MDA: A Reconfigurable Memristor-Based Distance Accelerator for Time Series Mining on Data Centers

Xiaowei Xu, *Student Member, IEEE*, Feng Lin, *Member, IEEE*, Wenyao Xu, *Member, IEEE*, Xinwei Yao, *Member, IEEE*, Yiyu Shi, *Senior Member, IEEE*, Dewen Zeng, 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 data mining is distance function, which is the fundamental element of many high data mining tasks. Recently various software optimization and hardware acceleration techniques have been proposed to tackle the challenge. However, each of these techniques is only designed or optimized for a specific distance function. To address this problem, in this paper we propose MDA, a high-throughput reconfigurable memristor-based distance accelerator for real-time and energy-efficient data mining with time series in data centers. Common circuit structure is extracted for efficiency, and the circuit can be configured to any specific distance functions. Particularly, we adopt the emerging device memristor for the design of MDA. Comprehensive experiments are presented with public available datasets to evaluate the performance of the proposed MDA. Experimental results show that compared with existing works, MDA has achieved a speedup of 3.5×–376× on performance and an improvement of 1–3 orders of magnitude on energy efficiency with little accuracy loss.**

***Index Terms*—Data center, data mining, distance function, memristors, time series.**

I. Introduction

ENERGY efficiency of data centers has been a primary focus in the past 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 by 2020 a major portion of the load will come from Internet-of-Things, which will yield over 4.4 zettabytes (5.5×10^{21} Bytes) of time series data by 2020 [3]. These time series data are transmitted to data centers for real-time mining [14]. It is therefore of utmost interest to explore techniques that handle time series data with high throughput and high energy efficiency.

The computational bottleneck of many data mining tasks such as classification and similarity search is the calculation of distance function [37], which is used to evaluate the similarity of two time series. Distance functions have a relatively high complexity, yet all data mining tasks will invoke it a huge number of times. Thus, the calculation of distance functions consumes a large fraction of the data mining time. For example, research results show that the computation of distance function takes up to more than 80% of the runtime for subsequence similarity search task [40].

Recently, software optimization and hardware acceleration have been widely exploited for distance functions. Dynamic time warping (DTW) has been optimized with lower bound methods [30], field programmable gate array (FPGA) [35], [40], [46]–[48], graphics processing unit (GPU) [35], and application-specific integrated circuit (ASIC) [23]. Manhattan distance (MD) has been accelerated with GPU [7]. Longest common subsequence (LCS), Hausdorff distance (HauD), and Hamming distance (HamD) have also been accelerated by GPU [19], [28], [39]. Edit distance (EdD) has been optimized on GPU [8] and ASIC [36]. However, each data center handles a variety of applications which use different distance functions. For example, a Cisco data center needs to deal with healthcare [13] and smart city applications [6]. The former adopts HamD for iris authentication [39] and LCS for electrocardiogram (ECG) similarity [9], while the latter uses DTW for vehicle classification [41]. None of these existing works on different platforms (GPU, FPGA, and ASIC) can work well in this scenario as they are optimized for a single distance function only. It remains an open problem in the literature how to design a reconfigurable accelerator that works for all popular distance functions with high throughput and high energy efficiency, which is of ultimate importance in data centers.

Meanwhile, the nonlinear analog dynamics of memristors has been extensively explored for nanoelectronic memories, computer logic, and neuromorphic/neuromemristive

Manuscript received November 25, 2017; revised February 22, 2018; accepted March 30, 2018. Date of publication May 8, 2018; date of current version April 19, 2019. This work was supported by the National Science Foundation of China under Grant 61272070. This paper was recommended by Associate Editor Y. P. Liu. *(Corresponding author: Dewen Zeng; 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 is with the Institute of Cyberspace Research, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: flin@zju.edu.cn).

W. Xu is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260-2500 USA (e-mail: 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).

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

D. Zeng and Y. Hu are with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: dewenzeng@hust.edu.cn; 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.2018.2834431

computer architectures [32]–[34]. Recently, memristors have also been used for query processing [11], tunable approximate computing [12], and distance acceleration [42]. Though these works also accelerated distance function calculation using memristors in submodules, they focused on specific applications such as query processing with only one distance function which cannot achieve high efficiency in the scenario of data centers.

In this paper, we address this problem by putting forward MDA, a novel reconfigurable memristor-based distance accelerator for high-throughput and high-energy-efficient time series data mining in data centers [43]. The contribution of this paper is threefold.

1) We present a specific analog circuit design as a unified hardware that can be reconfigured for a set of distance functions (including DTW, LCS, HauD, EdD, HamD, MD), and we extract the basic primitives to facilitate various distance functions to save chip area.
2) Memristors are adopted in analog circuit design for configurable resistance and accurate calculation.
3) We perform module and end-to-end evaluations, and experimental results show that compared with existing works, our work has achieved a speedup of 3.5×–376× on performance and an improvement of 1–3 orders of magnitude on energy efficiency with little accuracy loss.

The remainder of this paper is structured as follows. Section II describes the background and problem formulation. Section III presents the distance accelerator architecture and circuit designs. Module evaluation and end-to-end evaluation are presented in Sections IV and V, respectively. This paper is concluded in Section VI.

II. Background and Problem Formulation

In this section, the widely adopted six distance functions are introduced. DTW, LCS and EdD are dynamic programming methods, which can handle two sequences with different lengths, while HamD and MD only support sequences with the same length. HauD can also support two sequences with different lengths. In real applications, weight is introduced as the significance of each element is different. Interested readers can refer to [5], [17], [24], [27], [29], and [44] for the weighted version of DTW, LCS, MD, HamD, HauD, and EdD.

Distance functions are used to calculate the similarity between two sequences. Suppose there are two sequences P and Q as follows:

$$\begin{aligned} P &= \{P_1, P_2, \ldots, P_i, \ldots, P_m\} \\ Q &= \{Q_1, Q_2, \ldots, Q_j, \ldots, Q_n\} \end{aligned} \tag{1}$$

where m and n are the length of Q and P, respectively.

A. Dynamic Time Warping

The procedure of DTW calculation is a dynamic programming-based iterates process. Specifically, DTW is to calculate a shortest warping path between two sequences P and Q, which is derived as shown in (2), where D is the cumulate distance in the warping path. w_{ij} is the weight, which equals to 1 for general DTW and to other values ($\neq 1$) for weighted DTW. Smaller DTW(P, Q) value corresponds to higher similarity. Usually the Sakoe-Chiba band [30] is adopted for DTW, and its constraint R restricts the warping path. DTW has been optimized with lower bound methods [30], FPGA [35], [40], GPU [35] and ASIC [23].

$$\begin{aligned} &D_{i,j} = w_{i,j}\left|P_i - Q_j\right| + \min\left\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\right\} \\ &D_{0,0} = 0;\ D_{0,j} = D_{i,0} = \infty; \quad 1 \le i \le n;\ 1 \le j \le m \quad . \\ &DTW(P, Q) = D_{n,m} \end{aligned} \tag{2}$$

B. Longest Common Subsequence

LCS is to find the longest common subsequence of two strings. In order to apply LCS to time series, threshold is introduced to determine whether two elements are equal or not. LCS also belongs to dynamic programming as shown in (3), where V_{step} is the contribution of two equal elements. It should be noted that unlike DTW, smaller LCS(P, Q) value corresponds to lower similarity. LCS has been accelerated by GPU [28].

$$\begin{aligned} &L_{i,j} = \begin{cases} 0, \quad \text{if } i = 0 \text{ or } j = 0 \\ L_{i-1,j-1} + w_{i,j}V_{\text{step}} \\ \text{if } i, j > 0 \text{ and } \left|P_i - Q_j\right| \le \text{ threshold} \\ \max\left(L_{i,j-1},\ L_{i-1,j}\right) \\ \text{if } i, j > 0 \text{ and } \left|P_i - Q_j\right| > \text{threshold} \end{cases} . \\ &\text{LCS}(P, Q) = L_{n,m} \end{aligned} \tag{3}$$

C. Edit Distance

EdD is the number of operations in individual characters to transform one string into another. Thus, lower EdD value means higher similarity. The permitted operations include *replacement*, *insertion*, and *deletion*. By introducing threshold, EdD can also handle time series as shown in (4). EdD has been optimized on GPUs [8] and ASICs [36].

$$\begin{aligned} &E_{i,j} = \begin{cases} \min\left(E_{i-1,j} + w_{i-1,j}V_{\text{step}}, E_{i,j-1} + w_{i,j-1}V_{\text{step}}\right. \\ \left.E_{i-1,j-1} + w_{i-1,j-1}V_{\text{step}}\right) \\ \text{if } |P_i - Q_j| \le \text{threshold} \\ \min\left(E_{i-1,j} + w_{i-1,j}V_{\text{step}},\ E_{i,j-1} + w_{i,j-1}V_{\text{step}}\right. \\ \left.E_{i-1,j-1}\right) \\ \text{if } |P_i - Q_j| > \text{threshold} \end{cases} . \\ &E_{i,0} = i,\ E_{0,j} = j,\ EdD(P, Q) = E_{n,m} \end{aligned} \tag{4}$$

D. Hausdorff Distance

HauD measures how far two subsets are from each other. Low HauD value means two sets are close (high similarity) or each point in one set is close to each point in another set. The computation of HauD is shown in (5). HauD has been accelerated by GPU [19].

$$\text{HauD} = \max_{j \in n}\left(\min_{j \in n} w_{i,j}\left|P_i - Q_j\right|\right). \tag{5}$$

E. Hamming Distance

HamD is the number of positions at which the corresponding characters are different. Like LCS and EdD, threshold is adopted for time series. The calculation process is shown in (6). HamD has been accelerated by GPU [39].

$$\begin{aligned} &H_i = \begin{cases} H_{i-1} \text{ if } |P_i - Q_i| \le \text{ threshold} \\ H_{i-1} + w_i V_{\text{step}} \text{ if } |P_i - Q_j| > \text{threshold} \end{cases} . \\ &H_0 = 0,\ n = m, \quad \text{HamD}(P, Q) = H_n \end{aligned} \tag{6}$$

F. Manhattan Distance

MD is a simple but rather popular method for time series [7], which is the sum of absolute differences in the corresponding positions. The calculation process is given as shown in (7). MD has been accelerated with GPU [7].

$$\mathrm{MD}(P, Q) = \sum_{i}^{n} w_i |P_i - Q_i|, \quad n = m. \tag{7}$$

G. Problem Formulation

From the above discussion it is clear that any existing accelerator is for a specific distance function only, and cannot be shared by multiple functions. However, this is exactly what is needed in data centers. In this paper, we formulate the problem of reconfigurable distance accelerator as follows: given memristors and basic circuit devices, find a circuit structure design that can be reconfigured to support multiple distance functions with high performance, high energy efficiency and low area consumption.

III. Accelerator Architecture

A. Architecture Overview

The proposed MDA comprises four modules: 1) a digital-to-analog convertor (DAC) array; 2) a computation module; 3) a control and configuration module; and 4) an analog-to-digital convertor (ADC) array as shown in Fig. 1. The DAC and ADC arrays are used to convert time series data between digital signals and analog signals. The control and configuration module has two responsibilities: 1) control the dataflow between modules and 2) reconfigure circuit connections in the computation module to perform specific distance functions with the configuration lib.

The configurable computation module calculates the distance functions. In order to save chip areas, we extract the basic primitive, the processing element (PE) of the analog circuits of distance functions. Each PE is compromised of several basic elements which will be discussed in detail in the next section. The connections between the basic elements in PE is realized with transmit gates (TGs). All the adopted six distance functions are aggregated into two structures for the connection between PEs: 1) matrix structure (for DTW, LCS, HauD and EdD) and 2) row structure (for MD and HamD) as shown in Fig. 1. The circuit structures for different algorithms have a high similarity with each other in matrix and row structure, respectively. The reuse of op-amps and their corresponding memristors are labeled as shown in Fig. 3. It can be noticed that the configuration of connections for the two structures is relatively simple, and the circuit elements have a high resources utilization. By configuring each PE and connections between PEs, the function of specific distance can be achieved. The details of configurations are discussed in Section III-B. When the sequence length is larger than the number of PEs in each row or column, tiling technique will be applied and the throughput will decrease.

In analog circuits, memristor is used for computation due to two reasons. First, using memristors as normal resistors enables the fine-tuning of memristance, which contributes to mitigate the impact of process variation and parasitic resistance. Second, By setting memristors to specific resistance, computation can be realized. A typical calculation of memristors is shown in the row structure in Fig. 1. 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 MD, DTW, LCS, HamD, EdD, and HauD, the ratio of 1 is adopted, and only the high resistance state (HRS) and low resistance state (LRS) of memristors are used. Recently, weighted version of MD [29], DTW [17], LCS [5], HamD [44], EdD [27], and HauD [24] have 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. Within analog circuits, the computation is conducted in a parallel manner. We discover that with identical circuit structure, 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 Section III-C.

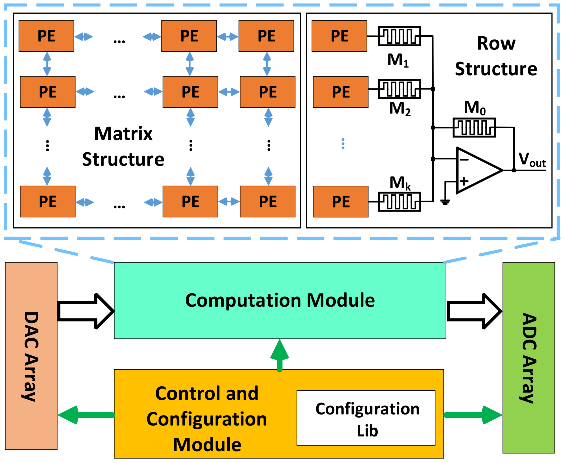


Fig. 1. Architecture of the distance accelerator, MDA.

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 [22], which is achieved with a low load voltage as discussed in Section IV-B. Thus, the polarity of memristors will not affect the performance, which is not indicated in all the figures in this paper.

B. Hardware Implementation

1) Circuit of Processing Element: PE can be configured to a variety of distance functions according to the configuration lib. As shown in Fig. 2, it is compromised of several basic elements: absolution module, minimum module, individual subtractor module, control module, and connection module. The absolution module and minimum module are used to calculate the absolution value of two numbers and the minimum value of three numbers, respectively. Particularly, the minimum calculation is a combination of subtraction and maximum

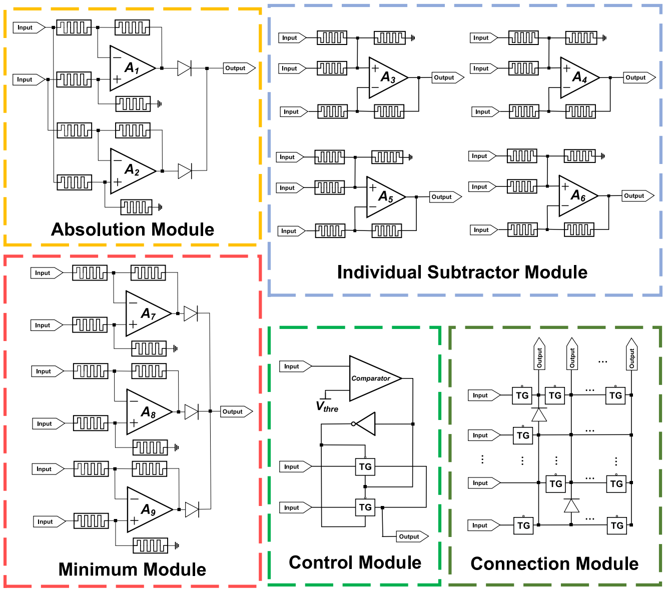


Fig. 2. Overall circuit structure of PE.

calculation. The details of the two operations are discussed and analyzed in detail with specific distance functions. The control module includes a comparator and a either/or circuit, which is used to select the appropriate output according to the input to the comparator. The individual subtractor module can be configured to subtraction or addition operation. The connections between the basic elements are realized with the connection module which is a TG-based sparse array. Specific connections between the inputs and outputs can be realized by configuring the TGs. Note that the TG-based array is sparse, which means that some inputs can only be connected to some specific outputs. In the TG-based array there exists some diodes to calculate the maximum value of several inputs as shown in Fig. 3(d1) and Fig. 3(d2). Note that as each PE is independent and the connection is flexible, MDA can be configured to several groups, each of which supports one distance function calculation.

Compared with accelerating only one distance function, our reconfigurable approach comes with a cost. We need to add more circuit devices in each PE to support multiple distance functions. Also, the connection configuration between PEs becomes complex resulting with more area consumption. This is the cost we have to pay to achieve flexibility. Note that data centers especially benefit from such flexibility.

2) Circuit of Dynamic Time Warping: The DTW calculation module is shown in Fig. 3(a), 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. Weight factor $w_{i,j}$ supports weighted DTW, which can be achieved by configuring 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

$$\begin{aligned} D_{i,j} &= w_{i,j}\left|P_i - Q_j\right| + \min\left(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\right) \\ &= w_{i,j}\left|P_i - Q_j\right| + \big\{V_{cc}/2 - \max\big(V_{cc} - D_{i,j-1} \\ &\qquad V_{cc}/2 - D_{i,j-1}, V_{cc}/2 - D_{i,j-1}\big)\big\} \quad \text{Step 1} \\ &= w_{i,j}\left|P_i - Q_j\right| - \big\{\max\big(V_{cc}/2 - D_{i,j-1}, V_{cc}/2 \\ &\qquad - D_{i,j-1}, V_{cc}/2 - D_{i,j-1}\big) - V_{cc}/2\big\} \quad \text{Step 2.} \end{aligned} \tag{8}$$

The minimum module obtains the minimum value of $D_{i,j-1}$, $D_{i-1,j}$, and $D_{i-1,j-1}$. As diodes are perfect for maximum value calculation, we transform the minimum calculation to a maximum problem as shown in (8), where V_{cc} is the supply voltage. In Step 1, the minimum problem is converted to a maximum problem, which can be easily calculated with diodes. However, there is a 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 $V_{cc}/4$, and there is no current for the input. Thus, the maximum value for the output is $V_{cc}/4$, which is insufficient for DTW calculation. Step 2 is introduced to tackle the problem. The input and $V_{cc}/2$ switch their roles as shown in Fig. 3(a). Then, the output is the minimum value with a negative sign, which can be easily solved by converting addition to substraction.

3) Circuit of Longest Common Subsequence: The PE circuit of LCS is shown in Fig. 3(b). The calculation of $L_{i,j}$ depends on the elements of sequences and PEs besides it.

The PE circuit contains two modules: a selecting module and a computing module. The selecting module fulfills the calculation of conditions in (3). To determine whether P_i is equal to Q_j, we first calculate the absolute value of (P_i-Q_j), and then compare the absolute value with a threshold voltage V_{thre}. If the absolute value is less than the threshold voltage, we assume that P_i is equal to Q_j, otherwise not. The TG determines which part should connect to the output.

The computing module is consisted of two parts. The first part calculates the sum of $L_{i-1,j-1}$ and $w_{i,j}V_{\text{step}}$. The second part outputs the maximum value of $L_{i,j-1}$ and $L_{i-1,j}$ with diodes. Weight factor $w_{i,j}$ supports weighted LCS by configuring memristors M_1, M_2, M_3, M_4, and M_5. Assuming $M_1/M_2 = k_1$, M_3 should be set to $w_{i,j}k_1M_2$, and the relation of M_4 and M_5 is $M_5/M_4 = (1 + k_1)w_{i,j}$.

4) Circuit of Edit Distance: Fig. 3(c) shows that the PE circuit of EdD includes two modules: 1) a computing module and 2) a minimum module. In the computing module, we have three computation paths. The first computation path is associated with $E_{i-1,j-1}$, which is the result of the left-lower PE. We calculate the absolute value of $(P_i - Q_j)$ and use a comparator to determine whether P_i is equal to Q_j. If P_i is equal to Q_j, the output of the comparator will be high and the output of the first path will be $E_{i-1,j-1} + w_{i-1,j-1}V_{\text{step}}$, otherwise will be $E_{i-1,j-1}$. The second and the third path share the same circuit structure, and the outputs are $E_{i-1,j} + w_{i-1,j}V_{\text{step}}$ and $E_{i,j-1} + w_{i,j-1}V_{\text{step}}$, respectively. V_{step} is a unit voltage, and the exact result can be obtained by dividing $E(m,n)$ by V_{step}. For weighted LCS the configuration of memristors around op-amp A_3, A_4, and A_5 in Fig. 3(c) are the same with that in Fig. 3(b).

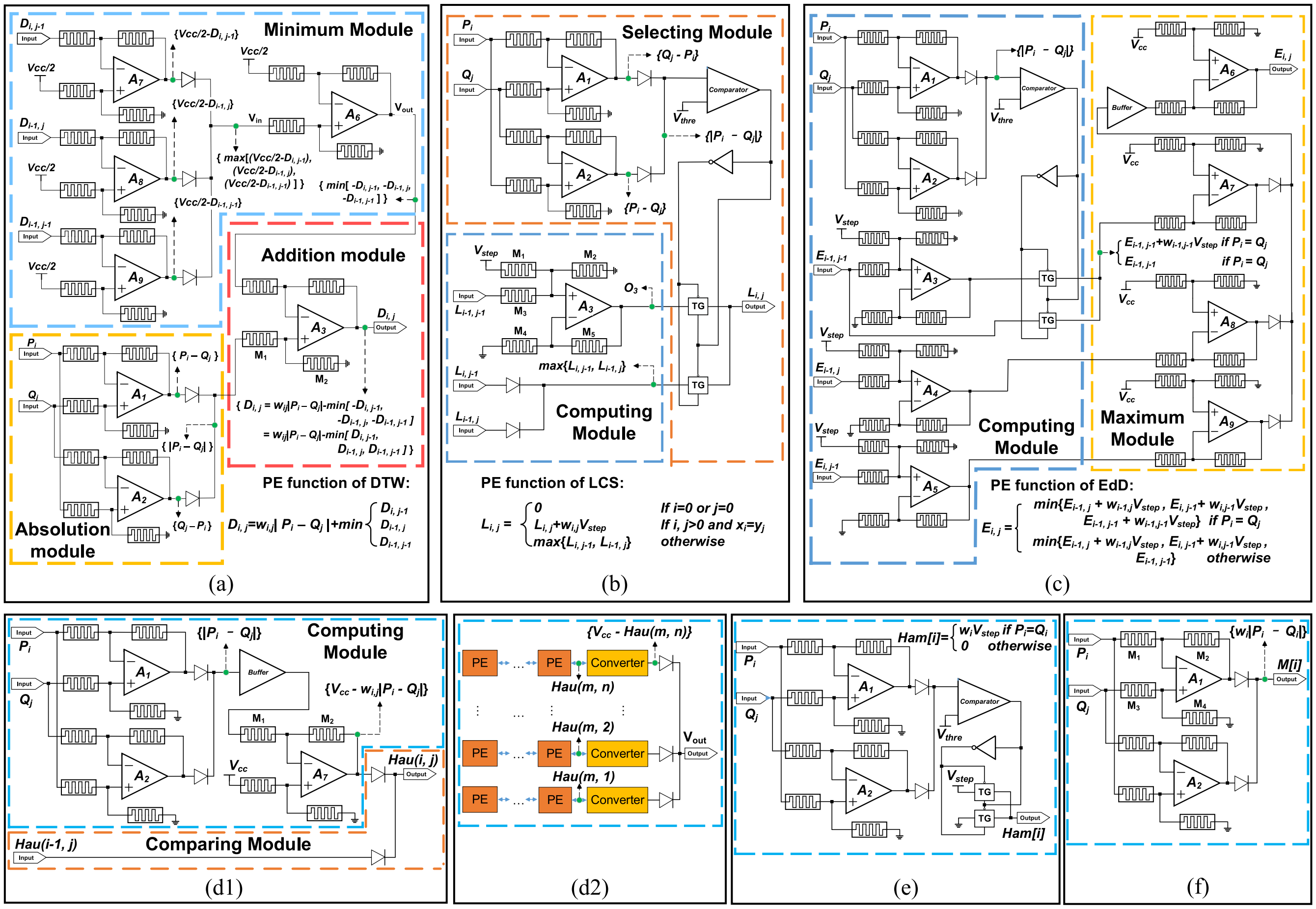

Minimum Module
$D_{i,j-1}$
Input
$V_{cc}/2$
A_7
$\{V_{cc}/2-D_{i,j-1}\}$
$D_{i-1,j}$
A_8
$\{V_{cc}/2-D_{i-1,j}\}$
$D_{i-1,j-1}$
A_9
$\{V_{cc}/2-D_{i-1,j-1}\}$
V_{in}
A_6
V_{out}
$\{ \max[(V_{cc}/2-D_{i,j-1}), (V_{cc}/2-D_{i-1,j}), (V_{cc}/2-D_{i-1,j-1})] \}$
$\{ \min[-D_{i,j-1}, -D_{i-1,j}, -D_{i-1,j-1}] \}$
Addition module
A_3
M_1
M_2
$D_{i,j}$
Output
$\{ D_{i,j} = w_{ij}|P_i - Q_j| - \min[-D_{i,j-1}, -D_{i-1,j}, -D_{i-1,j-1}] = w_{ij}|P_i - Q_j| - \min[D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}] \}$
P_i
Q_j
A_1
$\{ P_i - Q_j \}$
$\{ |P_i - Q_j| \}$
A_2
$\{Q_j - P_i\}$
Absolution module
PE function of DTW:
$D_{i,j}=w_{i,j}|P_i - Q_j| + \min \begin{cases} D_{i,j-1} \\ D_{i-1,j} \\ D_{i-1,j-1} \end{cases}$
(a)
Selecting Module
$\{Q_j - P_i\}$
Comparator
V_{thre}
$\{|P_i - Q_j|\}$
$\{P_i - Q_j\}$
V_{step}
M_3
M_4
M_5
O_3
$L_{i-1,j-1}$
$\max\{L_{i,j-1}, L_{i-1,j}\}$
$L_{i,j-1}$
$L_{i-1,j}$
TG
$L_{i,j}$
Computing Module
PE function of LCS:
$L_{i,j} = \begin{cases} 0 & \text{If } i=0 \text{ or } j=0 \\ L_{i,j}+w_{i,j}V_{step} & \text{If } i,j>0 \text{ and } x_i=y_j \\ \max\{L_{i,j-1}, L_{i-1,j}\} & \text{otherwise} \end{cases}$
(b)
$E_{i-1,j-1}$
$E_{i-1,j}$
$E_{i,j-1}$
A_4
A_5
V_{cc}
Buffer
$E_{i,j}$
$\begin{cases} E_{i-1,j-1}+w_{i-1,j-1}V_{step} & \text{if } P_i = Q_j \\ E_{i-1,j-1} & \text{if } P_i = Q_j \end{cases}$
Maximum Module
PE function of EdD:
$E_{i,j} = \begin{cases} \min\{E_{i-1,j} + w_{i-1,j}V_{step}, E_{i,j-1}+ w_{i,j-1}V_{step}, E_{i-1,j-1} + w_{i-1,j-1}V_{step}\} & \text{if } P_i = Q_j \\ \min\{E_{i-1,j} + w_{i-1,j}V_{step}, E_{i,j-1}+ w_{i,j-1}V_{step}, E_{i-1,j-1}\} & \text{otherwise} \end{cases}$
(c)
$\{V_{cc} - w_{i,j}|P_i - Q_j|\}$
$Hau(i, j)$
$Hau(i-1, j)$
Comparing Module
(d1)
$\{V_{cc} - Hau(m, n)\}$
PE
Converter
$Hau(m, n)$
$Hau(m, 2)$
$Hau(m, 1)$
(d2)
$Ham[i]=\begin{cases} w_iV_{step} & \text{if } P_i=Q_i \\ 0 & \text{otherwise} \end{cases}$
$Ham[i]$
(e)
$\{w_i|P_i - Q_i|\}$
$M[i]$
(f)

Fig. 3. PE circuit structures of DTW, LCS, EdD, HauD, HamD and MD. Particularly, HauD has a different PE connection.

The minimum module calculates the minimum value among the output of the three paths in the computing module. As the diodes can easily solve the maximum problem, we use a subtractor circuit to make it a maximum problem.

The same problem arises here, which also exists in the PE circuit structure of DTW. The current through the diode must be in the right direction, which means the output of the diodes in the maximum module must be higher than $V_{cc}/2$. In order to solve the problem, we add a buffer at the output of the diodes to ensure that the output can be lower than $V_{cc}/2$.

5) Circuit of Hausdorff Distance: Fig. 3(d1) shows the PE circuit structure of HauD, which is compromised of a computing module and a comparing module. The computing module is consisted of two steps, the first step is to calculate the absolute value of (P_i-Q_j). As explained in Section III-B2, diodes and V_{cc} are also used here to solve the minimum problem in the second step.

The comparing module outputs the maximum value of $D(i-1, j)$ and $V_{cc} - w_{i,j}|P_i - Q_j|$. We add a buffer between the output of diodes and the negative input of A_3 [shown in Fig. 3(d1)], therefore the output voltage of $w_{i,j}|P_i - Q_j|$ can be below $V_{cc}/2$. For weighted HauD, the configuration of memristors $M_2/M_1 = M_3/M_4 = w_{i,j}$ should be applied.

Fig. 3(d2) shows the PE circuit structure of HauD. Given Q_j, we check every elements of sequence P and calculate the value of $Hau(m, j)$, which is the maximum value of $V_{cc} - w_{i,j}|P_i - Q_j|$ $((1 \le i \le k))$. With the same processing for Q_j in sequence Q, we have $Hau(m, 1)$, $Hau(m, 2)$, $\ldots$, $Hau(m, n)$. Then, a converter is used to process each $Hau(m, j)$ in which the output is the difference of V_{cc} and $Hau(m, j)$. Therefore, the output of the converter is the minimal $w_{i,j}|P_i - Q_j|$ where j is fixed and i varies. Finally, we use diodes to output the maximum value of all minimal $w_{i,j}|P_i - Q_j|$, and the result is the HauD of P and Q.

6) Circuit of Hamming Distance: The PE circuit structure of HamD is shown in Fig. 3(e). The absolute value calculation module and a comparator are used to calculate whether P_i is equal to Q_j. If P_i is equal to Q_j, the output of the comparator will be high, and the output of $Ham[i]$ will be V_{step}. Otherwise, the output will connect to the ground, and $Ham[i]$ will remain zero. When all PEs finish computation, an analog adder is adopted to add all $Ham[i]$, and the output is the HamD of P and Q. Weighted HamD is achieved by configuring memristors to $M_0/M_k = w_k$ in the row structure in Fig. 1.

7) Circuit of Manhattan Distance: Fig. 3(f) shows the PE circuit structure of MD, which is the subset of that of HamD. Like HamD, when all the PE fulfill computation, we use an analog adder to add all $D[i]$, and the output is the MD of P and Q. For weighted MD, the configuration is the same with weighted HamD.

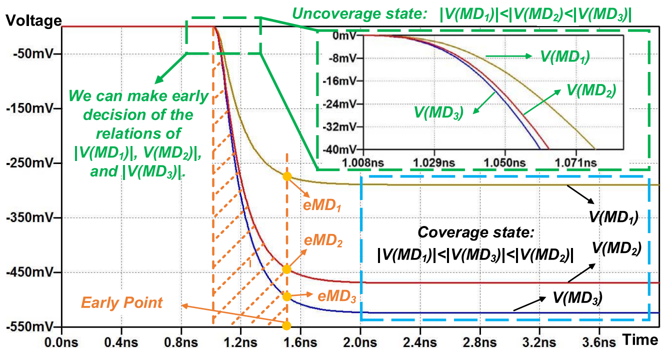


Fig. 4. Early determination in analog circuits.

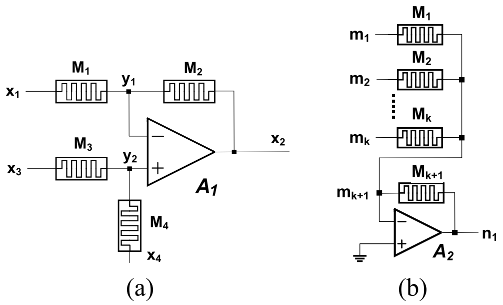


Fig. 5. Resistance tuning circuit. (a) Analog subtractor and (b) analog adder.

C. Implementation Details

1) Optimization: In the row structure, each input has an equal position to each other, and the circuit structure for each input is identical. With this character, early decision can be achieved, which means HamD and MD can process sequences with a shorter time rather than the convergence time. The detail is illustrated with MD in Fig. 4. It can be noted that the relation of $|V(MD_1)|$, $|V(MD_2)|$ and $|V(MD_3)|$ in the unconvergence state and the convergence state are the same. This feature in analog domain is extremely useful for many data mining tasks. For example, in classification we can obtain the value at the *Early Point* shown in Fig. 4. The sequence with the minimum value obtained at the *Early Point* is also the one with the minimum value obtained in the convergence state.

2) Resistance Tuning: All the resistances in the distance accelerator are memristors. Thus, resistance tuning is required to make appropriate configurations for efficient computation [21]. 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. 5. Note that resistance tuning is also performed when the configuration remains for some time as memristance leakage/drift exists in memristors. Thus, timing and power consumption will also increase slightly due to the extra configuration. Note that we do not take writing time (including wait time [25], about 1/3 to 1/4 of writing time) for resistance tuning into consideration in this paper. It should be pointed out that writing/tuning is only performed periodically with a relative large period for distance function calculation in the scenario of data centers. Thus, writing time including wait time will only increase the overall processing time slightly, and has very small influence on the performance.

For analog subtractors as shown in Fig. 5(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, we set $y_2 = 0$ and $x_1 = 0.1$. By measuring x_2, the radio of M_1/M_2 can be verified. For example, for analog subtractors in Fig. 5(a), M_1 and M_2 are set to HRS. Thus, if $x_2 = 0.1$ V, $M_1/M_2 = 1$ is configured successfully. When verifying M_3/M_4, we set $x_3 = 0.1$ V and $x_4 = 0$. By measuring y_2, the radio of M_3/M_4 can be verified. 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. 5(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 $M_1 = M_{k+1}$ is achieved. Otherwise, M_1 will be modulated according to the offset to the configuration. The process of modulation and verification can be iterated for high precision. The above tuning process for M_1 will be applied to other memristors.

3) Impact of Process Variation: Considering process variation, the actual resistance of memristors have a tolerances of ±20% to ±30%, 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 only depends on the ratio of memristor resistances. In a similar way, dynamic voltage (IR) drop will also have very limited influence on the solution quality. Thus, tolerance control technique [10] can be used to restrict the tolerance between two memristors to be lower than 1%. Second, post-fabrication resistance tuning can further reduce the negative effects of process variation.

IV. Module Evaluation

In this section, we perform module evaluations of the proposed MDA with respect to accuracy, throughput, and energy efficiency. SPICE [26] and MATLAB [15] are adopted for simulating the performance of MDA.

A. Experimental Setup

We adopt three data sets (Beef, Symbols, and OSU Leaf) from the UCR Time Series Classification Archive [18]. For each data set, we formalize the sequences with different lengths.

We implement the proposed design in SPICE [26] with the 32-nm technology node of TSMC [2], and the simulation setup is presented in Table I. Note that the choice of technology node will affect the design parameters but will not affect the circuit topology or the general conclusions to be drawn. It should be noted that we focus on the computation part in the simulation, and weights are set to 1 to make a fair comparison with existing works. It should be highlighted that different weights have

TABLE I
SPICE PARAMETERS FOR DISTANCE ACCELERATOR SETUP

Parameters	Configuration
Open loop gain of op-amp	1×10^4
Gain-bandwidth product of op-amp (GHz)	50
V_{cc}(V)	1.0
Voltage resolution	20mV
Threshold voltage of diodes (V)	0 [22]

TABLE II
PARAMETERS FOR STOCHASTIC BIOLEK'S MODEL

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%

little influence on the performance. For the sake of generality, the parameters of op-amps and diodes are set to typical values according to [22]. Particularly, a parasitic capacitance of 20fF is added to each circuit net to model the effect of parasitic capacitance [22]. The parameter voltage resolution is to translate sequence values to voltages. Considering the balance between simulation time and comparison quality, the longest sequence length is set to 40. Considering sequence length, we set the voltage resolution to 20 mV. The translation is as follows: the sequence value 1 is translated to 20 mV. Other values follow the same principle, e.g., 1.2 and −0.5 are translated to 24 mV and −10 mV, respectively. The stochastic Biolek's model [4], [25] considering nondeterministic digital dynamics for memristor simulation is adopted, and the parameters are shown in Table II where V_0 and τ are the parameters of time and voltage units, respectively, V_{T_0} is an initial dynamic stochastic threshold, $\Delta\ V$ is the voltage margin, R_{off} and R_{on} are the state parameters, and $\Delta R_{\text{on/off}}$ is the standard deviation of the $R_{\text{on/off}}$ that varies between the switching cycles.

For algorithms such as EdD, LCS, and HamD, a threshold voltage (V_{thre}) and a unit voltage (V_{step}) are used. Considering the longest sequence length is 40, we set V_{step} to 10 mV in case the output voltage overflows. Unlike V_{step}, V_{thre} is application-specific, and it is configured to 10 mV in the experiment.

The evaluation metrics are accuracy, throughput, and energy efficiency. For accuracy, the results from MDA and double-precision calculation are compared. For throughput, the convergence time in analog domain is used for evaluation, and smaller convergence time means higher throughput. Energy efficiency is defined as follows:

$$E_{\text{efficiency}} = \frac{N}{E} = \frac{N/t}{E/t} = Th/P \tag{9}$$

where N is the total sequence number, t is the runtime, E is the total consuming energy, Th is the throughput, and P is the power. Thus, we discuss energy efficiency based on throughput and power.

B. Results and Analysis

We present performance evaluation for each module of these algorithms. The convergence time indicating how fast the module can operate and the relative error are discussed. The convergence time is defined as the interval between the rising edge of the input and the timestamp when the output is within 0.1% of the final value. For each algorithm module, we randomly choose a pair of data from the same class and a pair from different classes in one dataset. The length of the time series data are converted to different lengths. Totally ten similarity computations are presented for each dataset. This process is repeated for all the three datasets.

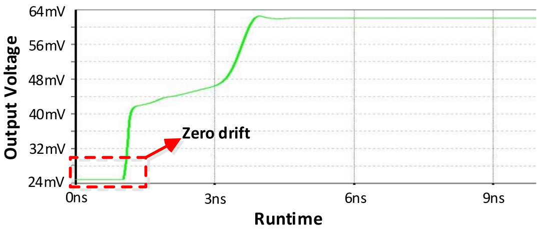


Fig. 6. Waveform of the output voltage of DTW (corresponding to the final output) computation with sequence length of 20.

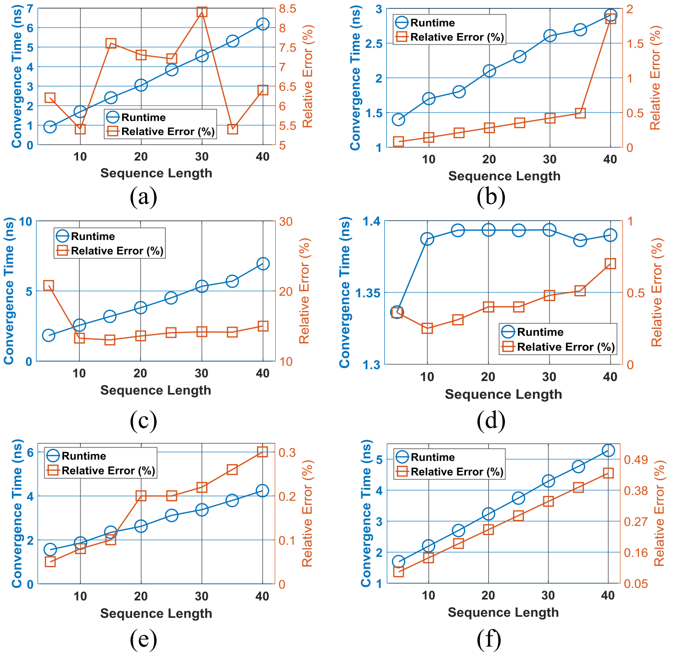
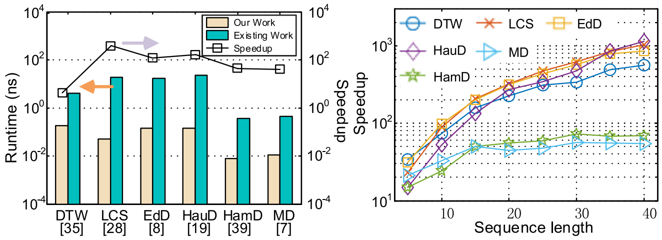


Fig. 7. Convergence time and relative error of distance functions. (a) DTW. (b) LCS distance. (c) Edit distance. (d) Hausdorff distance. (e) Hamming distance. (f) Euclidean distance.

An example of the output waveform is shown in Fig. 6. The output voltage increases gradually with the runtime, and there exists some fluctuations when it comes to convergence state. The rising speed varies because the capacity along the propagation path for each PE varies. Note that there exists zeros drifts in the calculation.

The convergence time and relative error of the six distance functions is shown in Fig. 7. We can observe that the convergence time for all distance functions are almost linear to the sequence length except for HauD. This linearity is due to the fact that the current propagation path of all the distance functions expect HauD have a linear capacitance to the input size. We can discover that the convergence time of HauD stays almost constant when the sequence length is larger than 10. This is because the convergence time is determined by the output voltage and the amount of capacitance in the current propagation path. For HauD, it should be noted that the result



(a) (b)

Fig. 8. Performance comparison of this paper and (a) existing works based on FPGAs and GPUs and (b) CPU implementation.

of each submodule is only used for the maximum calculation in the submodule right to it, whose calculation time is very short and can be ignored compared to other calculation. Thus, these submodules work almost in parallel, and the increase of sequence length has almost no effect on the runtime. With the fact that the output voltage of HauD will not increase when the sequence length increase, the convergence time of HauD stays constant basically.

Considering the relative error, it does not have a strong correlation with sequence length and is purely characterized by the property of the datasets. It can be noticed that the relative error of DTW and EdD is larger than others', which is caused by the fact that larger zero drift exists for PEs of DTW and EdD as shown in Fig. 6. This error introduced by zero drift adds a bias to the final results, which will not affect the accuracy of end-to-end applications.

In the module performance experiment, all the results are not influenced by the nondeterminism of the stochastic Biolek's model. This is due to the following two reasons. Note that in order for stochastic behavior of memristors to be significant, two conditions need to be satisfied: the voltage drop is larger than the threshold voltage, and the voltage duration is longer than the transition time [25]. First, all memristors are under a voltage far less than the threshold voltage of memristors. For DTW, the input voltages in the absolution module are very small, which are far lower than the threshold voltage of 3.0 V. 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. Other distance functions have the same situations. Second, the computation time is far less than the transition time of about 1 μs for memristors, and the running time for distance functions is about several nanoseconds. Considering the above two conditions, the possibility for stochastic resistance change is rather low with several hundreds of experiments.

C. Comparison With Existing Works

We compare our method with existing works on both GPU/FPGA and CPU platforms. The performance comparison to compute 1 million distance calculations of this paper and existing works [7], [8], [19], [28], [35], [39] on GPU/FPGA is shown in Fig. 8(a). The sequence length is set to 128. As all existing hardware accelerations and our work have a linear time complexity of the sequence length, the processing time of each element in sequences is analyzed for speedup discussion. For HamD and MD, the optimization method early determination is adopted, and the point with one-tenth convergence time is set as *Early Point*. For DTW comparison, the lower bound module for task-level optimization in work [35] is regarded as a DTW module to calculate the throughput, which is also the ideal maximum throughput. We can notice that our work has a speedup of 3.5×–376× for the six distance functions. The runtime of LCS and HamD in this paper is shorter than that of others. This is because the convergence time in analog circuits is influenced by output voltages which are smaller for LCS and HamD.

As existing works have different configurations for different applications, we also make an appropriate comparison of this paper and a CPU implementation with the same datasets. The desktop computer is with Windows 8.1 operation system and a quad-core CPU. The code is written in C language and compiled by Microsoft Visual Studio 2015. The optimization level is set to maximum speed $O2$. As shown in Fig. 8(b), our work has a speedup of 20×–1000× compared to CPU with different sequence lengths. The speedup gets larger with longer sequences. It should be noted that the speedup for HamD and MD are smaller than the other four distance functions. This is because that the time complexity of the two distance functions is $O(n)$, while that of others are $O(n^2)$.

A rough power and area analysis is presented for energy efficiency discussion. The power and area for a recently popular op-amp with a gain-bandwidth product 303GHz is 197 μW and 0.16 mm^2 [45], respectively, under 0.35 μm technology node, and the power and area for the 32-nm technology node are projected to 18 μW and 1312 μm^2, respectively, with ideal scaling for capacitance. The same procedure goes for a recent 8-bit 1.6 Gsample/s DAC [38] in 90 nm technology node, and the projected power and area for the adopted DAC are 32 mW and 0.02 mm^2, respectively. A recent 8.8GSample/s ADC in 32-nm technology node with a low power of 35 mW and an area of 0.025 mm^2 [20] is adopted. The number of PEs in each column and row is set to 128, which is the same with [35]. For sequence length larger than 128, tiling technique can be applied.

The power consumption of MDA depends on specific distance functions. Note that leakage is also included in the overall power calculation. 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 widely applied Sakoe-Chiba band constraint $R = 5\% \times n$ is adopted. The power consumption of the active op-amps is $(7R(2n - R)) \times 18$ μW $= 0.20$ W, while the power consumptions of DACs and ADCs are $\lceil \text{Throughput}_{\text{in}}/1.6 \text{ GSample/s} \rceil \times 32$ mW $= 0.13$ W and $\lceil \text{Throughput}_{\text{out}}/8.8 \text{ GSample/s} \rceil \times 35$ mW $= 0.035$ W. Assuming at least one memristor is set to HRS from the source to the ground, the power consumption of memristors is $(7R(2n - R)) \times 2 \times 10$ μW $= 0.22$ W. Thus, the total energy consumption for DTW configuration is 0.58 W. Following the same principle, the total power consumptions of the distance accelerator for LCS, EdD, HauD, HamD, and MD are 2.97 W, 6.36 W, 2.64 W, 2.95 W, and 2.16 W, respectively.

For the power consumption of the existing work, we use Xilinx Power Estimators [16] to estimate the power according to the used logical resources and clock frequency for FPGA implementations. For GPU implementations, we adopt 80% of the maximum power as the typical power. Thus, power consumptions of exiting work for DTW, LCS, EdD, HauD, HamD, and MD are 4.76 W (FPGA) [35], 240 W(GPU) [28], 175 W(GPU) [8], 120 W(GPU) [19], 150 W(GPU) [39], and 137 W (GPU) [7], respectively. Considering speedups, the improvement of energy efficiency is one to three orders of magnitudes (26.7×–8767×). Though more detailed implementation will weaken the speedup, the distance accelerator still has a higher energy efficiency.

The area of MDA is dominated by op-amps, DACs, and ADCs as there are only tens of memristors in each PE which occupy much less area than op-amps. Thus, we estimate the area of MDA with op-amps, DACs, and ADCs. The area of each PE is $10 \times 1312\ \mu\text{m}^2 = 0.013\ \text{mm}^2$, and the area of all PEs is $128^2 \times 0.013\ \text{mm}^2 = 195.19\ \text{mm}^2$. The areas for DACs and ADCs are $\lceil \text{Throughput}_{\text{in}}/1.6\ \text{GSample/s} \rceil \times 0.02\ \text{mm}^2 = 0.08\ \text{mm}^2$ and $\lceil \text{Throughput}_{\text{out}}/8.8\ \text{GSample/s} \rceil \times 0.025\ \text{mm}^2 = 0.025\ \text{mm}^2$, respectively. Thus, the estimated area of MDA is 195 mm^2, which is comparable with that of existing works [7], [8], [19], [28], [35], [39] using FPGAs and GPUs (100–400 mm^2).

V. End-to-End Evaluation

The two widely used applications, similarity search and classification are employed in the end-to-end evaluation. Specifically, the performance of MDA obtained via simulations with SPICE [26] and MATLAB [15] is compared with existing works on GPUs and FPGAs.

Considering that the highest data precision analog circuit can support is only 8 bits [31], and zero-drift error only adds bias to the final results as discussed in Section IV, we mainly discuss the accuracy of MDA for similarity search and classification applications in this section. As we focus on the performance of MDA, the involved optimization in the task level is not considered here. Therefore, the achieved speedup and energy efficiency are the same with that in Section IV-C. Note that speedup and energy efficiency are obtained with comparison with existing works with task-level optimization.

A. Experiment Setup

According to work [35], there is simply no significant difference made by reducing the dimensionality of all datasets from their original lengths to exactly 128. Thus, we apply the same operation to all datasets used in the experiments. We also set the number of column and row of PEs in MDA to both 128. For DTW, the DTW constraint of 5% is used. As the existing implementations do not support variable weighting factors, the weighting factor, σ_i, is set to 1. MATLAB is used to simulate accuracy with different data precisions. Considering data precision, data length of 8 bits is adopted for MDA, which is the highest data precision analog circuit can support [31]. Note that as discussed in Section IV-B, analog calculation produces no error about the relations of the distance values between sequences. Thus, in the experiment we focus on the accuracy loss introduced by low data precision in analog domain. For DTW with FPGA implementations, data length of 8 bits is used. For other distance functions with GPU implementations, double float precision is used.

It should be emphasized that there exists a big difference between the data precision of 8 bits in analog domain and in FPGA implementations. In FPGA implementations of DTW in [35], only inputs are with 8 bits data precision, while data precision of intermediate variables and outputs are according to computation requirement which can be much larger than 8 bits. However, in the analog domain, data precision of 8 bits means inputs, intermediate variables and outputs are all constrained to only 8 bits. Thus, the data precision of FPGA implementations is still much higher than that of analog computation. By analyzing the computation pattern of distance functions (except HauD), we can discover that the output is the sum of n values (n is the sequence length and is 128 in the experiment). In FPGA implementations, the sum of 128 8-bits numbers requires a data precision of 15 bits. However, in the analog domain, the data precision of the sum is limited by 8 bits, which will introduce serious overflow problem. Tradeoff exists for bits allocation for overflow and input data precision. More bits for overflow means that the input data precision is too low to obtain an acceptable accuracy, while more bits for input data precision will lead to serious overflow problem. In order to tackle the overflow problem, in fact we only need to keep the final result of distance functions (usually with the lowest distance value) in the range, and the overflowed values have no influences on the final results. This will largely reduce the required bits for overflow. Particularly overflow conditions are specific to distance functions and applications, which determines the corresponding bits design. In the experiment, with some test experiments our configuration is as follows.

1) Most of the input data precision for DTW and MD is 6 bits, and is 5 bits for only some datasets because six bits will cause serious overflow problem.
2) The input data precision for HauD is 8 bits as only maximum and minimum computation is involved.
3) For LCS, EdD and HamD, a threshold and a step are used, the step is set to constant 1 as the maximum distance for these algorithms is 128× step, unlike step, threshold is application specific.
4) The input data precision for LCS, EdD and HamD is 8 bits as the maximum distance is determined by step, which is set to constant 1 to eliminate overflow problem.

B. Similarity Search

Twenty datasets from UCR Time Series Classification Archive [18] are adopted. All the sequences except randomly selected one in each dataset is jointed together as the test sequence, and the selected one sequence is used as the query sequence. Similarity search is to find the subsequence from the test sequence, which has the minimum distance with the query sequence.

TABLE III
RESULTS OF SIMILARITY SEARCH WITH 20 DATASETS

Dataset	Sequence length	DTW		LCS		EdD		HauD		HamD		MD	
		Accuracy*	Diff.	Accuracy	Diff.	Accuracy	Diff.	Accuracy	Diff.	Accuracy	Diff.	Accuracy	Diff.
50words	57600	41089th 41089th	12%	14454th 41090th	1%	41090th 41090th	14%	22810th 22810th	33%	41090th 41090th	0%	41089th 41089th	4%
FaceAll	71680	128th 128th	5%	128th 128th	1%	128th 2176th	0%	55th 55th	7%	2304th 2304th	0%	2304th 2304th	10%
FacesUCR	25600	22784th 22784th	10%	22784th 22784th	0%	13312th 13312th	0%	23001th 23001th	0%	13313th 13313th	6%	13313th 13313th	13%
Haptics	19840	13311th 13311th	21%	11645th 11645th	0%	11647th 11647th	0%	19671th 19671th	2%	4485th 4484th	13%	4350th 4350th	5%
Wafer	128000	59520th 6400th	25%	53762th 53762th	0%	53762th 53762th	0%	12961th 60776th	100%	53762th 53762th	0%	53762th 48641th	34%
Inline-Skate	12800	11006th 11007th	10%	11004th 8182th	0%	11002th 10104th	0%	4531th 4531th	15%	8183th 10104th	0%	11002th 11002th	5%
Cricket_X	49920	3199th 3199th	19%	11008th 11008th	0%	11009th 11009th	0%	11299th 11312th	9%	11009th 11009th	0%	24063th 24063th	1%
Cricket_Y	49920	2048th 2048th	11%	37120th 37120th	0%	37121th 37121th	0%	3742th 3742th	1%	37121th 37121th	0%	2047th 27903th	14%
Cricket_Z	49920	47874th 47874th	5%	47873th 47873th	0%	47873th 47873th	0%	42565th 42565th	19%	47873th 47873th	0%	47873th 47873th	1%
ECG-FiveDays	2944	1786th 1790th	14%	1786th 1786th	0%	1786th 1786th	0%	1710th 1710th	3%	1786th 1786th	0%	1786th 1786th	34%
FaceFour	3072	639th 766th	18%	2047th 2047th	0%	2047th 2047th	0%	1686th 1686th	1%	2045th 2045th	7%	1440th 159th	27%
Fish	22400	4864th 3712th	21%	384th 384th	0%	384th 384th	0%	129th 7567th	14%	384th 384th	0%	8832th 4864th	3%
Gun_Point	6400	1663th 2816th	54%	1151th 1151th	0%	1151th 1151th	0%	1658th 1658th	30%	1151th 1151th	0%	1664th 1664th	43%
Star-LightCurves	128000	71168th 29503th	17%	7552th 7552th	1%	7552th 7552th	100%	7526th 5726th	14%	7552th 7552th	0%	57983th 57983th	4%
Lighting2	7680	4097th 4097th	20%	2560th 2560th	0%	2560th 2560th	0%	4097th 4097th	22%	2560th 2560th	0%	4099th 4099th	35%
Lighting7	8960	2176th 2176th	0%	2176th 2176th	0%	2177th 2177th	0%	2641th 2641th	0%	2177th 2177th	27%	2177th 2177th	2%
MALLAT	7040	6021th 6021th	9%	6016th 2816th	1%	2816th 2816th	100%	5662th 5662th	32%	2816th 2816th	0%	6016th 6016th	3%
Medical-Images	37719	26631th 34254th	9%	4059th 4059th	0%	4059th 4059th	0%	26538th 26538th	20%	4059th 4059th	0%	11682th 11682th	6%
MoteStrain	1680	502th 502th	11%	998th 500th	2%	1001th 1001th	4%	1112th 1173th	8%	1001th 1001th	0%	1005th 501th	12%
OSULeaf	25600	14592th 14592th	13%	14591th 14591th	0%	14591th 14591th	7%	3301th 14408th	14%	16000th 16000th	14%	15999th 11903th	13%

* **Accuracy means the index of the first element of the most similar subsequence in the similarity search application**. In column 'Accuracy', the upper and lower index is for MDA and existing works, respectively. Diff. means relative difference between the distances of existing works and MDA.

Table III shows the results of the similarity search task. For all the dataset, MDA can find the same subsequence with existing works with a percent of 70%, 80%, 90%, 70%, 95%, and 70% in distance function DTW, LCS, EdD, HauD, HamD, and MD, respectively. The average percent is 79%, which is still high considering the low data precision. Note that two subsequences in which the index of the first elements are near to each other are regarded as the same subsequence, e.g., for dataset Inlinestake with DTW computation, the subsequence with the first element of 11006th and another one with the first element of 11007th are assumed as the same subsequence.

It can be noted that though MDA and existing works can find the same subsequence, the relative difference maybe high, e.g., for dataset ECGFiveDays with EdD computation, the relative difference is 100%. However, the relative difference maybe low even MDA and existing works find different subsequences, e.g., for dataset 50 words with LCS computation, the relative difference is only 1% though the obtained subsequences are different. Further more, in some conditions though the relative difference is large, the real difference is low. For example, for dataset StarLightCurves with EdD calculation, the relative difference is 100%. However, the distance value is 0 for GPU and is 2 for MDA, which means 128 elements are matched for GPU and 126 elements are matched for MDA. In fact, there are only two mismatched elements for the 128 elements in the query, which is with a low error. Thus, low

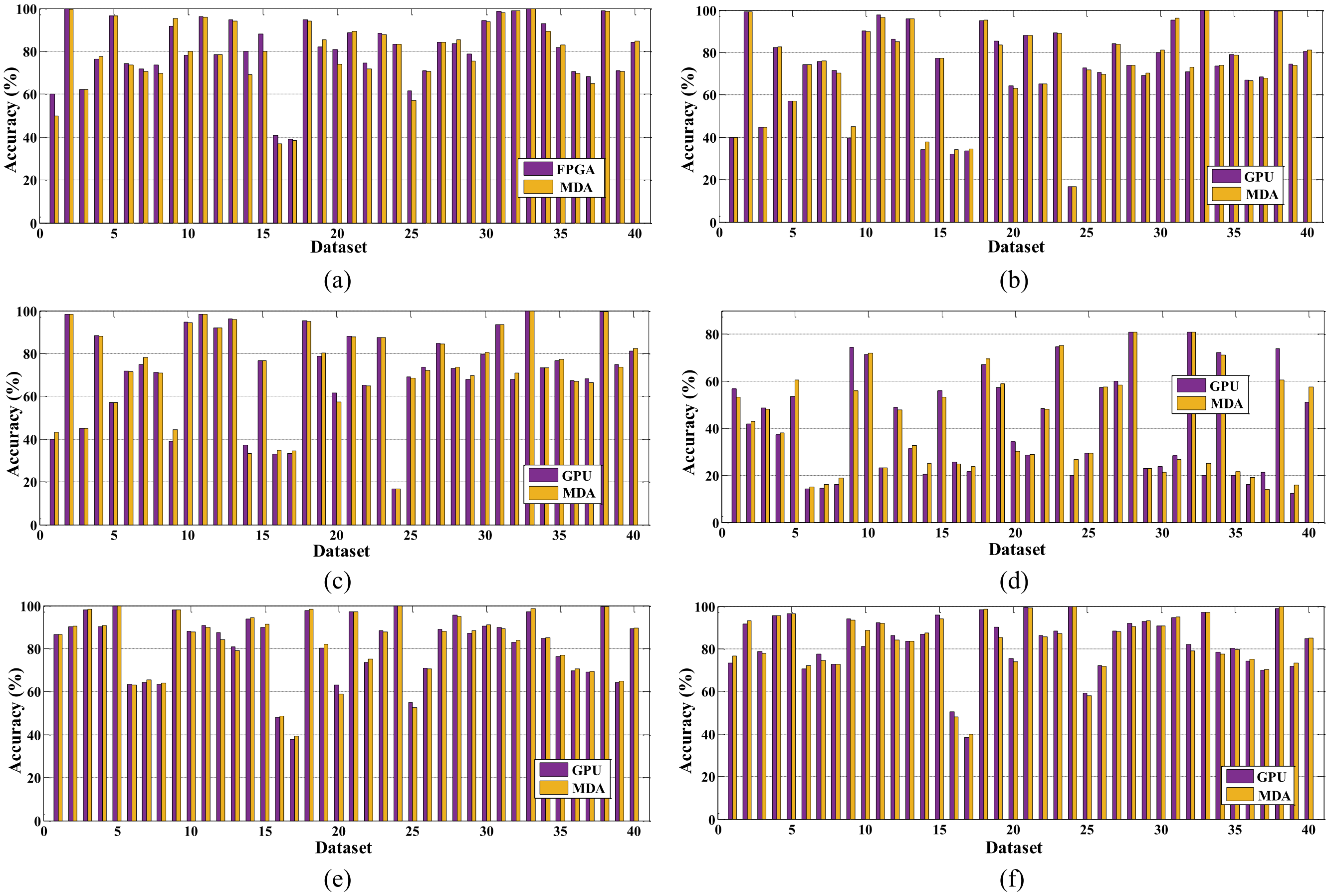


Fig. 9. Classification accuracy using *k*NN and (a) DTW, (b) LCS, (c) EdD, (d) HauD, (e) HamD, and (f) MD with 40 datasets. The correspondence of dataset and the *x*-axis is as follows: (1, Beef), (2, CBF), (3, ChlorineConcentration), (4, CinC_ECG_torso), (5, Coffee), (6, Cricket_X), (7, Cricket_Y), (8, Cricket_Z), (9, DiatomSizeReduction), (10, ECGFiveDays), (11, FaceAll), (12, FaceFour), (13, FacesUCR), (14, fish), (15, Gun_Point), (16, Haptics), (17, InlineSkate), (18, ItalyPowerDemand), (19, Lighting2), (20, Lighting7), (21, MALLAT), (22, MedicalImages), (23, MoteStrain), (24, OliveOil), (25, OSULeaf), (26, SonyAIBORobot Surface), (27, SonyAIBORobot SurfaceII), (28, StarLightCurves), (29, SwedishLeaf), (30, Symbols), (31, synthetic_control), (32, Trace), (33, Two_Patterns), (34, TwoLeadECG), (35, uWaveGestureLibrary_X), (36, uWaveGestureLibrary_Y), (37, uWaveGestureLibrary_Z), (38, wafer), (39, WordsSynonyms), (40, yoga), where X is the *x*-axis and A is the name of dataset in the format (X, A).

data precision introduces some variances to the outputs and the resulting error is relatively low.

We can discover that for specific datasets, MDA finds different similar subsequences for the adopted six distance functions. This is caused by the fact that different distance functions have their own characteristics, and choosing the distance functions is determined by applications.

C. Classification With k-Nearest Neighbors

Forty datasets from UCR Time Series Classification Archivewith [18] with different sequence lengths are selected for classification application. *k* nearest neighbor with $k = 1$ is used for classification.

Fig. 9 shows the accuracy varies with 40 datasets. It can be noticed that existing works and MDA have almost the same accuracy in most cases. Compared with existing works, the average accuracy losses for are DTW, LCS, EdD, HauD, HamD, and MD are 1.4%, −0.2%, 0.23%, 0.12%, −0.03%, 0.17%, respectively, and the overall average accuracy loss is 0.14%. It can be concluded that MDA introduces almost no accuracy loss. We can also find that though most of the distance function introduce errors, LCS obtains a relatively high accuracy improvements (0.2%). This is highly caused by the fact that for LCS low data precision removes noises in the input data and therefore obtains high accuracy.

However, there still exists some cases that the accuracy loss is relatively large, e.g., for dataset 1 with DTW calculation, the accuracy loss is 10%. This is because dataset 1 is more sensitive on the input data precision for DTW. In practical uses, this problem can be solved by changing the adopted distance function for dataset 1. For example, MDA using HamD gets a much higher accuracy (86%) than GPU using HamD (86%), FPGAs using DTW (60%) and MDA using DTW (50%) for dataset 1.

We can also notice that for specific dataset, the accuracy of distance functions vary. For example, the accuracy of dataset 24 using LCS and EdD is lower than 20%. However, MD can achieve a high accuracy of 100%. This is due to the fact that choosing distance functions is specific to applications. Considering such specification, we can select suitable distance functions for application for comparison, and the accuracy loss will be even lower.

VI. Conclusion

In this paper, we propose MDA, a reconfigurable high-throughput and high-energy-efficient memristor-based distance accelerator for time series data mining in data centers. We adopt memristors to design analog circuits for six widely used distance functions including DTW, LCS, Hausdoff distance, EdD, HamD, and MD. The basis primitive of the circuits is extracted, which can be configured to any specific distance functions. Comprehensive experiments are presented with public available datasets. Compared with existing works, the performance of the proposed accelerator has a speedup of 3.5×–376× with limited accuracy loss. Energy analysis shows that the accelerator has an improvement of 1–3 orders of magnitude on energy efficiency. Though the data precision for MDA is low, there is little accuracy loss is for similarity and classification applications.

The future works will evaluate the effeteness and efficiency of the proposed MDA considering more detailed fabrication issues (e.g., defects) and runtime issues (e.g., memory reliability, wait time) with a computer-system architecture simulator (e.g., Gem5 [1]). The detailed design of I/Os of the proposed MDA for long sequences also needs to be investigated and analyzed in the future. Furthermore, we will improve the PE design to support more functions.

References

[1] (2018). *The gem5 Simulator*. [Online]. Available: http://www.gem5.org

[2] (2018). *TSMC*. [Online]. Available: http://www.tsmc.com

[3] A. Adshead. (9, 2014). *Data Set to Grow 10-Fold by 2020 As Internet of Things Takes Off*. [Online]. Available: computerweekly.com

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

[5] A. Banerjee and J. Ghosh, "Clickstream clustering using weighted longest common subsequences," in *Proc. SIAM*, Chicago, IL, USA, 2001, pp. 33–40.

[6] N. Z. Bawany and J. A. Shamsi, "Smart city architecture: Vision and challenges," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 11, pp. 246–255, 2015.

[7] D.-J. Chang, A. H. Desoky, M. Ouyang, and E. C. Rouchka, "Compute pairwise Manhattan distance and Pearson correlation coefficient of data points with GPU," in *Proc. IEEE SNPD*, 2009, pp. 501–506.

[8] R. Farivar, H. Kharbanda, S. Venkataraman, and R. H. Campbell, "An algorithm for fast edit distance computation on GPUs," in *Proc. IEEE InPar*, San Jose, CA, USA, 2012, pp. 1–9.

[9] T. S. Han, S.-K. Ko, and J. Kang, "Efficient subsequence matching using the longest common subsequence with a dual match index," in *Proc. Int. Workshop Mach. Learn. Data Min. Pattern Recognit.*, Leipzig, Germany, 2007, pp. 585–600.

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

[11] M. Imani, S. Gupta, A. Arredondo, and T. Rosing, "Efficient query processing in crossbar memory," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Taipei, Taiwan, 2017, pp. 1–6.

[12] M. Imani, D. Peroni, A. Rahimi, and T. Rosing, "Resistive CAM acceleration for tunable approximate computing," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2016.2642057.

[13] *Cisco Data Center for Healthcare*, Cisco Inc., San Jose, CA, USA, 2016.

[14] *Gartner Says the Internet of Things Will Transform the Data Center*, Cisco Inc., San Jose, CA, USA, 2014.

[15] (2017). *MathWorks Inc*. [Online]. Available: https://www.mathworks.com/

[16] (2016). *Xilinx Inc*. [Online]. Available: https://www.xilinx.com

[17] 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.

[18] 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

[19] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber, "Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer," *Vis. Comput.*, vol. 26, nos. 6–8, pp. 1007–1016, 2010.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] Y. Lu, C. L. Tan, W. Huang, and L. Fan, "An approach to word image matching based on weighted Hausdorff distance," in *Proc. DAR*, Seattle, WA, USA, 2001, pp. 921–925.

[25] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, "More efficient testing of metal-oxide memristor–based memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 6, pp. 1018–1029, Jun. 2017.

[26] L. W. Nagel *et al.*, "Simulation program with integrated circuit emphasis (SPICE)," Electron. Res. Lab., College Eng., Univ. California, Berkeley, CA, USA, Memorandum Rep. ERL-M382, 1973.

[27] F. M. Oliveira-Neto, L. D. Han, and M. K. Jeong, "Online license plate matching procedures using license-plate recognition machines and new weighted edit distance," *Transp. Res. C Emerging Technol.*, vol. 21, no. 1, pp. 306–320, 2012.

[28] A. Ozsoy, A. Chauhan, and M. Swany, "Fast longest common subsequence with general integer scoring support on GPUs," in *Proc. ACM PMAMM*, 2014, p. 92.

[29] V. Perlibakas, "Distance measures for PCA-based face recognition," *Pattern Recognit. Lett.*, vol. 25, no. 6, pp. 711–724, 2004.

[30] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD*, Beijing, China, 2012, pp. 262–270.

[31] 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.

[32] M. Saremi, "Carrier mobility extraction method in CHGs in the UV light exposure," *Micro Nano Lett.*, vol. 11, no. 11, pp. 762–764, Nov. 2016.

[33] M. Saremi, "A physical-based simulation for the dynamic behavior of photodoping mechanism in chalcogenide materials used in the lateral programmable metallization cells," *Solid State Ionics*, vol. 290, pp. 1–5, Jul. 2016.

[34] M. Saremi, H. J. Barnaby, A. Edwards, and M. N. Kozicki, "Analytical relationship between anion formation and carrier-trap statistics in chalcogenide glass films," *ECS Electrochem. Lett.*, vol. 4, no. 7, pp. H29–H31, 2015.

[35] D. Sart, A. Mueen, W. Najjar, E. Keogh, and V. Niennattrakul, "Accelerating dynamic time warping subsequence search with GPUs and FPGAs," in *Proc. ICDE*, Sydney, NSW, Australia, 2010, pp. 1001–1006.

[36] J. J. Tithi, N. C. Crago, and J. S. Emer, "Exploiting spatial architectures for edit distance algorithms," in *Proc. ISPASS*, Monterey, CA, USA, 2014, pp. 23–34.

[37] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77–97, 1st Quart., 2014.

[38] 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.

[39] N. A. Vandal and M. Savvides, "CUDA accelerated iris template matching on graphics processing units," in *Proc. IEEE BTAS*, Washington, DC, USA, 2010, pp. 1–7.

[40] Z. Wang *et al.*, "Accelerating subsequence similarity search based on dynamic time warping distance with FPGA," in *Proc. ACM FPGA*, Monterey, CA, USA, 2013, pp. 53–62.

[41] G. Weng, T. He, M. Chen, and Y. Shen, "The vehicle's classification recognition system based on DTW algorithm," in *Proc. IEEE WCICA*, Hangzhou, China, 2004, pp. 4169–4171.

[42] X. Xu *et al.*, "Accelerating dynamic time warping with memristor-based customized fabrics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 4, pp. 729–741, Apr. 2018.

[43] X. Xu, D. Zeng, W. Xu, Y. Shi, and Y. Hu, "An efficient memristor-based distance accelerator for time series data mining on data centers," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Austin, TX, USA, 2017, pp. 1–6.

[44] L. Zhang, Y. Zhang, J. Tang, K. Lu, and Q. Tian, "Binary code ranking with weighted Hamming distance," in *Proc. CVPR*, Portland, OR, USA, 2013, pp. 1586–1593.

[45] 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.

[46] 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.

[47] G. Deng *et al.*, "Scalable and parameterized dynamic time warping architecture for efficient vehicle re-identification," in *Proc. 4th IEEE Int. Conf. Transp. Inf. Safety (ICTIS)*, 2017, pp. 48–53.

[48] L. Zhang, D. Li, X. Zou, Y. Hu, and X. Xu, "Scalable and parameterized architecture for efficient stream mining," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 101, no. 1, pp. 219–231, 2018.

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 Ph.D. degree from the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN, USA, in 2015.

He is currently a Professor with the Institute of Cyberspace Research, College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He was an Assistant Professor with the University of Colorado Denver, Denver, CO, USA, a Research Scientist with the State University of New York at Buffalo, Buffalo, NY, USA, and an Engineer with Alcatel-Lucent (currently, Nokia), Boulogne-Billancourt, France. His current research interests include mobile sensing, Internet of Things security, trustworthy sensing, biometrics, and smart health.

Dr. Lin was a recipient of the Best Paper Award in 2017 IEEE BHI Conference and the Best Demo Award in 2018 ACM HotMobile Conference.

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.

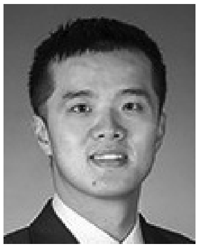
He is currently an Assistant Professor with the Computer Science and Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA. He owned five licensed U.S. and international patents, and has authored over 70 peer-reviewed journal and conference papers. His current research interests 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.

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 Internet of Things.

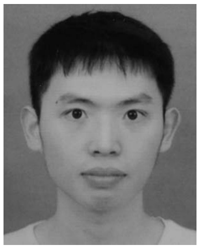
Dr. Yao has served on technical program committees of many IEEE/ACM conferences. He is a member of ACM.



Yiyu Shi (S'06–M'10–SM'15) received the B.S. degree (Hons.) 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.



Dewen Zeng received the B.S. degree in electronic science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2016, where he is currently pursuing the master's degree with the School of Optimal and Electronic Information.

His current research interests include data mining and embedded computing.

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 was a co-recipient of the Best Contribution Award at International Workshop of Logic and Synthesis 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.