minimum problem is converted to a maximum problem, which can be easily calculated with diodes. However, there is a serious problem in the designs according to step 1. With diodes, the input current for the analog subtracter is fixed to positive, which means there is no negative current. As a result, the diode works in the cutoff region when the input is less than $Vcc/4$, and there is no current for the input. Thus, the maximum value of the output is $Vcc/4$, which is sufficient for DTW calculation. Step 2 is introduced to tackle the problem. The input and $V_{cc}/2$ switches their roles and are connected as shown in Fig. 6. Then the output is the minimum value with a negative sign, which can be easily solved by converting addition to subtraction. Weight factor $w_{i,j}$ supports weighted DTW, which can be achieved by configure memristors M_1 and M_2 to $M_1/M_2 = (2 - w_{i,j})/w_{i,j}$. Other memristors are all with the same resistance.

$$\begin{aligned} W(i,j) &= w_{i,j}\left|P_i - Q_j\right| + \min\left(W_{i,j-1}, W_{i-1,j}, W_{i-1,j-1}\right) \\ &= w_{i,j}\left|P_i - Q_j\right| + \left\{V_{cc}/2 - \max\left(V_{cc} - W_{i,j-1}\right.\right. \\ &\quad \times \left.\left. V_{cc}/2 - W_{i,j-1}, V_{cc}/2 - W_{i,j-1}\right)\right\} \text{ Step 1} \\ &= w_{i,j}\left|P_i - Q_j\right| - \left\{\max\left(V_{cc}/2 - W_{i,j-1}, V_{cc}/2\right.\right. \\ &\quad - \left.\left. W_{i,j-1}, V_{cc}/2 - W_{i,j-1}\right) - V_{cc}/2\right\} \text{ Step 2.} \end{aligned} \tag{11}$$

We take a more detailed discussion of the minimum module. With the feature of diodes, the output voltage of each diode in the minimum module must be above zero. Therefore, if inputs $W(i, j)$, $W(i-1, j)$, and $W(i, j-1)$ are all larger than $V_{cc}/2$, the output voltages of A_1, A_2, and A_3 are all below zero, and the output voltages of the three diodes will be zero. This will result in a smaller output voltage of the MPE than it should be, which may cause a large relative error in the DTW computation. However, we can make a reasonable assumption here that MPEs with inputs $W(i, j)$, $W(i-1, j)$, and $W(i, j-1)$ all larger than $V_{cc}/2$ have no impact on the results from the perspective of tasks. For tasks such as similarity search and classification, only low distances between templates and the test are considered, which has influence on the task results. If this MPE is not in the shortest path, it has no impact in the results and the assumption is verified. Otherwise, the output voltage of the DTW computation must be larger than $V_{cc}/2$. With proper voltage quantization, $V_{cc}/2$ corresponds to a very large DTW value, and the actual DTW value is even larger. Thus, the template with this DTW value has no impact on the task results and the assumption is also valid.

2) Algorithm Optimization: In this section, we propose an effective early termination algorithm to accelerate DTW computation. Early abandon is a very effective method that can further prune the DTW computations [37]. In this paper, we propose an effective early termination algorithm, which can achieve early abandon in DTW computation in the analog domain. We observe that the voltage of DTW output is

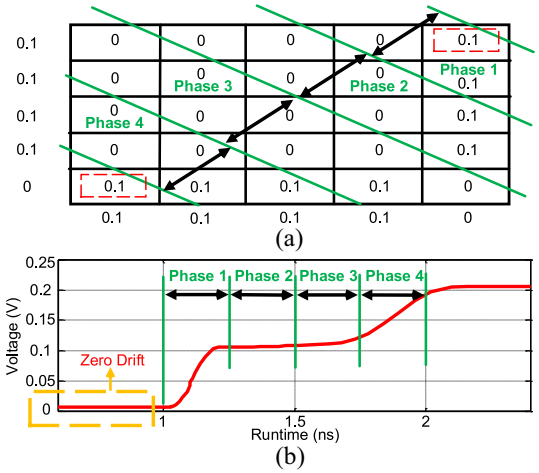


Fig. 7. Illustration of the effective early termination algorithm with DTW calculation modules. (a) DTW distance matrix is presented for sequence $P = [0.1v, 0.1v, 0.1v, 0.1v, 0v]$ and $Q = [0v, 0.1v, 0.1v, 0.1v, 0.1v]$. (b) DTW output increases with time.

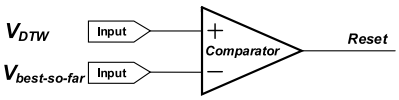


Fig. 8. Circuit for effective early termination algorithm.

monotonous with time. This is due to the fact that the minimum module in MPE guarantees the minimum value of its input as its output. Once the input of the minimum module increases, the output can only be updated to a larger value. To achieve such monotonicity, all the input should be zero in the initial state. A step-by-step procedure in the analog DTW calculation is shown in Fig. 7. The calculation is divided into four phases, and we can see that the distances in DTW distance matrix [1, 1] and [5, 5] contribute to the final DTW value of 0.2. As DTW distance matrix [5, 5] is near to the DTW output, the distance of 0.1 is added to DTW output in phase 1 in a short time interval. In phase 2 and phase 3, few distance matrixes can improve the DTW output, and the DTW output is almost constant. In phase 4, the distance of 0.1 in DTW distance matrix [1, 1] finally reaches the DTW output. The rising time is larger than that in phase 1, which is because that the propagation path is longer and the capacity of the path is larger. In order to enable effective early termination, a comparator is added as shown in Fig. 8. When the current voltage of the DTW output (the circuit is in unconvergence state) is larger than the $V_{\text{best-so-far}}$, the signal reset becomes active and the calculation can be aborted, which accelerates the whole computation.

D. Implementation Details

1) Reconfigurability With DTW: Memristors is introduced for the reconfigurability of Sakoe–Chiba band constraints. Specifically, the constraint reconfigurability is supported by configuring memristors connected with $A1$ in the minimum module as shown in Fig. 6. As a subtractor, $A1$ works according to (12). If no constraint is applied to the DTW cell, all

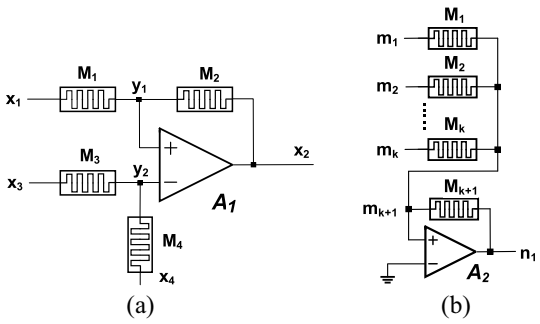


Fig. 9. Resistance tuning circuit: (a) analog subtractor and (b) analog adder.

memristors have the same resistance. If DTW constraint is on, the DTW cell needs to be shut down, and some memristors need to be tuned. The HRS and LRS of memristors can have the value difference by 2–3 orders of magnitude. By tuning M_2 and M_4 to HRS and M_1 and M_3 to LRS, (12) is translated as shown in (13), where δ is the resistance ratio of HRS and LRS of memristors

$$V_{\text{out}} = \left(\frac{M_4}{M_3} + 1\right) \times \left(\frac{M_1}{M_1 + M_2}\right) \times v_{\text{in}} - \frac{M_4}{M_3} \times V_{cc}/2 \quad (12)$$

$$V_{\text{out}} = v_{\text{in}} - \delta \times V_{cc}/2. \quad (13)$$

Thus, V_{out} is a large negative value, which is restricted to $-V_{cc}$ in practical circuits. With the addition module in DTW calculation modules, the output is V_{cc} which will not update the involved MPE.

2) Resistance Tuning: All the resistances in the DTW acceleration architecture are memristors. Thus, resistance tuning is required to make appropriate configurations for efficient computation [30]. This is also useful to minimize the influence of parasitic resistance. The process is presented as follows, which includes two parts, analog subtractor and analog adder as shown in Fig. 9. Note that we focus on the resistance tuning process, and the detailed programming circuit can be adopted from existing works [18], [24].

For analog subtractors as shown in Fig. 9(a), we set $y_1 = 0$ and $y_2 = 0$ in the first step. The four ports, x_1, x_2, x_3, and x_4 are used to modulate M_1, M_2, M_3, and M_4, respectively. In the second step, we verify the ratio of M_1/M_2 and M_3/M_4. When verifying $M_1/M_2 = k_1$, we set $y_2 = 0$ and $x_1 = 0.1$. By measuring x_2, the ratio k_1 can be verified with $x_2 = -k_1 \times 0.1$. For example, for analog subtractors in LBE, M_1 and M_2 should be set to LRS. Thus, if $x_2 = 0.1$ V, $M_1/M_2 = 1$ is configured successfully. When verifying $M_3/M_4 = k_2$, we set $y_1 = 0.1$ V and $x_3 = 0.1$. By measuring x_4, the ratio k_2 can be verified with $x_2 = -k_2 \times 0.1$. If verification is not successful, the first step will be applied to further modulate corresponding memristors. The two steps can be iterated several times for better precision.

For analog adders as shown in Fig. 9(b), we set $n_2 = 0$ in the first step. The $k + 1$ ports, $m_1, m_2, \ldots, m_k$ and m_{k+1} are adopted to modulate $M_1, M_2, \ldots, M_k$ and M_{k+1}, respectively. In the second step, M_{k+1} is regarded as the reference memristor, which is used to verify other memristors. We will set $m_1 = 0.1$ V and measure n_1 to verify M_1/M_{k+1}. If $n_1 = 0.1$ V, the configuration of M_1 is achieved. Otherwise, M_1 will

TABLE I
DTW ARCHITECTURE SETUP

Parameters	Configuration
Open loop gain of op-amp	1×10^4
Gain-bandwidth product of op-amp (GHz)	50
V_{cc}(V)	1
Voltage resolution	125mV for 1
Threshold voltage of diodes (V)	0

be modulated according to the offset to the configuration. The process of modulation and verification can be iterated for higher precision. The above tuning process for M_1 will be applied to other memristors.

For situations where the threshold voltage of memristors is larger than the supply voltage under some technology node, dual power can be adopted to avoid the possible damage to transistors by the large voltage for resistance tuning. With dual power, one supply voltage is for normal operation and another one is for memristor tuning. If memristor tuning is needed, the normal supply voltage is off and the tuning supply voltage is on. Each memristor need to be programmed individually, and the two ports of it need to be connected to the programming circuit.

3) Impact of Process Variation: Considering process variation, the actual resistance of memristors have a tolerance of $\pm20\%$ to $\pm30\%$, which will degrade the solution quality. Two steps are adopted to reduce the impact of process variation. First, we can discover that the solution quality is only the ratio of memristors. Thus, tolerance control technique [16] can be used to restrict the tolerance between two memristors lower than 1%. Second, post-fabrication resistance tuning can further reduce the negative effects of process variation.

IV. PERFORMANCE SIMULATION

In this section, we perform module and end-to-end evaluations of the proposed mDTW architecture with respect to accuracy, throughput, and energy efficiency. The three widely used applications, similarity search, classification, and anomaly detection are employed in the end-to-end evaluation. Specifically, the performance of the mDTW architecture obtained via simulations with SPICE [35] and MATLAB [21] is compared with existing works on GPUs and FPGAs.

A. Module Evaluation

1) Experimental Setup: We implement the proposed design in SPICE [35] with the 32 nm technology node, and the simulation setup is presented in Table I. For the sake of generality, the parameters of op-amps and diodes are set to typical values according to [8] and [31]. Particularly, a parasitic capacitance of 20fF is added to each circuit net to model the effect of parasitic capacitance [31]. The stochastic Biolek's model [7] for memristor is adopted and the parameters of the model are shown in Table II. All the parameters of the model are adopted from [7].

The parameter voltage resolution is to translate sequence values to voltages. We set the voltage resolution to a relatively large value of 125 mV. Longer sequence length need

TABLE II
STOCHASTIC BIOLEK'S MODEL PARAMETERS

V_0	τ	V_{T_0}	ΔV	R_{off}	R_{on}	$\Delta R_{on/off}$
0.156V	2.85×10^5s	3.0V	0.2V	100kΩ	1kΩ	5%

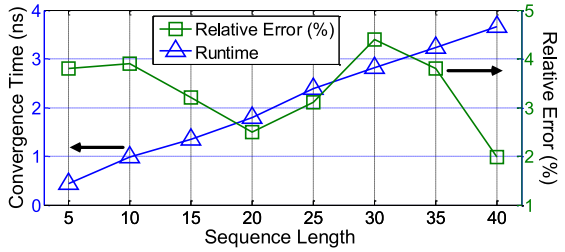


Fig. 10. Convergence time and relative error of LB module.

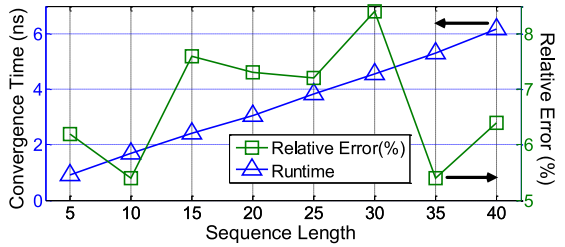


Fig. 11. Convergence time and relative error of DTW calculation module.

smaller voltage resolution. The translation process is as follows: after normalization, the sequence value 1 is translated to 125 mV. Other values follow the same principle, e.g., 1.2 and -0.5 are translated to 150 mV and -62.5 mV, respectively. In fact, it is still possible that the output voltage will overflow. As discussed in Section III-C, we make a reasonable assumption that overflowed outputs have no influence on data mining tasks. For DTW configuration, warping constraint of 5% is adopted.

2) Module Performance Characterization: First, we present performance evaluation for each module in the mDTW architecture. The convergence time which indicates how fast the analog circuit can operate is discussed. The convergence time is defined as the interval between the rising edge of the input and the timestamp when the DTW output is within 0.1% of the final value. We also analyze the relative error of the circuit solution compared to the optimal solution.

We adopt three data sets (Beef, Symbols, and OSU Leaf) from UCR Time Series Classification Archive [23]. For each data set, we formalize the sequences with different lengths. Note that two sequences for computation have the same length. From each dataset, we select 21 sequences and select the first sequence as the query, and LB and DTW computation are performed with the query and the rest 20 sequences.

The maximum convergence time and relative error of LB modules and DTW calculation modules are presented in Figs. 10 and 11, respectively. We can observe that the convergence time of both LB modules and DTW calculation modules are almost linear to the sequence length. This is due to the fact that both modules have a linear capacitance to the input size in the current propagation path. Specifically, for LB modules,

the capacitance of the combining point as shown in Fig. 3 increases linearly with the number of input or the sequence length. It can be noted that the relative error of DTW calculation modules is larger than that of LB modules, which is due to the fact that there exists zero drift in MPEs as shown in Fig. 7. In the experiment, we also find that though there exists relatively error, the magnitude relation of LB and DTW does not change, which means the ranks of both LB and DTW values are not affected by analog computation. Therefore, analog computation has no influence on the actual accuracy. However, analog computation has a relatively low data accuracy (usually 8 bits), which may degrade the accuracy. This data accuracy issue will be discussed in Section IV-B.

We can see that LB and DTW calculation modules operate with a high speed of about 0.1 ns and 0.18 ns per element. Compared with the state-of-the-art acceleration of DTW calculation modules (240 MHz, 4.17 ns per element) [42], we achieve a speedup of 23.3×. However, the state-of-the-art LB modules (150 MHz, 6.67 ns per sequence) [48] can fulfill one LB calculation per cycle with an adder tree, which outperforms our LB module when the sequence length is larger than 6.67 ns/0.01 ns = 667 (for LB modules, one-tenth convergence time is set as Early Interval). In Section IV-B, we will show that though the current LB module [48] have better performance with longer sequences, our mDTW can still have a better overall performance as the calculation time of LB module is far less than that of DTW module.

In the module performance experiment, all the results have low relative errors and are not influenced by the nondeterminism of the stochastic Biolek's model. This is due to the following two reasons. First, all memristors are under a voltage far less than the threshold voltage of memristors. It should be noted that only in the subthreshold voltage, it is probabilistic to form a single filament for stochastic resistance change. For LBEs, all inputs for analog subtractors are voltage values from sequence values. With a high voltage resolution, these input voltages are around several hundreds of millivolts which is far lower than the threshold voltage of 3.0 V. For analog adders in the LB module, the input voltage is small. The output voltage is usually very small, and there is a relatively very low possibility for M_0 to have a high voltage drop. For MPEs, the input voltages in the absolution module are very small, which is the same with the situations for LBEs. In the minimum module, the output voltage of diodes cannot be below zero, which makes the input voltages have a value less than or equal to $V_{cc}/2$. Thus, the voltage drop of all the memristors in the minimum module and the addition module is less than or equal to $V_{cc}/4 = 0.25$ V, which is also far lower than the threshold voltage of 3.0 V. Second, the computation time is far less than the transition time of about 1 μs for memristors. However, the running time for LB modules and DTW modules are about several nanoseconds. Considering the above two conditions, the nondeterminism of the stochastic Biolek's model has almost no influence on the LB and DTW computation in the analog domain.

However, if the computation time for LB modules and the experiment number both increase, the possibility of stochastic resistance change will also increase, which will decrease the computation accuracy. This can be tackled in two ways. First, triply redundant [15] can be applied for LB modules. Second, the circuit of the analog adder in the LB module needs to be improved to reduce the voltage drop of memristors. The above refinement will be our future work.

B. End-to-End Evaluation and Comparison

Second, we present end-to-end evaluations with three applications: 1) similarity search; 2) classification; and 3) anomaly detection with comparisons with state-of-the-art DTW accelerations. Accuracy, speedup, and energy efficiency are discussed in the experiments.

1) Experiment Setup: The experiment setup is as follows: 1) the LB module has 128 LBEs, and the DTW module has $128^2 \times R(2 - R)$ MPEs, and the DTW constraint of 5% is used; 2) for LB modules, one-tenth convergence time is set as Early Interval; and 3) effective early termination algorithm in DTW calculation module is not implemented as its pruning power is much weaker than that of LB. We perform the same process with the DTW accelerations with FPGAs [42], and the runtime is calculated with the ideal maximum throughput. Note that the FPGA implementation [48] for similarity search application is not considered here, which allows false dismissals [42]. The low-power DTW implementation with ASICs [32] is not considered here either as its performance is low. Parallel computation is not involved, and one LB module and one DTW calculation modules are taken into consideration for fair comparison. As the existing implementations do not support variable weighting factors, the weighting factor, σ_i, is set to 1. The experiment is simulated with SPICE and MATLAB, and MATLAB is responsible for simulating accuracy and the process of LB pruning.

Considering accuracy, data length of 8 bits is adopted for mDTW, which is the same with the FPGA implementation [42]. The reason for setting data length to 8 bits is that the highest data accuracy analog circuit can support is 8 bits [39]. However, for DTW computation if the data length of input data is 8 bits, the output will have a high possibility to be overflow. In order to tackle the overflow problem, we make a reasonable assumption here that the final result of DTW applications (usually with the lowest DTW value) will be in twice of the range of the input data. Thus, the input data is set to 7 bits and the output data is set to 8 bits. Note that as discussed in Section IV-A2, analog calculation produces no error about the relations of the DTW values between sequences. Therefore, we adopt MATLAB to simulate the accuracy of mDTW and FPGA with data length of 7 bits and 8 bits, respectively.

Energy efficiency is associated with runtime and power consumption as

$$E_{\text{efficiency}} = \frac{N}{E} \tag{14}$$

where N is the total sequence number, and E is the total consuming energy. A power estimation tool Power Estimator [1] is used to estimate the power consumption of FPGA. The power consumption of the FPGA implementation [42] with one LB module (8 stage adder tree with totally 255 adders) and one DTW module (128 process element for DTW size of

TABLE III
RESULTS OF SIMILARITY SEARCH

Platform	Dataset	Sequence Length	Query Length	LB Pruning Rate	Accuracy*	Execution Time	Speedup	Energy Efficiency (sequence/μJ)	Energy Improvement
FPGA	one-day-ECG	20140000	421	95.14%	165423th	27.49s	N/A	0.30	N/A
mDTW	one-day-ECG	20140000	421	90.84%	165427th	2.24s	12.29x	15.52	51.05x
FPGA	Two pattern	512000	128	90.74%	9301th	0.40s	N/A	0.53	N/A
mDTW	Two pattern	512000	128	87.63%	161010th	23.34ms	17.34x	37.81	72.03x
FPGA	UCR Data	1000000	128	63.36%	391984th	3.13s	N/A	0.13	N/A
mDTW	UCR Data	1000000	128	61.46%	391984th	0.14s	22.03x	12.14	91.48x

* Accuracy means the index of the first element of the most similar subsequence in the similarity search application.

128×128) is estimated to 3.742 W. We analytically model the power consumption of mDTW for energy efficiency discussion. The power for a recently popular op-amp with a gain-bandwidth product 303 GHz is 197 μW [53] under 0.35 μm technology node, and the power for the 32 nm technology node is projected to 18 μW with ideal scaling for capacitance. The same procedure goes for a recent 8-bit 1.6 Gsample/s DAC [46] in 90 nm technology node, and the projected power for the adopted DAC is 32 mW. A recent 35 mW 8.8 GSample/s ADC in 32 nm technology node [26] is adopted. The number of PEs in each column and row is set to 128. For sequence length larger than 128, tiling technique can be applied. For DTW configuration, the power consumption of the distance accelerator includes three parts: 1) op-amps; 2) ADCs/DACs; and 3) memristors around op-amps. The power consumption of the active op-amps is $(7R(2n-R)) \times 18\ \mu\text{W} = 0.20$ W, while the power consumptions of DACs and ADCs are $\lceil \text{Throughput}_{\text{in}}/1.6\ \text{GSample/s} \rceil \times 32\ \text{mW} = 0.13$ W and $\lceil \text{Throughput}_{\text{out}}/8.8\ \text{GSample/s} \rceil \times 35\ \text{mW} = 0.026$ W, respectively. Assuming at least one memristor is set to HRS from the source to the ground, the power consumption of memristors is $(7R(2n-R)) \times 2 \times 10\ \mu\text{W} = 0.22$ W. For LB module, the power consumption is 19.46 mW. Thus, the total energy consumption for mDTW is 0.58 W.

2) Similarity Search: The dataset one day *ECG* [4], *UCR* data [3], and two patterns [2] (joint all the sequences as one sequence) are selected for similarity search application. Similarity search is to find the candidate, which has the minimum DTW distance with the query from all the candidates.

Table III shows the results of the similarity search task. For the dataset *UCR* data [3], FPGA, and mDTW have the same accuracy as they find the same sequence as the most similar one to the query. For the dataset one day *ECG* [4], though the index of the first element has a minor difference of three, actually the corresponding subsequences are the same considering the query subsequence length of 421. However, for the dataset Two patterns [2], there exists errors with a 2.20% difference for DTW values, which is due to the data representations for FPGA with 8 bits and mDTW with 7 bits.

Compared with FPGA, mDTW has a speedup of 12.29×–22.03×, and the energy efficiency improvement is 51.05×–91.48×. We discover that the speedup for the dataset *UCR* data is the highest which is close to the speedup (23.3×) of DTW module in mDTW (0.18 ns per element) compared with that in FPGA (4.17 ns per element). This is due to the fact that the LB pruning rates are low (61.46%–63.36%) which means a lot of DTW calculations are required, and the processing time for LB is much shorter than that of DTW for each sequence. Thus, DTW processing time occupies the majority of the total processing time, and the speedup is determined by the performance ratio of DTW modules in mDTW and FPGA. For datasets one day *ECG* and Two patterns, as the LB pruning rate increases, the speedup decreases. This is because that the proportion of LB computation increases which is comparable with that of DTW computation, and the LB performance for FPGA and mDTW are comparable. We can also find that different data representations of 7 bits (mDTW) and 8 bits (FPGA) have a slight impact on the LB pruning rate. For energy efficiency, it has a positive correlation with the speedup, which follows the same principle.

3) Classification With k-Nearest Neighbors: Forty datasets from UCR Time Series Classification Archive with [2] with different sequence lengths are selected for classification application. kNN with $k = 1$ is used for classification.

Fig. 12 shows the accuracy varies with 40 datasets. It can be noticed that FPGA and mDTW have almost the same accuracy with different datasets. The average accuracy difference is 0.1%, which means mDTW can achieve a slightly better accuracy than FPGA.

The runtime and speedup are presented in Fig. 13. The speedup of mDTW is 16.30×–42.57×, and the average speedup is 23.4×. It can be learned that the speedup is around 22× for most datasets. This follows the same reason as discussed in Section IV-B2 for similarity search application: the LB pruning rate has a slight difference for mDTW (7 bits) and FPGA (8 bits), and the DTW processing time occupies the vast majority of the total processing time. However, for some datasets (e.g., fish and diatom size reduction) the LB pruning rate have a relatively large difference which results in different speedups. As shown in Fig. 14, the improvement of energy efficiency varies from 119.20× to 286.77×, and the average improvement is 160.43×.

4) Anomaly Detection: The long ECG data [5] is segmented to sequences of length 96. The peak points are from the *annotations.txt* from the database. With the peak point a_i, we get the segmented sequence as $a_{i-31}, \ldots, a_i, \ldots, a_{i+64}$. The first dimension of the 2-D data is used as representative.

A training processing [10] is needed to calculate threshold for anomaly detection. The first 20 normal sequences are stored as templates. The following 100 normal sequences are used to training to get the threshold. The rest

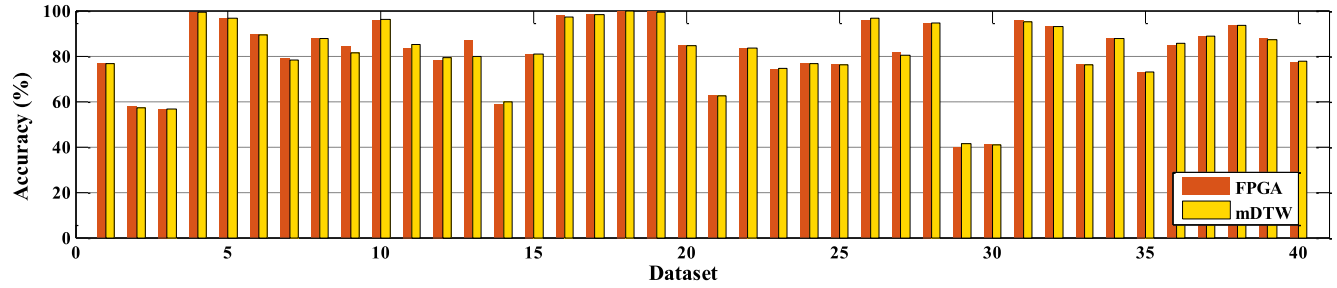


Fig. 12. Accuracy of classification application with *k*NN. The correspondence of dataset and the *x*-axis is as follows: (1, 50Words), (2, Adiac), (3, Beef), (4, CBF), (5, Coffee), (6, ECG200), (7, Face All), (8, Face Four), (9, fish), (10, Gun_Point), (11, Lighting2), (12, Lighting7), (13, Olive Oil), (14, OSU Leaf), (15, Swedish Leaf), (16, synthetic_control), (17, Trace), (18, Two_Patterns), (19, wafer), (20, yoga), (21, Chlorine_Concentration), (22, AdCinC ECG torsoiac), (23, Cricket *X*), (24, Cricket *Y*), (25, Cricket *Z*), (26, Diatom Size Reduction), (27, ECG Five Days), (28, Faces UCR), (29, Haptics), (30, Inline Skate), (31, Italy Power Demand), (32, MALLAT), (33, Medical Images), (34, Mote Strain), (35, Sony AIBO Robot Surface), (36, Sony AIBO Robot Surface II), (37, Star Light Curves), (38, Symbols), (39, Two Lead ECG), (40, uWave Gesture Library *X*), where *X* is the *x*-axis and *A* is the name of dataset in the format (*X*, *A*).

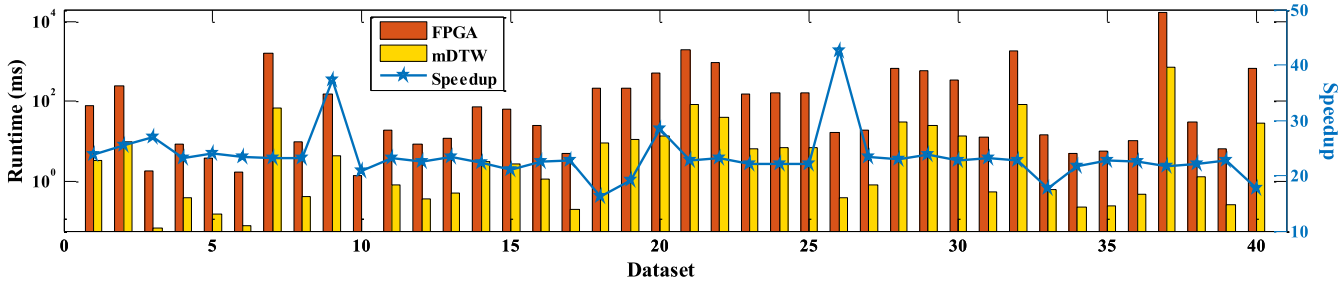


Fig. 13. Runtime and speedup of classification application with *k*NN. See the caption of Fig. 12 for the detail information of the *x*-axis dataset.

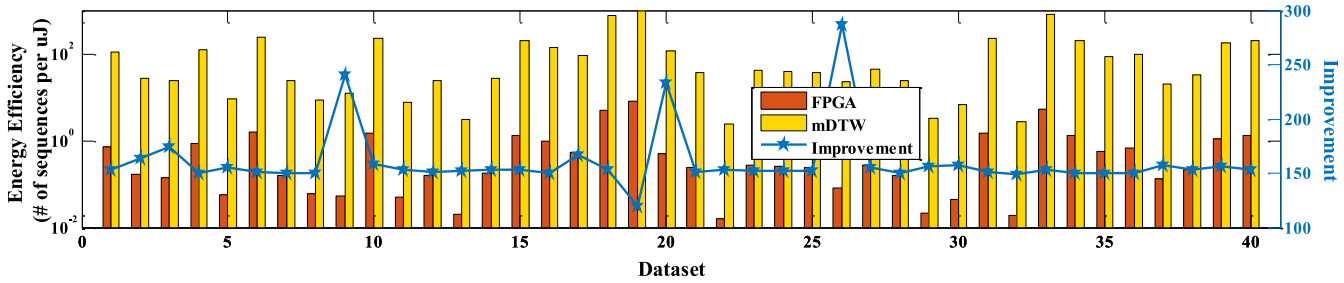


Fig. 14. Energy efficiency and improvement of classification application with *k*NN. See the caption of Fig. 12 for the detail information of the *x*-axis dataset.

of the ECG sequences are treated as the data to be detected.

For compact demonstration, accuracy and precision in the confusion matrix are selected to discuss the performance. As shown in Fig. 15, the accuracy and precision of FPGA and mDTW are almost the same for different datasets.

Fig. 16 shows the runtime comparison of FPGA and mDTW. The speedup varies from 20.95× to 23.64× with different dataset IDs, which follows the same principle discussed in Sections IV-B2 and IV-B3. As shown in Fig. 17, the energy efficiency improvements of mDTW over FPGA is 143.74×–152.49×.

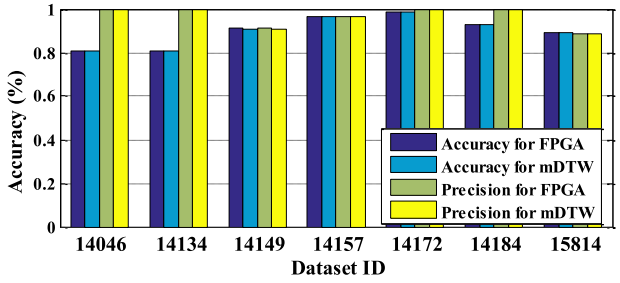


Fig. 15. Accuracy and precision of anomaly detection application.

V. Related Work

There is a large body of work exploring the novelty of memristor in new emerging fields beyond the straightforward memory application. One of the most important applications is neuromorphic systems [17], [28], [29], which adopts crossbar structure for efficient computation. Another important memristor application is analog and digital circuits. Logic/arithmetic operation [9] and programmable modules (e.g., filters [33], Chaos circuit [34], and memristor-based arithmetic-logic unit [47]) benefit from the feature of programmability. Memristor has also been applied to efficient architectures for area and energy optimization, e.g., FPGA+memristor [11] and FPAA+memristor [27].

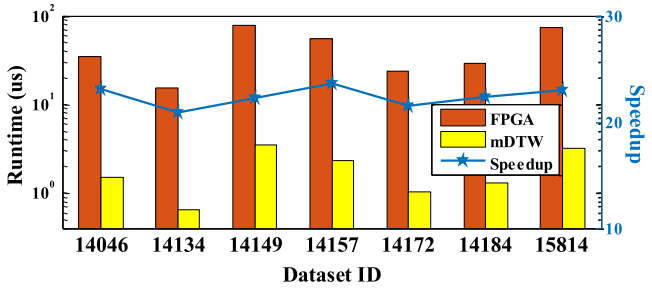


Fig. 16. Runtime of anomaly detection application.

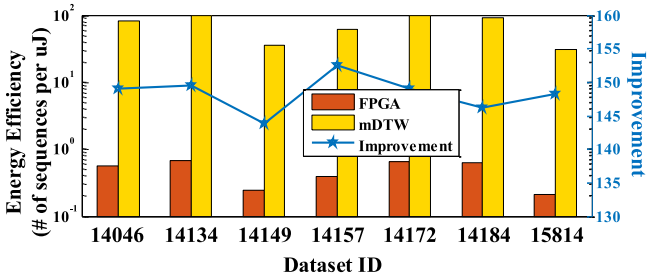


Fig. 17. Energy efficiency and improvement of anomaly detection application.

Recently, memristor has been adopted for algorithm optimization in analog domain. Liu and Zhang [31] implemented an analog substrate with memristors for max flow problem and achieves a speedup of 150× to 1500×. Memristor has also been applied to maze [36], shortest path algorithm [50], and bin packing algorithm [43]. The features in analog domain give a new sight for the improvement of performance. This paper is another exploring work of memristors in the matching algorithm application.

VI. Conclusion

In this paper, we propose a high-throughput and energy efficient mDTW architecture for time series mining on data centers. We performed memristor-based analog circuit designs for LB modules and DTW calculation modules. Based on the feature of analog circuits, we developed a novel ELB algorithm and an effective early termination algorithm for DTW computation acceleration. We also achieved reconfigurability in DTW calculation modules with memristors. Comprehensive experiments are presented with public available datasets. Module evaluation and end-to-end evaluation including similarity search, classification, and anomaly detection applications are presented. Experimental results show that the proposed mDTW can achieve a speedup and an energy efficiency improvement of 12×–43× and 51×–287×, respectively.

References

[1] *Xilinx Power Estimator (XPE)*. Accessed on Dec. 6, 2016. [Online]. Available: https://www.xilinx.com/products/technology/power/xpe.html

[2] *The UCR Time Series Classification Archive*. Accessed on Oct. 15, 2016. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data/

[3] *The UCR Suite Trillion Dataset*. Accessed on Oct. 15, 2016. [Online]. Available: http://www.cs.ucr.edu/~eamonn/trillion.zip

[4] *The UCR Suite*. Accessed on Oct. 15, 2016. [Online]. Available: http://www.cs.ucr.edu/~eamonn/ucrsuite.html

[5] *The MIT-BIH Long Term Database*. Accessed on Oct. 5, 2016. [Online]. Available: http://www.physionet.org/physiobank/database/ltdb/

[6] A. Adshead. (2014). *Data Set to Grow 10-Fold by 2020 As Internet of Things Takes Off*. [Online]. Available: http://www.computerweekly.com

[7] M. Al-Shedivat, R. Naous, G. Cauwenberghs, and K. N. Salama, "Memristors empower spiking neurons with stochasticity," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 2, pp. 242–253, Jun. 2015.

[8] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE simulations of memristors, memcapacitors and meminductors," *Radioengineering*, vol. 22, no. 4, p. 945, 2013.

[9] J. Borghetti *et al.*, "A hybrid nanomemristor/transistor logic circuit capable of self-programming," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 6, pp. 1699–1703, 2009.

[10] I. Boulnemour, B. Boucheham, and S. Benloucif, "Improved dynamic time warping for abnormality detection in ECG time series," in *Proc. Int. Conf. Bioinformat. Biomed. Eng.*, Granada, Spain, 2016, pp. 242–253.

[11] J. Cong and B. Xiao, "mrFPGA: A novel FPGA architecture with memristor-based reconfiguration," in *Proc. NANOARCH*, San Diego, CA, USA, 2011, pp. 1–8.

[12] G. Cormode, "Fundamentals of analyzing and mining data streams," in *Proc. Tutorial Workshop Data Stream Anal.*, Caserta, Italy, 2007, pp. 1–5.

[13] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[14] A. W.-C. Fu, E. Keogh, L. Y. H. Lau, C. A. Ratanamahatana, and R. C.-W. Wong, "Scaling and time warping in time series querying," *VLDB J.*, vol. 17, no. 4, pp. 899–921, 2008.

[15] R. S. Glaser *et al.*, "Process control interface system having triply redundant remote field units," U.S. Patent 5 428 769, Jun. 27, 1995.

[16] R. A. Hastings, *The Art of Analog Layout*. Upper Saddle River, NJ, USA: Prentice-Hall, 2006.

[17] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of BSB recall function using memristor crossbar arrays," in *Proc. DAC*, San Francisco, CA, USA, 2012, pp. 498–503.

[18] M. Hu *et al.*, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. 53rd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1–6.

[19] C. Hundt, B. Schmidt, and E. Schomer, "CUDA-accelerated alignment of subsequences in streamed time series data," in *Proc. ICPP*, Minneapolis, MN, USA, 2014, pp. 10–19.

[20] *Gartner Says the Internet of Things Will Transform the Data Center*, G. Inc., Toronto, ON, Canada, 2014.

[21] (2017). *M. Inc*. [Online]. Available: https://www.mathworks.com/

[22] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, 2011.

[23] E. Keogh and T. Folias, *The UCR Time Series Data Mining Archive*, Univ. California, Riverside, CA, USA, 2002. [Online]. Available: http://www.cs.ucr.edu/eamonn/TSDMA/index.html

[24] S. Kim *et al.*, "Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity," *Nano Lett.*, vol. 15, no. 3, pp. 2203–2211, 2015.

[25] S.-W. Kim, S. Park, and W. W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases," in *Proc. ICDE*, Heidelberg, Germany, 2001, pp. 607–614.

[26] L. Kull *et al.*, "A 35mw8 b 8.8 GS/S SAR ADC with low-power capacitive reference buffers in 32nm digital SOI CMOS," in *Proc. VLSI*, Kyoto, Japan, 2013, pp. C260–C261.

[27] M. Laiho *et al.*, "Analog signal processing on a FPAA/memristor hybrid circuit," in *Proc. ISCAS*, Melbourne, VIC, Australia, 2014, pp. 2265–2268.

[28] B. Li *et al.*, "Memristor-based approximated computation," in *Proc. ISLPED*, Beijing, China, 2013, pp. 242–247.

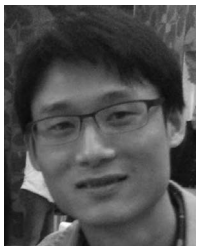
[29] B. Li, Y. Wang, Y. Wang, Y. Chen, and H. Yang, "Training itself: Mixed-signal training acceleration for memristor-based neural network," in *Proc. ASP-DAC*, Singapore, 2014, pp. 361–366.

[30] B. Liu *et al.*, "Digital-assisted noise-eliminating training for memristor crossbar-based analog neuromorphic computing engine," in *Proc. DAC*, Austin, TX, USA, 2013, pp. 1–6.

[31] G. Liu and Z. Zhang, "A reconfigurable analog substrate for highly efficient maximum flow computation," in *Proc. DAC*, San Francisco, CA, USA, 2015, pp. 1–6.
[32] R. Lotfian and R. Jafari, "An ultra-low power hardware accelerator architecture for wearable computers using dynamic time warping," in *Proc. DATE*, Grenoble, France, 2013, pp. 913–916.
[33] F. Merrikh-Bayat and S. Bagheri-Shouraki, "Mixed analog-digital crossbar-based hardware implementation of sign–sign LMS adaptive filter," *Analog Integr. Circuits Signal Process.*, vol. 66, no. 1, pp. 41–48, 2011.
[34] B. Muthuswamy, "Implementing memristor based chaotic circuits," *Int. J. Bifurcation Chaos*, vol. 20, no. 5, pp. 1335–1350, 2010.
[35] L. W. Nagel and D. O. Pederson, "SPICE: Simulation program with integrated circuit emphasis," Electron. Res. Lab., College Eng., Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/ERL M382, 1973.
[36] Y. V. Pershin and M. Di Ventra, "Solving mazes with memristors: A massively parallel approach," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 4, 2011, Art. no. 046703.
[37] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Beijing, China, 2012, pp. 262–270.
[38] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proc. CVPR*, vol. 2. Madison, WI, USA, 2003, pp. II-521–II-527.
[39] A. Rodríguez-Vázquez *et al.*, "ACE16k: The third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 5, pp. 851–863, May 2004.
[40] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.
[41] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *Proc. ICDE*, Istanbul, Turkey, 2007, pp. 1046–1055.
[42] D. Sart, A. Mueen, W. Najjar, E. Keogh, and V. Niennattrakul, "Accelerating dynamic time warping subsequence search with GPUs and FPGAs," in *Proc. ICDM*, Sydney, NSW, Australia, 2010, pp. 1001–1006.
[43] D. Stathis, I. Vourkas, and G. C. Sirakoulis, "Solving AI problems with memristors: A case study for optimal 'bin packing,"' in *Proc. PCI*, Athens, Greece, 2014, pp. 1–6.
[44] D. B. Strukov, G. S. Snider, D. R. Stewart, and S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
[45] J. Tarango, E. Keogh, and P. Brisk, "Instruction set extensions for dynamic time warping," in *Proc. 9th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Syn.*, Montreal, QC, Canada, 2013, pp. 1–10.
[46] W.-H. Tseng, J.-T. Wu, and Y.-C. Chu, "A CMOS 8-bit 1.6-GS/s DAC with digital random return-to-zero," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 1, pp. 1–5, Jan. 2011.
[47] I. Vourkas and G. C. Sirakoulis, "High-radix arithmetic-logic unit (ALU) based on memristors," in *Memristor-Based Nanoelectronic Computing Circuits and Architectures*. Cham, Switzerland: Springer, 2016, pp. 149–172.
[48] Z. Wang *et al.*, "Accelerating subsequence similarity search based on dynamic time warping distance with FPGA," in *Proc. ISFPGA*, Monterey, CA, USA, 2013, pp. 53–62.
[49] Y. Yang *et al.*, "Observation of conducting filament growth in nanoscale resistive memories," *Nat. Commun.*, vol. 3, Mar. 2012, Art. no. 732.
[50] Z. Ye, S. H. M. Wu, and T. Prodromakis, "Computing shortest paths in 2D and 3D memristive networks," in *Memristor Networks*. Cham, Switzerland: Springer, 2014, pp. 537–552.
[51] Y. Zhang, K. Adl, and J. Glass, "Fast spoken query detection using lower-bound dynamic time warping on graphical processing units," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5173–5176.
[52] H. Zhou *et al.*, "Energy-efficient pipelined DTW architecture on hybrid embedded platforms," in *Proc. 6th Int. Green Comput. Conf. Sustain. Comput. Conf. (IGSC)*, Las Vegas, NV, USA, 2015, pp. 1–8.
[53] L. Zuo and S. K. Islam, "Low-voltage bulk-driven operational amplifier with improved transconductance," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2084–2091, Aug. 2013.

Xiaowei Xu (S'14) received the B.S. and Ph.D. degrees in electronic science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2011 and 2016, respectively.

He is currently a Researcher with the School of Optimal and Electronic Information, Huazhong University of Science and Technology. He is currently a Visitor with the Department of Computer Science, University of Notre Dame, Notre Dame, IN, USA. His current research interests include biometrics, data mining, and embedded computing.



Feng Lin (S'11–M'15) received the B.S. degree from Zhejiang University, Hangzhou, China, in 2006, the M.S. degree from Shanghai University, Shanghai, China, in 2009, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN, USA, in 2015.

He was with Alcatel-Lucent, Boulogne-Billancourt, France, from 2009 to 2010. He is currently a Research Scientist with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA. His current research interests include signal processing, embedded sensing, human–computer interface, and their applications in wireless health and biometrics.

Aosen Wang (S'15) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2011. He is currently pursuing the Ph.D. degree in computer science and engineering with the State University of New York at Buffalo (University at Buffalo), Buffalo, NY, USA.

Then he joined Vimicro, Beijing, China, as an Algorithm and Software Engineer. His current research interests include low-power computer architecture and energy-efficient machine learning.

Xinwei Yao (M'14) received the Ph.D. degree from the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2013.

He is currently an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include terahertz-band communication networks, electromagnetic nanonetworks, wireless ad hoc and sensor networks, wireless power transfer, and the Internet of Things.

Dr. Yao has served on technical program committees of many IEEE/ACM conferences. He is a member of ACM.

Qing Lu received the B.E. degree in electronic and information engineering from the Dalian University of Technology, Dalian, China, in 2012 and the M.S. degree from Hong Kong Polytechnic University, Hong Kong, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of Notre Dame, Notre Dame, IN, USA.

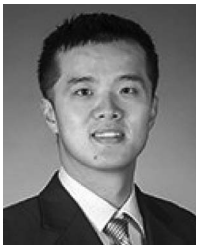
His current research interests include very large scale integration and system design, field-programmable gate arrays, and machine learning.

Mr. Lu was a recipient of the Best Paper Award in ATC 2015.

Wenyao Xu (M'13) received the Ph.D. degree from the Electrical Engineering Department, University of California at Los Angeles, Los Angeles, CA, USA, in 2013.

He is currently an Assistant Professor with the Computer Science and Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA. He holds five licensed U.S. and international patents, and has authored over 70 peer-reviewed journal and conference papers. His current research interests include embedded systems, computer architecture, wireless health, low-power technologies, and their applications in biomedicine, healthcare, and security.

Dr. Xu was a recipient of the Best Paper Award of the IEEE Conference on Implantable and Wearable Body Sensor Networks in 2013, and the Best Demonstration Award of ACM Wireless Health Conference in 2011.



Yiyu Shi (S'06–M'10–SM'15) received the B.S. (Hons.) degree in electronic engineering from Tsinghua University, Beijing, China, in 2005 and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 2007 and 2009, respectively.

He is currently an Associate Professor with the Department of Computer Science and Engineering and the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA. His current research interests include 3-D integrated circuits, hardware security, and renewable energy applications.

Prof. Shi was a recipient of several best paper nominations in top conferences, including the IBM Invention Achievement Award in 2009, the Japan Society for the Promotion of Science Faculty Invitation Fellowship, the Humboldt Research Fellowship for Experienced Researchers, the IEEE St. Louis Section Outstanding Educator Award, the Academy of Science (St. Louis) Innovation Award, the Missouri S&T Faculty Excellence Award, the National Science Foundation CAREER Award, the IEEE Region 5 Outstanding Individual Achievement Award, and the Air Force Summer Faculty Fellowship.

Yu Hu (M'10) received the B.Eng. and M.Eng. degrees from the Computer Science and Technology Department, Tsinghua University, Beijing, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA, USA, in 2009.

From 2010 to 2012, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. Since 2012, he has been a Professor with the School of Optimal and Electronic Information, Huazhong University of Science and Technology, Wuhan, China. His current research interests include intelligent transportation systems, connected vehicles, and embedded computing, general aspects of field-programmable gate arrays.

Dr. Hu was a recipient of the Outstanding Graduate Student Award from Tsinghua University in 2005, and a corecipient of the Best Contribution Award at the International Workshop of Logic and Synthesis in 2008. His research has been nominated for the Best Paper Award multiple times at the International Conference on Computer-Aided Design and Design Automation Conference.