In-memory Search with Memristors for Highly Efficient Similarity-Measurement-Based Data Mining

Yi Li (liyi@hust.edu.cn)
Huazhong University of Science and Technology https://orcid.org/0000-0003-1930-8550
Ling Yang
Huazhong University of Science and Technology https://orcid.org/0000-0002-5727-688X
Xiao-Di Huang
Huazhong University of Science and Technology
Houji Zhou
Huazhong University of Science and Technology
Yingjie Yu
Huazhong University of Science and Technology https://orcid.org/0000-0001-7813-7064
Han Bao
Huazhong University of Science and Technology
Jiangcong Li
Huazhong University of Science and Technology
Shengguang Ren
Huazhong University of Science and Technology https://orcid.org/0000-0001-6966-1542
Feng Wang
Huazhong University of Science and Technology
Lei Ye
Huazhong University of Science and Technology https://orcid.org/0000-0001-5195-1867
Yuhui He
Huazhong University of Science and Technology
Jia Chen
AI Chip Center for Emerging Smart Systems https://orcid.org/0000-0003-4814-5420
Guiyou Pu
Huawei Technologies Co., Ltd. https://orcid.org/0000-0002-6014-9806
Xiang Li
Huawei Technologies https://orcid.org/0000-0001-6457-583X
Xiang Shui Miao
School of Optical and Electronic Information, Huazhong University of Science and Technology https://orcid.org/0000-0002-3999-7421

Article

Keywords:

Posted Date: April 26th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1529611/v1

Abstract

Similarity search, that is, finding similar items in massive data, is a fundamental computing problem in many fields such as data mining, and information retrieval. However, for large-scale and high-dimension data, it suffers from high computational complexity, requiring tremendous computation resources. Here, based on the one-selector-one-resistor memristors, for the first time, we propose an in-memory search (IMS) system with two innovative designs. First, by exploiting the natural distribution law of the devices resistance, a hardware local sensitive hash encoder has been designed to transform the real-valued vectors into more efficient binary codes. Second, a compact memristive ternary content addressable memory is developed to calculate the Hamming distances between the binary codes in parallel. Our IMS system demonstrated a 168× energy efficiency improvement over all-transistors counterparts in clustering and classification tasks, while achieving a software-comparable accuracy, thus providing a low-complexity and low-power solution for in-memory data mining applications.

Introduction

Similarity search is an essential problem in data mining[1], it includes tasks such as finding the top-*k* most similar documents from a database or identifying the class of an image by computing the distances between the query object and the stored data. In practice, the data are usually organized in the form of feature vectors, and the similarity of the vectors is generally described by the distance between them. However, the complexity of distance computing grows dramatically with the increasing data volume *v* and vector dimensionality *d*. For instance, $v \times d$ floating-point multiplications are required to compute the Euclidean distance and Cosine distance, two widely used distance functions for real-valued vectors. That involves massive memory accesses, resulting in huge power consumption and long latency when the *v* and *d* are large. The challenge is known as the curse of dimensionality[2].

One promising approach is to perform the search in computational memory. Recently, in-memory computing (IMC) based on emerging non-volatile memories such as memristors, phase change memory, and ferroelectric field effect transistor, has demonstrated unprecedentedly high-energy-efficiency in data-intensive computation such as machine learning[3-10], scientific computing[11-13], and image processing[14], owing to its natural parallelism to perform vector matrix multiplication *in situ*[15-20] based on Ohm's law and Kirchhoff's law. Unlike the crossbar serving as the one-step vector-matrix multiplication circuit, non-volatile content addressable memory (CAM) and the specific ternary CAM (TCAM) have been proposed to perform search operations with a O(1) complexity[21]. However, most of the proposed CAMs still rely on transistors. That makes it hard to downscale the cell area and improve the integration density, even for the most compact two-transistor-two-resistor (2T2R) structure. Besides, CAM only supports the binary Hamming distance (HD), which is a low-complexity but low-precision distance function. Euclidean and Cosine distances cannot be performed on the CAM. Therefore, the CAM has essential limits when used in the search architecture[22].

In this work, we extend the IMC concept to more specific similarity search problems and propose a highly efficient in-memory search (IMS) hardware architecture for similarity-measurement-based data mining applications, as shown in **Fig. 1**. The IMS system contains two basic parts, and both can be implemented on the memristor arrays. The first is a hardware encoder. By exploiting the Gaussian distribution of the memristor resistance, the random projection-based local sensitive hash function can be naturally realized on the array. In this step, the real-valued vectors are transformed into binary codes without distortion of the data similarity relations, implying that the Euclidean distance and Cosine distance can be approximately calculated on the CAM. The second part is a compact and fully passive TCAM consisting of two one-selector-one-resistor (1S1R) memristors only. By replacing the three-terminal transistor with the naturally integrated two-terminal volatile selector, not only can the integration density be increased but also the overall resistance can be improve, thus reducing the search energy. Our IMS system is another showcase of in-memory computing for the future storage class memory-based high-performance search engine.

Results

In-memory search system using 1S1R memristor array

The similarity search is implemented with a memristive system as shown in Fig. 2a (the circuit diagram and photograph of the hardware system are shown in fig. s1). This system consists of an encoder used to map real-valued vectors into binary codes and a TCAM circuit used to calculate the HD between the query and stored data. In this work, we built the IMS system using V/HfO_x/HfO_2/Pt 1S1R devices organized in a 32×32 crossbar array (Fig. 2b). As illustrated in the cross-sectional transmission electron microscopy image of the device stacked structure (Fig. 2c), the interfacial VO_x layer between V and HfO_x severs as a volatile selector with its highly reproducible insulator- metal-transition characteristics[23]. Besides, the VO_x-based selector contributes to higher resistance and enhances the low resistance state (LRS) of the whole device to a sub-high resistance region (90 kΩ ~ 243 kΩ) while in the off state ($V<|V_{th,s}|$, range C and D), as depicted in the typical 1S1R I-V curves of 100 cycles (Fig. 2d). The higher overall resistance can reduce the search energy compared to the pure-binary memristors. Meanwhile, the 1S1R device maintains a stable large resistance ratio of 17.3 between the high resistance state (HRS) and sub-high resistance even after 10^{10} times of switching and 10^4 s retention (Fig. 2e-f). That provides a wide and reliable sense margin for the TCAM (more details about the device can be found in fig.s2 in the Supplementary Information). In addition, the switching energy of SET/RESET can achieve 17.5 fJ/538 fJ, and the read energy is 0.23 fJ/0.013 fJ in the LRS/HRS, enabling an ultralow-power feature for the IMS system.

Memristor based hardware local sensitive hash encoder

Local sensitive hash (LSH) is widely used in dimensionality reduction for the approximate nearest neighbour search, and is usually described by a random projection operation expressed by Eq. (1), where x is the real-valued feature vector with a dimensionality of d×1, W is an n×d Gaussian random projection matrix whose elements follow the Gaussian distribution with a mean of 0, b is a random offset vector with a size of n×1, and y is the binary code obtained from the sign function that outputs 1 when Wx + b is positive otherwise outputs 0. For simplicity, in this study, W also represents [W b] and x also represents $[x\ 1]^T$ (further discussion can be found in fig.s3 in the Supplementary Information).

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) = \operatorname{sign}\left(\begin{bmatrix}\boldsymbol{W} & \boldsymbol{b}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ 1\end{bmatrix}\right)$$
$$\boldsymbol{y}_i = \begin{cases} 0 & \boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{b}_i \le 0 \\ 1 & \boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{b}_i > 0 \end{cases} \qquad (1)$$

According to the experimental results, the conductance values in HRS of the 1S1R devices follow a lognormal distribution, as shown in **Fig. 3a**. That is consistent with the results of prior studies[24, 25]. In addition, we found the conductance difference (G^+ - G^-) of any pair of devices follows a Gaussian distribution with a mean of near-zero (Fig. 3b, more proofs about the distribution can be seen in fig. s2). Therefore, the conductance difference matrix $\boldsymbol{G}$ of a 1S1R array in HRS could be regarded as a natural Gaussian random matrix $\boldsymbol{W}$. Of course, the matrix $\boldsymbol{W}$ can be also represented with the LRS array because the LRS follows the Gaussian distribution[25], and thus the conductance difference (G^+ - G^-) of LRS follows the Gaussian distribution too. For lower computing energy, HRS are adopted in this work. According to the results, a memristive random projection circuit is designed, as shown in Fig. 3c. The conductance difference of each two rows of the device represents a row vector ($\boldsymbol{G}_i^+ - \boldsymbol{G}_i^-$). The $\boldsymbol{x}$ is mapped to the voltage pulses $\boldsymbol{V}_x$ applied to the array in parallel and the output differential currents ($\boldsymbol{I}_i^+ - \boldsymbol{I}_i^-$) are the result of ($\boldsymbol{G}_i^+\boldsymbol{V}_x - \boldsymbol{G}_i^-\boldsymbol{V}_x$) according to Ohm's law and Kirchhoff's current law. Subsequently, by connecting the two

differential rows on the same comparator, the comparators are used to implement the difference and sign function at the same time. After the above process, an *n*-bit binary code $\boldsymbol{V}_y$ is finally obtained according to the circuit Equation (2). The advantage of the hardware encoder is that the functions of the random number generator and vector-matrix multiplier are realized simultaneously in one step in the same circuit.

$$\boldsymbol{V}_y = \mathrm{sign}\left(\boldsymbol{G}^{+}\boldsymbol{V}_x - \boldsymbol{G}^{-}\boldsymbol{V}_x\right) \quad (2)$$

Figure 3d illustrates the encoding process of the iris dataset, which contains 150 4×1 feature vectors for three types of irises (setosa, versicolor and virginica). For simplicity, only the features of calyx width, petal length, and petal width are selected and an additional 150×1 vector whose elements are ones is grouped with the 150×3 feature matrix for offset. The grouped feature vectors are mapped into the analogue voltages, and are then applied to the four rows of the array. The 150 16-bit codes V_y are obtained with the 4×32 memristor array and the comparators. However, it can be observed that there are many common bits in these raw codes (more detailed information can be found in fig. s3 in the Supplementary Information), which are redundant for distinguishing the different types of irises and will lead to the waste of computing resources.

To obtain more compact codes, a common bits compression (CBC) approach is proposed in this work. The non-common bits were extracted by bitwise accumulation. Since the distributions of 0 and 1 are significantly unbalanced, the sum of the common bits will be outside the range [bottom threshold (TH_B), top threshold (TH_T)]. Only the bits whose sum locates in the range [TH_B, TH_T] remain. After the CBC, the final codes consist of only 4 bits (reduced from 16 bits). With the 4-bit codes, the setosa was successfully distinguished. However, the versicolor and virginica are too similar to be distinguished with the short codes, implying a longer code length is required.

To verify the similarity-preserving capability, we simulated the encoder with all four features of the iris datasets. Figure 3e and Fig. 3f illustrate the similarity matrix of original real-valued features based on Euclidean distance and the similarity matrix of 32-bit codes based on normalized HD, respectively. It is obvious that the similarity matrices based on the HD also are able to accurately describe the similarity relationship among the data. That indicates the similarity information of the data is not distorted after encoding. This can be also proved by the preservation of the inter-class distance and intra-class distance as shown in Fig. 3g. It clearly shows that similar items in Euclidean space will still have a shorter distance in Hamming space. According to our simulation results, for the iris dataset, the effective code length *m* after compression is approximately a quarter of the length of raw code ($m = n / 4$), as shown in Fig. 3h. In addition, Cosine distance can also be replaced by HD with the same operations with a function without the offset b (fig. s5). In short, we have proved that by using the hardware encoder and CBC-LSH scheme, the Euclidean distance and Cosine distance could be replaced by the HD and calculated on the binary CAM approximately.

2S2R TCAM for Hamming distance computing acceleration

Instead of the Euclidean distance and Cosine distance, for the binary data, HD is the more efficient distance metric, which can be computed fast with bitwise operation. The calculation of HD is given by an XOR-accumulation operation expressed by Eq. (3), where Q_i / P_i denotes the i^{th} bit of the *m*-bit code *Q*/*P*. TCAMs based on non-volatile devices have shown a powerful performance in memory-centric applications such as pattern matching[26, 27], data query[28], tree-based machine learning[29, 30], and the memory augmented neural network[31, 32] owing to its ultra-high parallelism to perform the HD-based search. Most of the prior memristor-based TCAM structures are still limited by the small sense margin and large area overhand[33, 34]. The former is dependent on the devices HRS/LRS ratio, and the latter results from the use of transistors. Additionally, the search energy is also a key performance metric. Accordingly, our 1S1R device has shown a stable large HRS/LRS ratio and compact integrated structure to provide a sufficient sense margin and a small cell footprint. In addition, the high overall resistance of the 1S1R device enables the low search energy of TCAM.

$$HD(Q,P)=\sum_{i=1}^{m}Q_i\oplus P_i$$

3

Figure 4a schematically illustrates the structure of the proposed 2S2R TCAM and the corresponding definition of the states. For the data *P* stored in TCAM, 1 and 0 are defined as HRS-LRS and LRS-HRS, respectively, while for the query data *Q*, 1 and 0 are converted into two search voltage (V_s, 0.1 V) pulses (V_s-0 and 0-V_s), applied on the left search line (SL) and right search line ($\overline{\mathrm{SL}}$), respectively. For the *don't care* state X, *P* is defined as HRS-HRS. When *Q* = X, the two SLs of the corresponding column will be floated. The voltage sense amplifier (SA) is used to sense the voltage of the ML (V_{ML}) and output digital voltage by comparing the V_{ML} to a reference voltage (V_{REF}). In this cell structure, the ML is hardly charged while the query data *Q* matches with the stored data *P*. For instance, a 0 was written into the TCAM cell where R_1 is at the LRS and R_2 is at the HRS. To search 0, the SL (LRS device) and $\overline{\mathrm{SL}}$ (HRS device) are applied with 0 V and V_s respectively. The voltage of ML is pulled down to a level of approximately 0, which denotes the match case, as shown in Fig. 4b. On the contrary, while searching 1, the V_s is applied on the SL and 0 V is applied to the $\overline{\mathrm{SL}}$. Therefore, the V_{ML} is pulled up to a level close to the V_s, which denotes the mismatch, as shown in Fig. 4c.

For a single cell of TCAM, V_{ML} represents the result of XOR (*Q*, *P*). Further, for a TCAM array with *m* cells in each row, the V_{ML} corresponds to the HD between the *m*-bit stored data *P* and the query data *Q*, while the *m* search voltages are applied to the SLs simultaneously (S3). Figure 4d illustrates the structure diagram of the TCAM array. The detailed circuit structure and the relationship between the V_{ML} and HD are illustrated in fig.s6. For such a TCAM, before applying the search voltage to the SLs, the MLs must be initialized by discharging (V_{DCH}). The search voltages are applied to the SLs and the MLs are charged to the corresponding voltages depending on the number of matching bits between the stored data *P* and query data *Q*. Subsequently, the SA starts work by applying the enable signal (V_{SAEN}) and generates a digital output representing the result of match or mismatch according to the V_{REF}. If the V_{ML} is lower than V_{REF} the node *out* will output a high voltage near V_{DD} (match case); otherwise, remain the low voltage close to 0 (mismatch case). To verify the circuit, simulations were performed based on a 32×32 TCAM array with 32 cells on each ML according to the experimental results. The simulation results of the 32-bit TCAM are shown in fig.s6b and fig.s6c in the Supplementary Information. For clarity, here, we demonstrate the search operation of the 8-bit code 11111111. The TCAM stores nine codes, including 00000000, 00000001, ..., and 11111111. The voltage signals of query data (11111111) were then applied to the array. Figure 4e shows the simulation results of the search operation in the 1-bit mismatch (01111111, HD = 1) and full match (1111111, HD = 0) cases. After initialization, the search operation was completed within 6 ns. Figure 4f shows the voltage signals of the nine MLs, which denote the match degree of the query and stored data. These can be clearly distinguished and easily sensed by SA or analogue-digital converters.

As an essential performance merit, the sense margin (ΔV_{ML}, defined as the ML voltage difference between the cases of a full match and a 1-bit mismatch) for the 2S2R TCAM is mainly dependent on the HRS/LRS ratio, cells number per ML, and V_s (given by equation S9). Figure 4g depicts the relationship between the sense margin and the HRS/LRS ratio in different array sizes. It can be observed that to obtain a large sense margin, a small array and large HRS/LRS ratio are required. In addition, due to the resistance variations, the actual sense margin is usually narrower than in the ideal case as shown in fig. s6d. To improve the robustness of the TCAM, we also carefully calculated the best reference voltage for the SA (more detail analysis can be seen in fig. s6d).

Another performance merit is search delay, associated with the parasitic capacitance of the ML (C_{ML}) and the resistance of the devices. According to the equivalent RC model of the 2S2R TCAM (fig.s7), the search delay is dominated by the LRS and C_{ML}. The C_{ML} is dependent on the materials and the wire feature size, thus cannot be tuned once fabricated. But the

device resistance can be reprogrammed to meet various demands in practice. The results in Fig. 4h suggest that a lower LRS can provide a shorter search delay whereas results in higher search energy and severe IR-drop (fig.s8). Therefore, there is a trade-off between the search delay and power consumption. In addition, the case of the X-match was also considered and successfully demonstrated (fig. s9).

Figure 4i shows the comparison of the 2S2R TCAM with previous counterparts (more details can be found in table.s1in the Supplementary Information). Our passive TCAM shows significant improvement in terms of cell area, and search energy. First, the cell area of our TCAM is $8F^2$ ($4F^2 \times 2$) in the ideal case. The previous study has achieve a $16.3F^2$ cell area in 2-diode-2-resistor (2D2R) structure[35].Seconde, owing to the existence of the selector, the 1S1R device shows a sub-high resistance after being SET-switched to LRS when reading it with a voltage (V_s, 0.1 V) lower than $V_{th,s}$ (~ 1 V). That enables ultra-low search energy of 0.25 fJ / bit per search, although the higher resistance also results in a longer search latency of about 6 ns. Third, if the 2S2R TCAM are used in speed-first scenarios, it can work in the R mode by applying a search voltage higher than the $V_{th,s}$ to turn-on the selector. In this case, the LRS resistance is the actual value of the memristor as shown in Fig. 2a. With the lower LRS resistance, the search latency can be further reduced according to the results in Fig. 4h.

In-memory similarity search for clustering and classification

K-means clustering and *k*-nearest neighbour (*k*-NN) classification are two of the most important and typical similarity-measurement-based algorithms in data mining. In this work, the nearest similarity search based on HD is described by Eq. (4), where P_i and $P[i]$ denote the i^{th} data in the database. Based on Eq. (4), the classification can be described with Eq. (5), which is the key operation of *k*-means and *k*-NN, where $Class_{pred}$ is the classification result.

$$Q_{pred} = P\left[\underset{i\in\{1,\ldots,v\}}{\arg\min}\ HD\left(Q, P_i\right)\right] \quad (4)$$

$$Class_{pred} = \underset{i\in\{1,\ldots,c\}}{\arg\min}\ HD\left(Q, P_i\right) \quad (5)$$

To find the item Q_{pred}, whose distance to the query data Q is the smallest from the stored data P, an adjustable threshold scheme[40] was adopted in this study. As shown in Fig. 5a, based on the TCAM circuit, the V_{REF} applied to the SA was changed to a scanning voltage from 0 to V_s, instead of a constant voltage, where the row whose output state flips first is the nearest object (fig. s10). Based on such a scheme, we demonstrated *k*-means clustering with the iris dataset, which is encoded with different code lengths. With longer code lengths, it can achieve a higher average search accuracy and robustness (in Fig. 5b). An accuracy of 87.9% was achieved when the code length is 16 bits. It is very close to the software accuracy of 88.7% based on Euclidean distance, even within three iterations, as shown in Fig. 5c. The inset graph shows the average number of iterations required for different code lengths, indicating that *k*-means requires more iterations for longer codes. To further investigate the performance of the IMS system in the high-dimensional space, we simulated the *k*-means with the ISOLET dataset which contains 617 phonetic features of 26 alphabets from 300 subjects, and it achieves an accuracy of 60.2% with 1536-bit codes, which is even slightly higher than the accuracy of software method based on Euclidean distance. As shown in Fig. 5d, compared with hyperdimensional computing[41], our IMS system exhibited great superiority, especially in terms of code length. For low-dimensional data (iris), hyperdimensional computing still requires 10,000 bits to achieve an accuracy close to the Euclidean distance scheme, whereas our scheme requires only 16 bits. For

high dimensional data (ISOLET), hyperdimensional computing clustering is nearly invalid (33.1% accuracy). In contrast, our scheme shows excellent performance (60.2% accuracy) even with a shorter code length.

Furthermore, we demonstrated the *k*-NN classification with our IMS system. The classification accuracy on the iris dataset for different code lengths is shown in Fig. 5e. The 32-bit codes can provide an accuracy of 95.9% with the hardware implementation, which is very close to the software accuracy of 96.1% based on Euclidean distance. To investigate the robustness of our scheme, Monte Carlo simulations concerning the D2D resistance variation and device HRS / LRS ratio were performed, as shown in Fig. 5f. Note that, while the device HRS / LRS ratio is larger than 10, the *k*-NN is almost insensitive to variations within 200% (fig.s11). Moreover, we simulated the *k*-NN with a random subset of the Mixed National Institute of Standards and Technology (MNIST) handwritten digits datasets where each 28×28 picture is a 784-dimension expansion vector. The accuracy with different *k* is shown in fig. s12, which exhibits high performance even at a 500-bit code length, with an accuracy of 89% which is higher than the maximal accuracy obtained by *k*-NN based on Euclidean distance (86%). In addition, for the high-dimensional sparse text vectors (7599 dimensions) using Cosine distance, high accuracy of 97% in BBC sports news classification was obtained only with 2000-bit codes (fig. s13). These results also prove that the CAM can approximately calculate the Euclidean and Cosine distances after transforming the real-valued vectors into binary codes using the proposed CBC-LSH.

To reduce the computing complexity while querying an unknown object in the database, classification is performed first by identifying the nearest class centroid which was calculated a priori with *k*-means. A *k*-NN search is then performed over the data belonging to the same class as the query object, as shown in Fig. 5g and 5h. We simulated this two-stage search process with the iris dataset, where the three centroids were calculated and stored in the first-stage TCAM array, all samples in the same class are stored in the same array at the second stage, and the NN search and the *k*-NN search were implemented by adjusting the V_{REF}. This method can avoid the aimless search in the entire huge database and significantly reduce the search energy and delay[35].

Discussion

Our IMS has an extremely low computation complexity because it needs only some read operations during the computing process instead of the complex floating-point multiplications and additions as shown in Table 1 (the detail complexity calculation is shown in table.s5 in the Supplementary Information). Although some previous studies have proposed efficient methods to calculate the Euclidean distance and Cosine distance using analogue memristors, yet still requires a complex write-verify algorithm to map the values on the devices[42–44]. Besides, another benefit of binary computing is that it is better compatible with current technology both in hardware and operation. This can avoid using a large number of digital-analogue/analogue-digital converts in circuits.

Table 1
Performance comparisons between different implementations for search.

		In-memory HDC[31, 45]	All-CMOS Euclidean distance	All-CMOS LSH[46]	This work
Encoder	Operation	Memory Read	NA	Floating-point operation	Memory Read
	Energy (nJ)	420.80	0	0.385	0.01975
Distance computing	Operation	Memory Read	Floating-point operations	Bitwise XOR and popcount	Memory Read
	Energy (nJ)	9.44	3.61	0.019	0.00173
Total energy		430.24 (20032.59×)	3.61 (168×)	0.404 (18.8×)	0.02148 (1×)

HDC: Hyperdimensional computing.

By fully exploiting the compact feature of the CBC-LSH and the low-power 2S2R TCAM, a 168× improvement in energy efficiency was obtained compared with the Euclidean distance based on all CMOS methods. In addition, compared with the hyperdimensional computing[41, 45], the significant decrease in the code length (from 10,000 to 16 bits) resulted in a significant improvement in performance (20032.59×). The energy performance of IMS is 18.8 times better than the all-CMOS system (CPU + dedicated Hamming distance calculation circuits), even though the latter also uses the same CBC-LSH method[46].

Finally, we want to stress that Hash, especially the LSH, is one of the most important algorithms in database and search fields. It serves as the similarity-preserving encoder to transform the real-valued vector into binary code. In this study, on the one hand, we demonstrate the natural compatibility of the LSH and memristive hardware. On the other hand, we reconsidered the application of the hash in the similarity-measurement-based data mining field based on the emerging in-memory computing architecture and tried to address the challenges of high complexity and energy consumption faced by current technology. Our results demonstrated that IMS is remarkably promising in clustering and classification tasks. However, unlike hyperdimensional computing, the CBC-LSH only encodes the pre-extracted feature vectors. For non-vector data, such as image and time-series data, the CBC-LSH is incapable. In the future, to realize the end-to-end search, the CBC-LSH still requires additional neural networks (such as convolution and recurrent neural networks) as the automatic feature extractor, or different encoding strategies such as the permutation operation in hyperdimensional computing have to be introduced[22, 45].

Conclusions

In summary, we propose to perform search in memory for similarity-measurement-based data mining. To support more distance functions and reduce the computational complexity of similarity search, we provide a highly efficient memristive hardware encoder to perform the similarity-preserving transformation at first, generating compact binary representations. This allows the complex distance functions to be realized with the binary memristor devices. Subsequently, an ultra-low-power 2S2R TCAM is demonstrated, serving as the high parallel distance computing engine. With the in-memory search prototype, we have demonstrated the *k*-means clustering and *k*-NN classification, achieving software-comparable accuracies with shorter representation ($d \times$ 32-bit real-valued vectors vs. 16 ~ 32-bit codes), lower energy (168 : 1), and lower operation complexity (floating-point multiplications and additions vs. read operations). This work could thus be of

great significance for extending the in-memory computing concept to in-memory search to solve the complex similarity search problem faced by conventional digital computers.

Materials And Methods

Memristor array fabrication and electrical measurements

The memristor array was fabricated on a SiO_2/Si wafer. First, a 100 nm Pt bottom electrode was deposited on the SiO_2/Si substrate by direct current magnetron sputtering. A 100 nm SiO_2 layer was then grown by plasm-enhanced chemical vapour deposition and the via holes were formed by electron beam lithography and inductively coupled plasma etching with a diameter of 250 nm. The 5 nm HfO_2 layer and the 5 nm HfO_x layer were successively deposited by atomic layer deposition and radio frequency magnetron sputtering, respectively. Finally, the 100 nm V top electrode was deposited by direct current magnetron sputtering. All electrical measurements were performed using a Keysight B1500A semiconductor parameter analyser connected with the Cascade M150 probe station and the Keysight MXR404A oscilloscope.

Circuit experiments

The functions of the 2S2R TCAM circuit are demonstrated as followings. The input search signals applied on the SLs (top electrodes) were given by the B1500. The output signals (the ML voltage) were measured with the Keysight MXR404A oscilloscope. The circuit is verified by replacing the memristors with a 4×4 resistor array and replacing the SA with a voltage comparator (LM339N). The search voltage was generated in parallel by using a 2 − 1 MUX (ADG787) array controlled by the Arduino mega2560 microcontroller unit (MCU). More detailed descriptions can be found in fig.s1 in the Supplementary Information.

Simulations

All algorithm simulations were performed with Python 3.7 on Anaconda, and the SPICE simulation for the circuits was carried out on LTspice with the 22 nm technology nodal parameters, where the memristors were replaced by linear resistors. All the simulation parameters used in this work are listed in Table 2.

Table 2
Simulation parameters of this work.

Simulation parameter	Value
HRS, standard deviation (fig.s2)	2.25 MΩ, 0.63 MΩ
LRS, standard deviation (fig.s2)	130.35 kΩ, 18.54 kΩ
ML parasitic capacitance[47]	20 fF

For the encoder circuit, to simplify the circuit structure and constrain the distribution of the result of the dot product to adapt the work range of the voltage comparator, we use the voltage of the word line (V_{WL}) to replace the current as shown in fig.s4 in the Supplementary Information. According to Ohm's law and Kirchhoff's current law, the V_{WL} and its range are given by Eq. (6–7), where V_i is the voltage applied to the i^{th} bit line, G_i is the conductance of the i^{th} device.

$$V_{WL} = \frac{\sum_{i=1}^{d} V_i G_i}{\sum_{i=1}^{d} G_i}$$

6

$$V_{WL} \in [\min(V_i), \max(V_i)]$$

7

Because the *k*-means is demonstrated over the binary codes. The update of the cluster centroids is a little different from the real-valued vectors. For the k-means over real-valued vectors, the centroids in each iteration are obtained by calculating the mean values of each cluster. However, this method requires binary codes. Hence the i^{th} bit of the centroid code C_k of the k^{th} cluster was calculated using the majority function given by Eq. (8), where p_{ji} is the i^{th} bit of the j^{th} code, M_k denotes the sample amount of the k^{th} cluster.

$$C_{ki} = \begin{cases} 1 & \frac{\sum_{j=1}^{M_k} p_{ji}}{M_k} > 0.5 \\ 0 & else \end{cases}$$

8

Take an instance, if a cluster contains four codes which are 1001, 1100, 1101, and 0011. The centroid 1001 will be obtained as follow:

$$\begin{array}{r} 1001 \\ 1100 \\ 1101 \\ +\quad 0011 \\ \hline 1001 \end{array}$$

9

Datasets

1. Iris dataset[48] is widely used in machine learning. It contains four features (length of the sepal and the petal, width of the sepal and the petal) of three iris flowers species: setosa, virginica, and versicolor. Each specie has 50 samples.
2. ISOLET[49] is a high-dimensions dataset with 617 features, containing 7797 data points, which is also widely used in classification and clustering tasks. The dataset contains the voice features of 26 elements of the alphabet from 150 subjects, where each subjects spoke twice. Hence the amount is 7800 (= 150×26×2). But three samples with too large noise were unusable.
3. MNIST handwritten digits dataset[50] is a classical image dataset in the machine learning field. The training set and test set consist of 60000 and 10000 images of the digits from 0 to 9, respectively. In this work, for simplicity, we selected a subset including 1200 (1000 for training and 200 for testing) images randomly to demonstrate the *k*-NN.
4. Document classification: In this study, we used the BBC sports News dataset[51] which contains 737 documents from the BBC Sport website for five classes of news (athletics, cricket, football, rugby, tennis).It is a highly sparse dataset, where each document is described by a 7599 dimensions vector. However, only about 200 element are nonzero.

Distance functions

The Euclidean distance (ED) of two *d*-dimension vectors x and y can be calculated by:

$$ED\left(x,y\right)=\sqrt{\sum_{i=1}^{d}\left(x_i-y_i\right)^2}$$

10

The Normalized Cosine distance (NCD) is:

$$NCD\left(x,y\right)=\frac{1-\frac{xy}{\left|x\right|\left|y\right|}}{2}$$

11

The Normalized Hamming distance (NHD) is given by the Eq. (11), where *d* is the dimension (length) of the two binary vectors *x* and *y*.

$$NHD\left(x,y\right)=\frac{\sum_{i=1}^{d}x_i\oplus y_i}{d}$$

12

Declarations

Author-Contributions

Y. L. and L. Y. conceived and designed the project. L. Y. conceived the in-memory search methodology, carried out the device measurements and system-level simulations, X. H. contributed to the device fabrication and device characterization, S. R., and F. W. contributed to the device fabrication process, H. Z., Y. Y., H. B., J. L., J. C., G. P., and X. L. contributed to the algorithm and circuit design. X. M. coordinated the research project. L. Y., and Y. L. co-wrote the manuscript with contributions from all authors.

Competing interests

The authors declare no competing interests.

Acknowledgements

This work was supported by the National Key Research and Development Plan of MOST of China (Grant No. 2021ZD0201201), the National Natural Science Foundation of China (Grant No. 92064012), Hubei Key Laboratory for Advanced Memories, Hubei Engineering Research Center on Microelectronics, and the Chua Memristor Institute, also in part by Huawei Technologies Corp.

References

1. Dasgupta, S., Stevens, C. F. & Navlakha, S. A neural algorithm for a fundamental Computing Problem. Science (80-.). **358**, 793–796 (2017).
2. Verleysen M, François D. The curse of dimensionality in data mining and time series prediction. In: *International work-conference on artificial neural networks*). Springer (2005).
3. Yao P, *et al.* Fully hardware-implemented memristor convolutional neural network. Nature **577**, 641–646 (2020).

4. Zhang W, *et al.* Neuro-inspired computing chips. Nature Electronics **3**, 371–382 (2020).
5. Sun Z, Ambrosi E, Pedretti G, Bricalli A, Ielmini D. In-memory PageRank accelerator with a cross-point array of resistive memories. IEEE Transactions on Electron Devices **67**, 1466–1470 (2020).
6. Sun Z, Pedretti G, Bricalli A, Ielmini D. One-step regression and classification with cross-point resistive memory arrays. Science advances **6**, eaay2378 (2020).
7. Wang Z, *et al.* Fully memristive neural networks for pattern classification with unsupervised learning. Nature Electronics **1**, 137–145 (2018).
8. Lin P, *et al.* Three-dimensional memristor circuits as complex neural networks. Nature Electronics **3**, 225–232 (2020).
9. Sun L, *et al.* In-sensor reservoir computing for language learning via two-dimensional memristors. Science advances **7**, eabg1455 (2021).
10. Sheridan PM, Cai F, Du C, Ma W, Zhang Z, Lu WD. Sparse coding with memristor networks. Nature nanotechnology **12**, 784–789 (2017).
11. Zidan MA, *et al.* A general memristor-based partial differential equation solver. Nature Electronics **1**, 411–420 (2018).
12. Sun Z, Pedretti G, Ambrosi E, Bricalli A, Wang W, Ielmini D. Solving matrix equations in one step with cross-point resistive arrays. *Proceedings of the National Academy of Sciences* **116**, 4123–4128 (2019).
13. Le Gallo M, *et al.* Mixed-precision in-memory computing. Nature Electronics **1**, 246–253 (2018).
14. Li C, *et al.* Analogue signal and image processing with large memristor crossbars. Nature electronics **1**, 52–59 (2018).
15. Hu M, *et al.* Memristor-based analog computation and neural network classification with a dot product engine. Advanced Materials **30**, 1705914 (2018).
16. Liu Q, *et al.* 33.2 A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing. In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*). IEEE (2020).
17. Cai F, *et al.* A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. Nature Electronics **2**, 290–299 (2019).
18. Zidan MA, Strachan JP, Lu WD. The future of electronics based on memristive systems. Nature electronics **1**, 22–29 (2018).
19. Chen J, Li J, Li Y, Miao X. Multiply accumulate operations in memristor crossbar arrays for analog computing. Journal of Semiconductors **42**, 013104 (2021).
20. Sebastian A, Le Gallo M, Khaddam-Aljameh R, Eleftheriou E. Memory devices and applications for in-memory computing. Nature nanotechnology **15**, 529–544 (2020).
21. Graves CE, *et al.* In-Memory Computing with Memristor Content Addressable Memories for Pattern Matching. Advanced Materials **32**, 2003437 (2020).
22. Karunaratne G, *et al.* Robust high-dimensional memory-augmented neural networks. Nature communications **12**, 1–12 (2021).
23. Fu, Y. *et al.* Reconfigurable Synaptic and Neuronal Functions in a V/VOx/HfWOx/Pt Memristor for Nonpolar Spiking Convolutional Neural Network. Adv. Funct. Mater. **2111996**, 2111996 (2022).
24. Karpov, V. G. & Niraula, D. Log-Normal Statistics in Filamentary RRAM Devices and Related Systems. IEEE Electron Device Lett. **38**, 1240–1243 (2017).
25. Grossi, A. et al. Fundamental variability limits of filament-based RRAM. *Tech. Dig. - Int. Electron Devices Meet. IEDM* 4.7.1–4.7.4 (2017).
26. Kazemi A, *et al.* In-memory nearest neighbor search with fefet multi-bit content-addressable memories. In: *2021 Design, Automation & Test in Europe Conference &* Exhibition *(DATE)*). IEEE (2021).
27. Wang X, *et al.* A 4T2R RRAM Bit Cell for Highly Parallel Ternary Content Addressable Memory. IEEE Transactions on Electron Devices **68**, 4933–4937 (2021).

28. Li H, Jin H, Zheng L, Liao X. ReSQM: Accelerating Database Operations Using ReRAM-Based Content Addressable Memory. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **39**, 4030–4041 (2020).
29. Li C, *et al.* Analog content-addressable memories with memristors. Nature communications **11**, 1–8 (2020).
30. Pedretti G, *et al.* Tree-based machine learning performed in-memory with memristive analog CAM. Nature communications **12**, 1–10 (2021).
31. Ni K, *et al.* Ferroelectric ternary content-addressable memory for one-shot learning. Nature Electronics **2**, 521–529 (2019).
32. Li H, *et al.* One-shot learning with memory-augmented neural networks using a 64-kbit, 118 GOPS/W RRAM-based non-volatile associative memory. In: *2021 Symposium on VLSI Technology*). IEEE (2021).
33. Yang R, *et al.* Ternary content-addressable memory with mos 2 transistors for massively parallel data search. Nature Electronics **2**, 108–114 (2019).
34. Ly D, *et al.* Novel 1T2R1T RRAM-based Ternary Content Addressable Memory for Large Scale Pattern Recognition. In: *2019 IEEE International Electron Devices Meeting (IEDM)*). IEEE (2019).
35. Zhou K, *et al.* High-Density 3-D Stackable Crossbar 2D2R nvTCAM With Low-Power Intelligent Search for Fast Packet Forwarding in 5G Applications. IEEE Journal of Solid-State Circuits **56**, 988–1000 (2020).
36. Fedorov VV, Abusultan M, Khatri SP. An area-efficient ternary cam design using floating gate transistors. In: *2014 IEEE 32nd International Conference on Computer Design (ICCD)*). IEEE (2014).
37. Li J, Montoye RK, Ishii M, Chang L. 1 Mb 0.41 μm² 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing. *IEEE Journal of Solid-State Circuits* **49**, 896–907 (2013).
38. Ly, D. R. B. *et al.* Novel 1T2R1T RRAM-based Ternary Content Addressable Memory for Large Scale Pattern Recognition. Tech. Dig. - Int. Electron Devices Meet. IEDM **2019-Decem**, 2019–2022 (2019)..
39. Lin C-C, *et al.* 7.4 a 256b-wordlength reram-based tcam with 1ns search-time and 14× improvement in wordlength-energyefficiency-density product using 2.5T1R cell. In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*). IEEE (2016).
40. Liu C, *et al.* A high accuracy and robust machine learning network for pattern recognition based on binary RRAM devices. In: *2017 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*). IEEE (2017).
41. Imani M, Kim Y, Worley T, Gupta S, Rosing T. Hdcluster: An accurate clustering using brain-inspired high-dimensional computing. In: *2019 Design, Automation & Test in Europe Conference &* Exhibition *(DATE)*). IEEE (2019).
42. Jeong Y, Lee J, Moon J, Shin JH, Lu WD. K-means data clustering with memristor networks. Nano letters **18**, 4447–4453 (2018).
43. Zhou H, *et al.* Energy-Efficient Memristive Euclidean Distance Engine for Brain-Inspired Competitive Learning. Advanced Intelligent Systems **3**, 2100114 (2021).
44. Zhou, H., Li, Y. & Miao, X. Low-time-complexity document clustering using memristive dot product engine. Sci. China Inf. Sci. **65**, 1–10 (2022).
45. Karunaratne G, Le Gallo M, Cherubini G, Benini L, Rahimi A, Sebastian A. In-memory hyperdimensional computing. Nature Electronics **3**, 327–337 (2020).
46. Mattausch HJ, Yasuda M, Kawabata A, Imafuku W, Koide T. A 381 fs/bit, 51.7 nW/bit nearest hamming-distance search circuit in 65 nm CMOS. In: *2011 Symposium on VLSI Circuits-Digest of Technical Papers*). IEEE (2011).
47. Shen W, *et al.* A Novel Capacitor-based Stateful Logic Operation Scheme for In-memory Computing in 1T1R RRAM Array. In: *2020 4th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*). IEEE (2020).
48. UC Irvine Machine Learning Repository; https://archive-beta.ics.uci.edu/ml/datasets/iris.
49. UC Irvine Machine Learning Repository; https://archive-beta.ics.uci.edu/ml/datasets/isolet.
50. THE MNIST DATABASE of handwritten digits; http://yann.lecun.com/exdb/mnist/.

51. Machine Learning + Network Analysis Resources; http://mlg.ucd.ie/datasets/bbc.html.

Figures

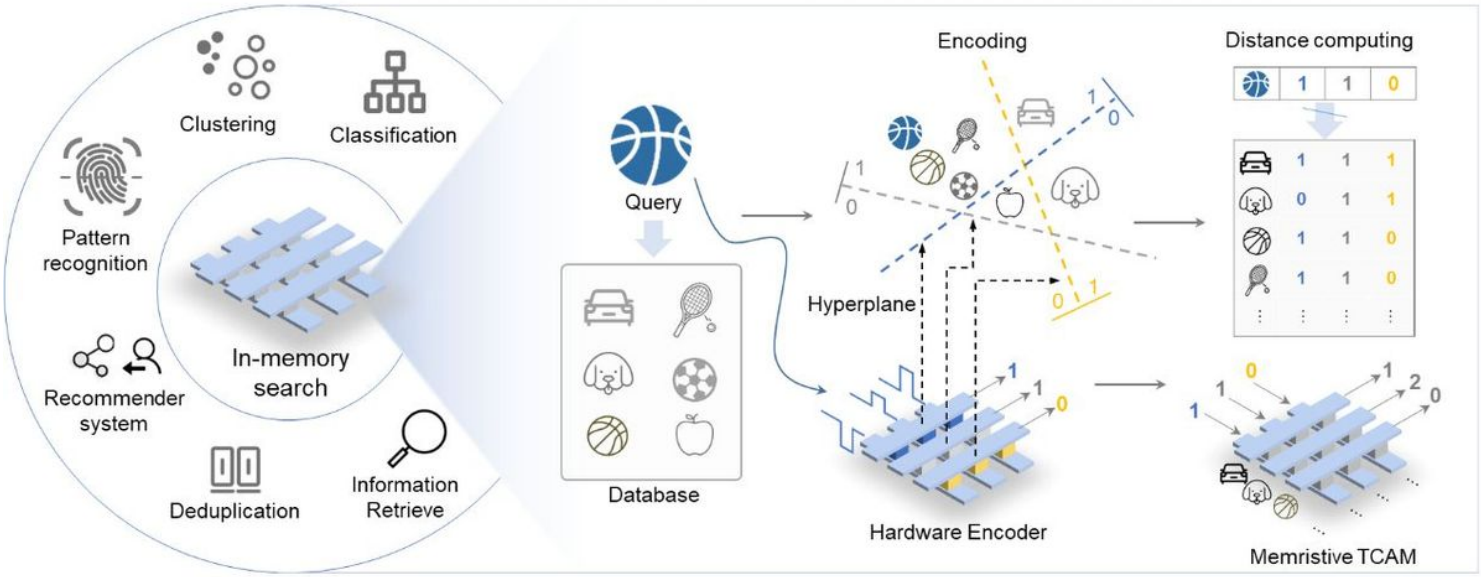


Figure 1

Proposed memristor-based in-memory search prototype. Similarity search, finding a similar item in the database, is a fundamental problem in many fields such as data mining including the classification, clustering etc. It is a data-intensive problem and requires huge computing source in general. To address this issue, we propose the in-memory search methodology to perform the complex encoding and distance computing on the memristive hardware *in situ*. The hardware encoder performs the local sensitive hash algorithm to map the real-valued vectors into binary codes. And the Hamming distance of the binary codes is calculated on the memristive content addressable memory to find the nearest neighbour item.

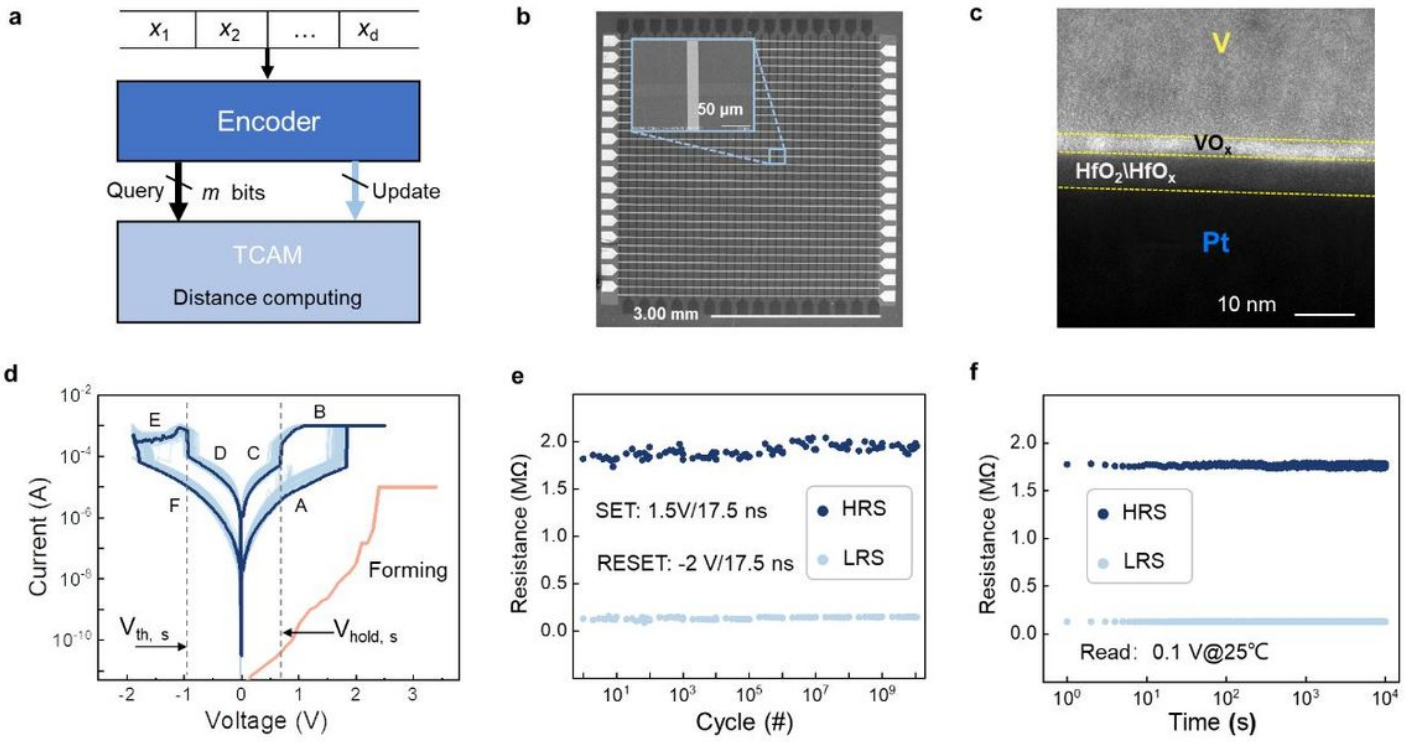


Figure 2

Schematic of the IMS system built with the V/HfO_x/HfO_2/Pt 1S1R devices. (**a**) The in-memory search system consists of a 1S1R-based encoder and a ternary content addressable memory (TCAM). (**b**) Scanning electron microscopy image of the fabricated 32×32 array. (**c**) Transmission electron microscopy image of a V/HfO_x/HfO_2/Pt device. (**d**) I-V curves of the forming process and subsequent 100 switching cycles (**e**) Endurance test of the 1S1R device under the SET (1.5 V / 17.5 ns) and RESET (-2 V / 17.5 ns). (**f**) Retention test results indicate a stable state maintenance over 10^4 s with trivial fluctuations.

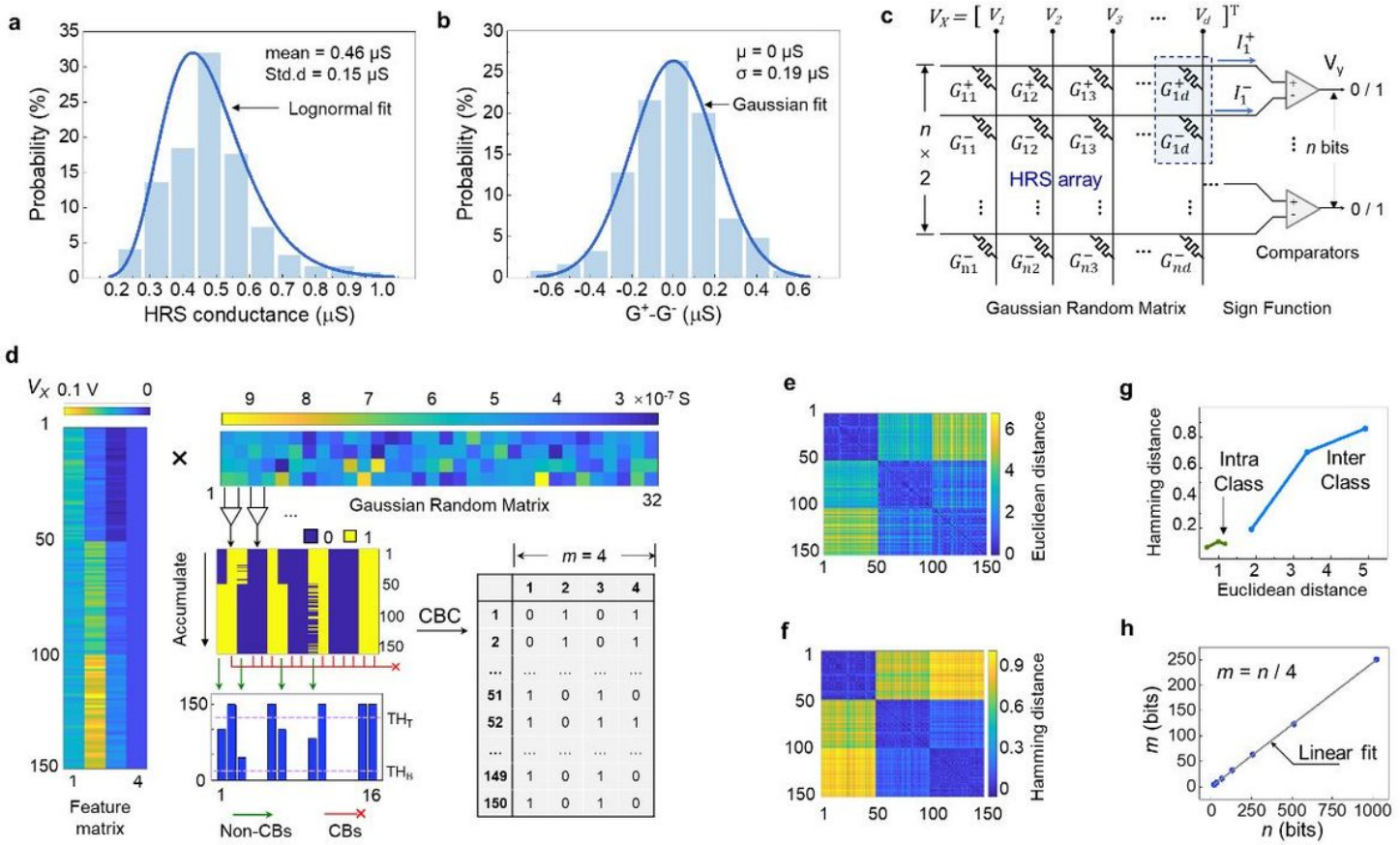
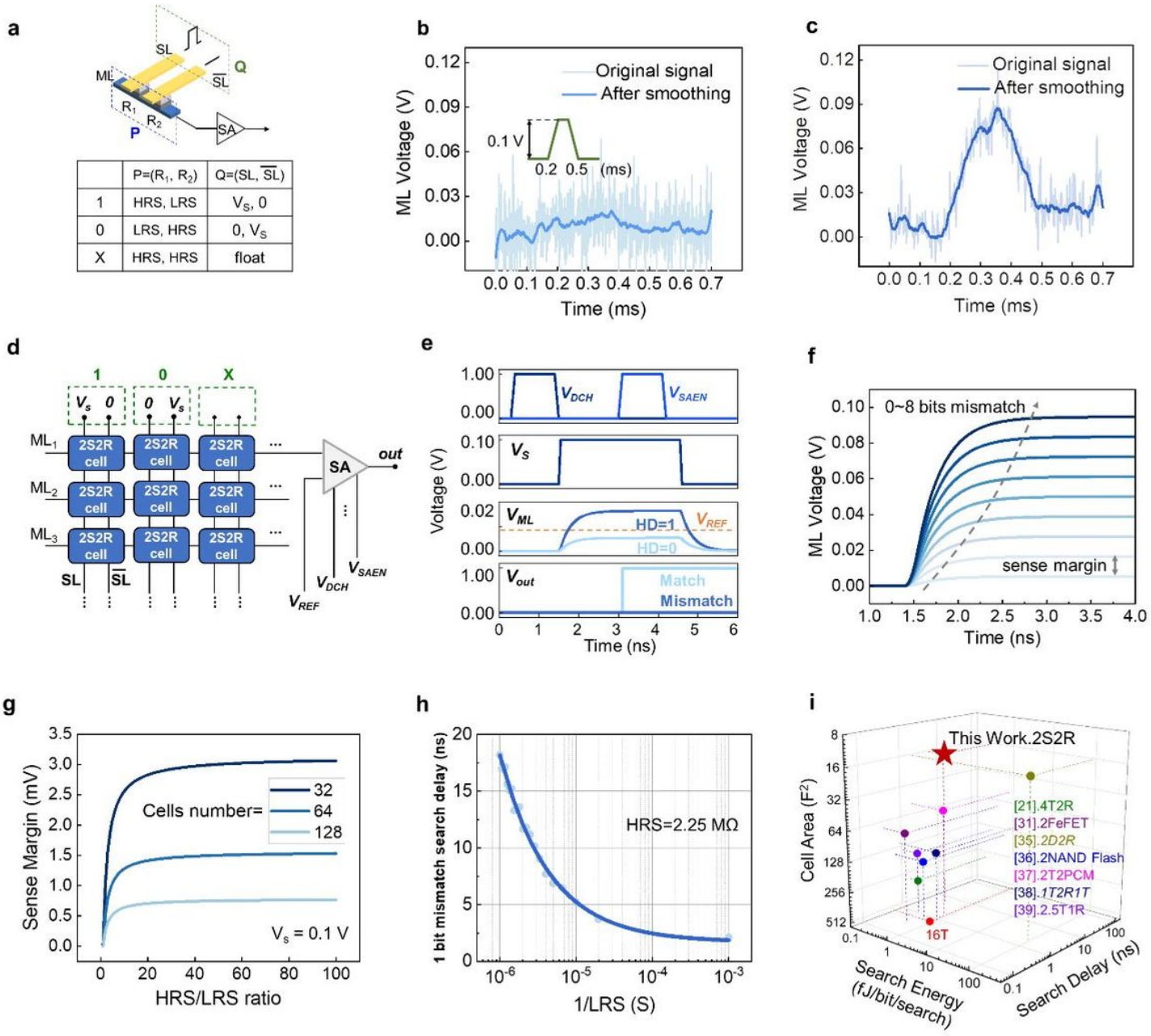


Figure 3

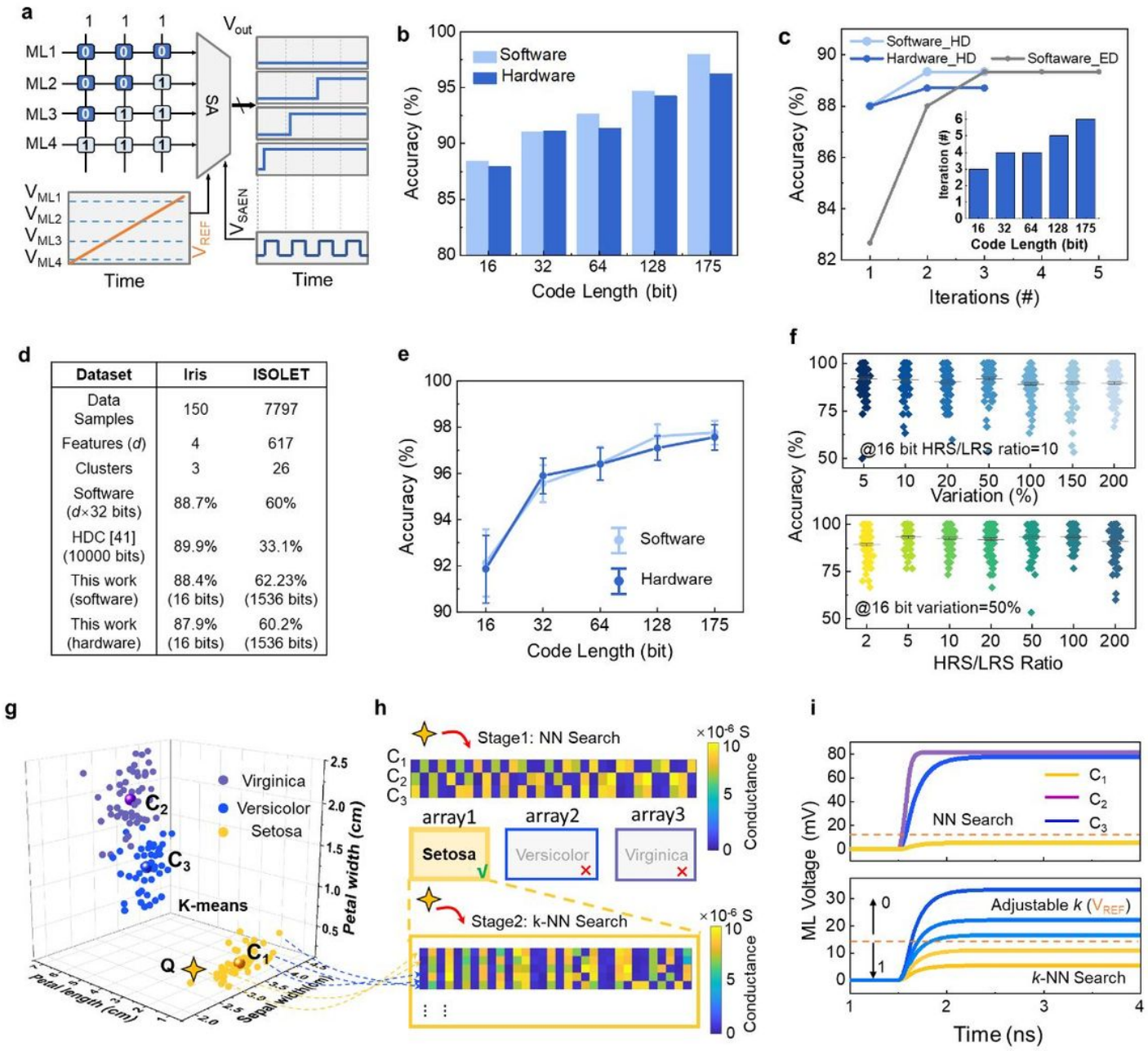
Hardware Encoder. (**a**) Distribution of the device conductance in the HRS. It follows a lognormal distribution with a mean value of 0.46 μS and a standard deviation (std.d) of 0.15 μS. (**b**) Conductance distribution of the differential pairs. It follows a Gaussian- distribution with a mean value (μ) of about 0 μS and a standard deviation (σ) of 0.19 μS. (**c**) Hardware implementation of the encoder. The memristor array written in the HRS serves as the Gaussian random matrix, where a pair of devices map a value, and the comparators realize the difference and sign functions. (**d**) Simulation results of the encoder with the iris dataset based on the experimental data. The 128 devices were organized into a 4×32 array as a 4×16 Gaussian random matrix. Furthermore, the feature vectors were mapped to the voltages with 4-bit digital-to-analogue converters (DACs). The common bits were found from the 16-bit raw codes by bitwise accumulation. After compressing the common bits, the dataset was encoded into a 4-bit codes group. (**e, f**) Similarity matrices based on the Euclidean distance and Hamming distance. (**g**) Relation between the Euclidean distances of original data and the Hamming distances of the codes after encoding, including the intra-class distances and inter-class distances. (**h**) Compression ratio of the iris dataset.



	P=(R1, R2)	Q=(SL, $\overline{SL}$)
1	HRS, LRS	VS, 0
0	LRS, HRS	0, VS
X	HRS, HRS	float

Figure 4

Search with the two-selector-two-resistor (2S2R) TCAM. (**a**) Schematic of the 2S2R TCAM and state definition. Experimental test results of the TCAM cell in (**b**) the match and (**c**) mismatch cases. (**d**) The structure diagram of the 2S2R TCAM array and the sense amplifier. (**e**) Simulation result of search operation of the 8-bit 2S2R TCAM in 1-bit mismatch and full match cases. (**f**) Simulation result of match line (ML) voltages in 0-8 bits mismatch case. (**g**) Relationship between the sense margin and HRS/LRS ratio in different array sizes. (**h**) Relationship between the LRS and the search delay. (**i**) Comparison of TCAMs based on different technologies. By exploiting the compact two-terminal passive 1S1R device, our 2S2R TCAM shows significant improvements in cell area (16.3 F^2) and ultra-low search energy (0.25 fJ / bit / search) with a search delay of 6 ns.



Dataset	Iris	ISOLET
Data Samples	150	7797
Features (*d*)	4	617
Clusters	3	26
Software (*d*×32 bits)	88.7%	60%
HDC [41] (10000 bits)	89.9%	33.1%
This work (software)	88.4% (16 bits)	62.23% (1536 bits)
This work (hardware)	87.9% (16 bits)	60.2% (1536 bits)

Figure 5

In-memory similarity search for *k*-means clustering and *k*-NN classification with iris datasets. (**a**) Hardware implementation of top-*k* search by using a scanning voltage as the V_{REF} of the sense amplifier (SA). (**b**) Simulation results of the *k*-means clustering with different code lengths. (**c**) Comparison of the iteration process of *k*-means. The insert graph shows the relation between the iterations and the code length. (ED: Euclidean distance) (**d**) Comparison of the *k*-means accuracy of different technologies (HDC: hyperdimensional computing). (**e**) Simulation results of the *k*-NN classification accuracy for different code lengths. (**f**) Reliability analysis of *k*-NN. The top panel and bottom panel show the influence on classification accuracy from the device resistance variations and HRS / LRS ratio. (**g - h**) Two-stage search scheme. The centroids of three classes (C_1, C_2, and C3) are obtained by *k*-means and stored in the first stage of TCAM for classification. Additionally, the data belonging to different classes are stored in different arrays in the second stage for *k*-NN searches in the class. (**i**) SPICE simulation result of the two-stage search.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupportinginformationInmemorySearchwithMemristorsforHighlyEfficientSimilarityMeasurementBasedDataMining.pdf