Efficient Biosignal Processing with Brain-inspired High-dimensional Computing:

A Universal ExG Classifier

Abbas Rahimi, Pentti Kanerva, Luca Benini, Jan M. Rabaey

ETH Zurich and UC Berkeley
Brain-inspired High-dimensional Computing

[P. Kanerva, An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors, Cogn Comput’09]

- Emulation of cognition by computing with high-dimensional vectors as opposed to computing with numbers
- Information distributed in high-dimensional space
- Supports full algebra

Superb properties:
- General and scalable model of computing
- Well-defined set of arithmetic operations
- Fast and one-shot learning (no need of back-prop)
- Memory-centric with embarrassingly parallel operations
- Extremely robust against most failure mechanisms and noise
- Energy efficient
Brain-inspired High-dimensional Computing

[P. Kanerva, An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors, Cogn Comput’09]

• Emulation of cognition by computing with high-dimensional vectors as opposed to computing with numbers
• Information distributed in high-dimensional space
• Supports full algebra

Superb properties:
• General and scalable model of computing
• Well-defined set of arithmetic operations
• Fast and one-shot learning (no need of back-prop)
• Memory-centric with embarrassingly parallel operations
• Extremely robust against most failure mechanisms and noise
• Energy efficient
What Are HD Vectors?

It is all about data representation

1\text{st} \ 2\text{nd} \ 3\text{rd} \ 4\text{th} \ 10000\text{th}

\[-1 \ +1 \ -1 \ -1 \ \ldots \ \ldots \ \ldots \ \ldots \ +1\]
What Are HD Vectors?

It is all about data representation

$[\begin{array}{cccccc}
-1 & +1 & -1 & -1 & \ldots & +1 \\
\end{array}]$

High-dimensional
What Are HD Vectors?

It is all about data representation

\[ [-1 \quad +1 \quad -1 \quad -1 \quad \ldots \ldots \ldots \ldots \quad +1] \]

1\textsuperscript{st} 2\textsuperscript{nd} 3\textsuperscript{rd} 4\textsuperscript{th} 10000\textsuperscript{th}

High-dimensional Holographic Distributed
What Are HD Vectors?

It is all about data representation

$$[-1 \; +1 \; -1 \; -1 \; \ldots \ldots \ldots \ldots \; +1]$$

1st 2nd 3rd 4th 10000th

High-dimensional Holographic Distributed Pseudorandom with i.i.d. components
What Are HD Vectors?

It is all about data representation

\[
[\begin{array}{cccc}
-1 & +1 & -1 & -1 \\
\end{array} \ldots \begin{array}{c}
+1
\end{array}]
\]

High-dimensional \Uparrow \quad Holographic \Uparrow \quad Distributed \Uparrow \quad Pseudorandom with i.i.d. components

Combine and Compare
What Are HD Vectors?

It is all about data representation

\[ \begin{bmatrix} -1 & +1 & -1 & -1 & \ldots & +1 \end{bmatrix} \]

1\textsuperscript{st} 2\textsuperscript{nd} 3\textsuperscript{rd} 4\textsuperscript{th} 10,000\textsuperscript{th}

High-dimensional Holographic Distributed Pseudorandom with i.i.d. components

Combine and Compare

Approximate computation with fixed-size long random patterns that provides a novel look at data representations, associated operations, circuits, and architectures.
Mapping to HD Vectors

• Each letter (symbol) is represented by an HD vector chosen at random with 10,000–d:

\[
\begin{align*}
A &= [-1, +1, -1, -1, -1, +1, -1, -1, -1, \ldots] \\
B &= [+1, -1, +1, +1, +1, -1, +1, -1, -1, \ldots] \\
C &= [-1, -1, -1, +1, +1, -1, +1, -1, -1, \ldots] \\
D &= [-1, -1, -1, +1, +1, -1, +1, -1, -1, \ldots] \\
&\vdots \\
Z &= [-1, -1, +1, -1, +1, +1, +1, -1, -1, \ldots]
\end{align*}
\]

• Every letter HD vector is dissimilar to others: \(\langle A, B \rangle = 0\)
• This assignment is fixed throughout computation
HD Arithmetic

- Componentwise addition (+) is good for representing sets, since sum vector is similar to its constituent vectors:
  \[ \langle A + B, A \rangle = 0.5 \]

- Componentwise multiplication (\(\ast\)) is good for binding, since product vector is dissimilar to its constituent vectors:
  \[ \langle A \ast B, A \rangle = 0 \]

- Permutation (\(\rho\)) makes a dissimilar vector by rotating, it good for representing sequences:
  \[ \langle A, \rho A \rangle = 0 \]

- \(\ast\) and \(\rho\) are invertible and preserve distance
Example: Computing Language Profile

How to encode “Ich bin”? 
Example: Computing Language Profile

How to encode “Ich bin”?  

\[ I = +1 -1 -1 +1 -1 -1 \ldots +1 +1 -1 -1 \]  

\[ C = +1 -1 +1 +1 +1 +1 \ldots +1 -1 +1 -1 \]  

\[ H = +1 +1 +1 -1 -1 +1 \ldots +1 -1 +1 +1 \]
Example: Computing Language Profile

How to encode “Ich bin”?

Trigram encoding: “Ich” = $\rho \rho I \ast \rho C \ast H$

\[
\begin{align*}
I &= +1 -1 -1 +1 -1 -1 \ldots +1 +1 -1 -1 +1 -1 \\
C &= +1 -1 +1 +1 +1 +1 \ldots +1 -1 +1 -1 +1 \\
H &= +1 +1 +1 -1 -1 +1 \ldots +1 -1 +1 +1
\end{align*}
\]

“Ich” = +1 +1 -1 +1 .... .... +1 +1 -1 -1
Example: Computing Language Profile

How to encode “Ich bin”?

Trigram encoding: “Ich” = $ppI \ast pC \ast H$

\[ I = +1 -1 -1 +1 -1 -1 \ldots +1 +1 -1 -1 +1 -1 \]

\[ C = +1 -1 +1 +1 +1 +1 \ldots +1 -1 +1 -1 +1 \]

\[ H = +1 +1 +1 -1 -1 +1 \ldots +1 -1 +1 +1 \]

“Ich” = +1 +1 -1 +1 \ldots \ldots +1 +1 -1 -1

Adding trigrams: “Ich bin” =
Example: Computing Language Profile

How to encode “Ich bin”?

Trigram encoding: “Ich” = $\rho \rho I \ast \rho C \ast H$

$I = +1 -1 -1 +1 -1 -1 \ldots +1 +1 -1 -1 +1 -1$

$C = +1 -1 +1 +1 +1 +1 \ldots +1 -1 +1 -1 +1$

$H = +1 +1 +1 -1 -1 +1 \ldots +1 -1 +1 +1$

“Ich” = +1 +1 -1 +1 .... .... +1 +1 -1 -1

Adding trigrams: “Ich bin” =

“Ich” = +1 +1 -1 +1 -1 +1 .... ....

“ch” = -1 -1 +1 +1 -1 +1 .... ....

“hb” = -1 -1 +1 +1 +1 -1 .... ....

“bi” = +1 -1 +1 -1 -1 -1 .... ....

“bin” = -1 +1 +1 -1 -1 +1 .... ....
Example: Computing Language Profile

How to encode “Ich bin”?

Trigram encoding: “Ich” = $p p I \ast p C \ast H$

\[
\begin{align*}
I &= +1 -1 -1 +1 -1 -1 \ldots +1 +1 -1 -1 +1 -1 \\
C &= +1 -1 +1 +1 +1 \ldots +1 -1 +1 -1 +1 \\
H &= +1 +1 +1 -1 -1 +1 \ldots +1 -1 +1 +1
\end{align*}
\]

Adding trigrams: “Ich bin” =

\[
\begin{align*}
"Ich" &= +1 +1 -1 +1 \ldots \ldots +1 +1 -1 -1 -1 \\
"ch" &= -1 -1 +1 +1 -1 +1 \ldots \\
"h b" &= -1 -1 +1 +1 +1 -1 \ldots \\
"bi" &= +1 -1 +1 -1 -1 \ldots \\
"bin" &= -1 +1 +1 -1 -1 +1 \ldots \\
\end{align*}
\]

= -1 -1 +1 +1 -1 +1 \ldots

“Ich bin”
EU Language Recognition

**Identical hardware** for both learning and inference

Train with 100 KB of text from 21 EU languages

**Train** text: “der emissionserlöses soll fuer den weiteren ausbau des qualitativ ...”

![Diagram](image)

- **Item Memory**
  - Letter HD vector 10,000-d

- **Encoding**: $(\ast, +, \rho)$ operations

- **Language** HD vector 10,000-d
EU Language Recognition

**Identical hardware** for both learning and inference

Train with 100 KB of text from 21 EU languages

Train text: “der emissionserloes soll fuer den weiteren ausbau des qualitativ ...”

- Item Memory
  - Letter HD vector
  - 10,000-d

- Encoding: \((*,+,\rho)\) operations

- Language HD vector
  - 10,000-d

- German
  - \(-1\ -1\ +1\ +1\ -1\ ...

- Associative Memory
EU Language Recognition

Identical hardware for both learning and inference

Train with 100 KB of text from 21 EU languages

Train text: “der emissionserloes soll fuer den weiteren ausbau des qualitativ ...”

Item Memory

Letter HD vector

10,000-d

Encoding:

(\ast,+,\rho) operations

Language HD vector

10,000-d

German

\begin{align*}
-1 & -1 +1 +1 -1 & \\
-1 & +1 +1 -1 +1 & \\
\end{align*}

Associative Memory

21 \times 10,000 learned language patterns
EU Language Recognition

**Identical hardware** for both learning and inference

Train with 100 KB of text from 21 EU languages

Test with 1,000 sentences for each language

**Train** text: “der emissionserloes soll fuer den weiteren ausbau des qualitativ ...”

**Test** sentence: “daher stimme ich gegen anderungsantrag welcher”

**Item Memory**

- Letter HD vector 10,000-d
- Encoding: \((\ast, +, \rho)\) operations
- Language HD vector 10,000-d

**German**

- 21 \times 10,000 **learned** language patterns

**Associative Memory**

- -1 -1 +1 +1 -1 ....
- -1 +1 +1 -1 +1 ....

**Item Memory**

- Letter HD vector 10,000-d
- Encoding: \((\ast, +, \rho)\) operations
- Query HD vector 10,000-d

**Associative Memory**

- -1 -1 +1 +1 -1 ....
- -1 +1 +1 -1 +1 ....

Search on **learned** language HD vectors

Identified language
Generic HD Processing Unit

HD Space

HD vectors → $d$

HD vectors → $d$

HD vectors → $d$
Generic HD Processing Unit

Part of preprocessing can be eliminated
Generic HD Processing Unit

Maps input vectors into $d$-dimensional ($d \approx 10,000$) pseudo-orthogonal random vectors.

Nanodevice opportunity to exploit process randomness and utilize variability.

Part of preprocessing can be eliminated.

Seed Generation

LD Space

Preprocessing/Transformations

Inputs

$HD$ Space

$Seed$ Generation

$HD$ vectors

$Preprocessing/Transformations$

$m$

$HD$ vectors

Part of preprocessing can be eliminated

Maps input vectors into $d$-dimensional ($d \approx 10,000$) pseudo-orthogonal random vectors.

Nanodevice opportunity to exploit process randomness and utilize variability.

Seed Generation

$HD$ vectors

$Preprocessing/Transformations$

$m$

$HD$ vectors

Part of preprocessing can be eliminated

Maps input vectors into $d$-dimensional ($d \approx 10,000$) pseudo-orthogonal random vectors.

Nanodevice opportunity to exploit process randomness and utilize variability.
Generic HD Processing Unit

Part of preprocessing can be eliminated

Maps input vectors into d-dimensional \((d\approx10,000)\) pseudo-orthogonal random vectors

Nanodevice opportunity to **exploit process randomness and utilize variability**

Encodes all input information into single HD vector using simple local operators \(\ast,+,\rho\)
Generic HD Processing Unit

- **Preprocessing/Transformations**: Maps input vectors into \( d \)-dimensional \((d \approx 10,000)\) pseudo-orthogonal random vectors.
- **Seed Generation**: Nanodevice opportunity to **exploit process randomness and utilize variability**.
- **HD Encoder**: Encodes all input information into single HD vector using simple local operators \((\ast, +, \rho)\).
- **Associative Memory**: Finds closest match in trained data. Can be continuously updated.

Part of preprocessing can be eliminated.
HD Processing for Gesture Recognition

Preprocessing

60 Hz Notch → Exp. filter → Quani. 21

CiM

iM

n-1

R[t]

\( \rho(R[t-1]) \)

\( \rho^{n-1}(R[t-n+1]) \)

EMG (Label[t]) += ngram[t]

Match

Out
HD Processing for Gesture Recognition

Mapping in HD space

EMG (Label[t]) += ngram[t]

Match

Out
HD Processing for Gesture Recognition

![Diagram of HD Processing for Gesture Recognition](image)

- **CH1**
  - Value: 60 Hz Notch
  - Name: 'CH1'

- **CH2**
  - Value: 60 Hz Notch
  - Name: 'CH2'

**Temporal encoder**
- \( R[t] \)
- \( \rho(R[t-1]) \)
- \( \rho^{n-1}(R[t-n+1]) \)

**HD encoder**
- \( n_{gram}[t] \)
- \( EMG (Label[t]) + n_{gram}[t] \)

**Match**

**Out**
HD Processing for Gesture Recognition

\[ \text{EMG (Label[t])} = \text{ngram[t]} \]

Temporal encoder

\[ R[t], \rho(R[t-1]), \ldots, \rho^{n-1}(R[t-n+1]) \]

\[ \text{ngram[t]} \]

\[ \text{EMG (Label[t])} + \text{ngram[t]} \]

\[ \text{Match} \]

\[ \text{Out} \]

CH1

\[ \text{value} \quad \text{Exp. filter} \quad \text{Quani. 21} \quad \text{CiM} \quad \text{iM} \]

CH2

\[ \text{value} \quad \text{Exp. filter} \quad \text{Quani. 21} \quad \text{CiM} \quad \text{iM} \]
Many variants of same ...

**Applications**

<table>
<thead>
<tr>
<th></th>
<th>#I</th>
<th>#C</th>
<th>HD</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification [ISLPED’16]</td>
<td>1</td>
<td>21</td>
<td>96.7%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Text categorization [DATE’16]</td>
<td>1</td>
<td>8</td>
<td>94.2%</td>
<td>86.4%</td>
</tr>
<tr>
<td>Speech recognition [ICRC’17]</td>
<td>1</td>
<td>26</td>
<td>95.3%</td>
<td>93.6%</td>
</tr>
<tr>
<td>EMG gesture recognition [ICRC’17]</td>
<td>4</td>
<td>5</td>
<td>97.8%</td>
<td>89.7%</td>
</tr>
<tr>
<td>Flexible EMG [ISCAS’18]</td>
<td>64</td>
<td>5</td>
<td>96.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>EEG brain-machine interface [BICT’17]</td>
<td>64</td>
<td>2</td>
<td>74.5%</td>
<td>69.5%</td>
</tr>
</tbody>
</table>

**Diagram:**

- **Plant**
  - EMG Channels
  - Spatial Encoding
  - Temporal Encoding
  - Query GV
  - Associative Memory (AM)
  - Cosine

- **Controller**
  - Actuation: change N
  - Measurement: cosine similarity

- **Encoder**
  - 1st Electrode
  - Preprocessing
  - BPF
  - iM

- **Temporal Encoder**
  - p(R[t−1])
  - p^k−1(R[t−N+1])
  - EMG (Label[t]) += ngram[t]

- **Spatial Encoder**
  - Temporal Encoder
  - ngram[t]
  - EMG (Label[t]) += ngram[t]

- **Plant**
  - 64th Electrode
  - Preprocessing
  - BPF
  - iM

- **Controller**
  - Actuation: change N
  - Measurement: cosine similarity

- **Encoder**
  - 1st Electrode
  - Preprocessing
  - BPF
  - iM

- **Temporal Encoder**
  - p(R[t−1])
  - p^k−1(R[t−N+1])
  - EMG (Label[t]) += ngram[t]

- **Spatial Encoder**
  - Temporal Encoder
  - ngram[t]
  - EMG (Label[t]) += ngram[t]
Many variants of same ...

Applications | #I | #C | HD | Baseline
---|---|---|---|---
Language identification [ISLPED'16] | 1 | 21 | 96.7% | 97.9%
Text categorization [DATE'16] | 1 | 8 | 94.2% | 86.4%
Speech recognition [ICRC'17] | 1 | 26 | 95.3% | 93.6%
EMG gesture recognition [ICRC'16] | 4 | 5 | 97.8% | 89.7%
Flexible EMG [ISCAS'18] | 64 | 5 | 96.6% | 88.9%
EEG brain-machine interface [BICT'17] | 64 | 2 | 74.5% | 69.5%
Applications

<table>
<thead>
<tr>
<th></th>
<th>#I</th>
<th>#C</th>
<th>HD</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification</td>
<td>1</td>
<td>21</td>
<td>96.7%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Text categorization</td>
<td>1</td>
<td>8</td>
<td>94.2%</td>
<td>86.4%</td>
</tr>
<tr>
<td>Speech recognition</td>
<td>1</td>
<td>26</td>
<td>95.3%</td>
<td>93.6%</td>
</tr>
<tr>
<td>EMG gesture recognition</td>
<td>4</td>
<td>5</td>
<td>97.8%</td>
<td>89.7%</td>
</tr>
<tr>
<td>Flexible EMG</td>
<td>64</td>
<td>5</td>
<td>96.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>EEG brain-machine interface</td>
<td>64</td>
<td>2</td>
<td>74.5%</td>
<td>69.5%</td>
</tr>
</tbody>
</table>
Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>#I</th>
<th>#C</th>
<th>HD</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification [ISLPED’16]</td>
<td>1</td>
<td>21</td>
<td>96.7%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Text categorization [DATE’16]</td>
<td>1</td>
<td>8</td>
<td>94.2%</td>
<td>86.4%</td>
</tr>
<tr>
<td>Speech recognition [ICRC’17]</td>
<td>1</td>
<td>26</td>
<td>95.3%</td>
<td>93.6%</td>
</tr>
<tr>
<td>EMG gesture recognition [ICRC’16]</td>
<td>4</td>
<td>5</td>
<td>97.8%</td>
<td>89.7%</td>
</tr>
<tr>
<td>Flexible EMG [ISCAS’18]</td>
<td>64</td>
<td>5</td>
<td>96.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>EEG brain-machine interface [BICT’17]</td>
<td>64</td>
<td>2</td>
<td>74.5%</td>
<td>69.5%</td>
</tr>
</tbody>
</table>
Many variants of same ...

<table>
<thead>
<tr>
<th>Applications</th>
<th>#I</th>
<th>#C</th>
<th>HD</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language identification [ISLPED’16]</td>
<td>1</td>
<td>21</td>
<td>96.7%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Text categorization [DATE’16]</td>
<td>1</td>
<td>8</td>
<td>94.2%</td>
<td>86.4%</td>
</tr>
<tr>
<td>Speech recognition [ICRC’17]</td>
<td>1</td>
<td>26</td>
<td>95.3%</td>
<td>93.6%</td>
</tr>
<tr>
<td>EMG gesture recognition [ICRC’16]</td>
<td>4</td>
<td>5</td>
<td>97.8%</td>
<td>89.7%</td>
</tr>
<tr>
<td>Flexible EMG [ISCAS’18]</td>
<td>64</td>
<td>5</td>
<td>96.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>EEG brain-machine interface [BICT’17]</td>
<td>64</td>
<td>2</td>
<td>74.5%</td>
<td>69.5%</td>
</tr>
</tbody>
</table>

- 2× lower energy than SVM
- **Embedded Accelerator**
  (28nm, 1.5mm², 2mW)  
  [DAC’18]
- 256 channels: 10 ms real-time constraints
- 10× lower energy than ARM Cortex-M4
HD Learns $3 \times$ Faster

EMG (8 gel-based electrodes)

SVM needs $3.2 \times$ more trials [ICRC’16]
HD Learns $3 \times$ Faster

EMG (8 gel-based electrodes)

SVM needs $3.2 \times$ more trials [ICRC’16]

EEG (64 electrodes)

Gaussian needs $3 \times$ more trials and preprocessing [BICT’17]
HD Learns $3 \times$ Faster

EMG (8 gel-based electrodes)
SVM needs $3.2 \times$ more trials [ICRC’16]

EEG (64 electrodes)
Gaussian needs $3 \times$ more trials and preprocessing [BICT’17]

Flexible high-density electrode array [ISCAS’18]

Train/Test 30 mins apart

One-shot: 89.2%
HD Learns $3 \times$ Faster

EMG (8 gel-based electrodes)

SVM needs $3.2 \times$ more trials [ICRC’16]

Flexible high-density electrode array [ISCAS’18]

EEG (64 electrodes)

Gaussian needs $3 \times$ more trials and preprocessing [BICT’17]

Train/Test 30 mins apart

One-shot: 89.2%

Train/Test 1 day apart

With repositioning: 82.0%

SVM: 51%
The True Opportunity for HD
3D Integration and Nanoscale Devices

Random HD vectors: ~ 10k bits

Letters/ signals/ features

Applications

Utilizing RRAM stochasticity

Pulse Amplitude (V)
Pulse Width (ns)
0.7 0.8 0.9 1.0 1.1
0% 25% 50% 75% 100%

011101010101...011101010100...
101100100101...101010011110...

# of items
The True Opportunity for HD 3D Integration and Nanoscale Devices

Letters/ signals/ features

Applications

Representation
Utilizing RRAM stochasticity

Random HD vectors: ~ 10k bits

In-memory Processing
3D vertical RRAM for MAP operations [IEDM’16]
The True Opportunity for HD 3D Integration and Nanoscale Devices

Letters/ signals/ features

Applications

Random HD vectors: ~ 10k bits

Representation

Utilizing RRAM stochasticity

In-memory Processing

3D vertical RRAM for MAP operations [IEDM’16]

Learning/ Inference

Analog resistive associative memory [HPCA’17]
HD is Extremely Robust Against Errors
HD is Extremely Robust Against Errors
HD is Extremely Robust Against Errors
HD is Extremely Robust Against Errors

- **HD is Extremely Robust**
  - Against Errors

**Graphs:**
- **Left Graph:**
  - Title: Probability of failure for each memory cell
  - Y-axis: Accuracy (%)
  - X-axis: Probability of failure
  - Data points:
    - KNN
    - HD (10K)
  - Observations:
    - HD (10K) is significantly more robust than KNN
    - HD (10K) maintains higher accuracy at higher error probabilities

- **Right Graph:**
  - Title: Proportion of Hard Errors (%)
  - Y-axis: Accuracy (%)
  - X-axis: Proportion of Hard Errors (%)
  - Data points:
    - KNN
    - HD (2K)
    - HD (10K)
  - Observations:
    - HD (10K) significantly outperforms KNN and HD (2K)
    - HD (10K) retains high accuracy across a wide range of error proportions

**Comparisons:**
- **9x** improvement for HD (10K) compared to KNN
- **83x** improvement for HD (10K) compared to HD (2K)
HD is Extremely Robust Against Errors

Robustness in low SNR:
- Seed hypervectors with **i.i.d. components**
- MAP operations are nearly **i.i.d.-preserving**
- **Holographic**: a failure in a component is not “contagious”
- HD algorithm is **data-driven** with (almost) no control flow conditions
High Order Bits

• Simple HD architectural templates to encode analog input signals for various biosignal applications
• Fully scalable
• Identical hardware for leaning and inference
• Fast learning under low SNR conditions
  • Enabling online and continuous learning!
Relevant publications


• A. Rahimi, P. Kanerva, J. del R. Millan, J. M. Rabaey, “Hyperdimensional Computing for Noninvasive Brain-Computer Interfaces: Blind and One-Shot Classification of EEG Error-Related Potentials,” In 10th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT), March 2017. [Best Paper] [PDF] [PPTX] [Artifact]


• A. Rahimi, S. Benatti, P. Kanerva, L. Benini, and J. M. Rabaey, “Hyperdimensional Biosignal Processing: A Case Study for EMG-based Hand Gesture Recognition,” In IEEE International Conference on Rebooting Computing (ICRC), 2016. [PDF] [PPTX] [Artifact] [Video]

• A. Rahimi, P. Kanerva, and J. M. Rabaey, “A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing,” In ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), 2016. [PDF] [PPTX] [Artifact]


