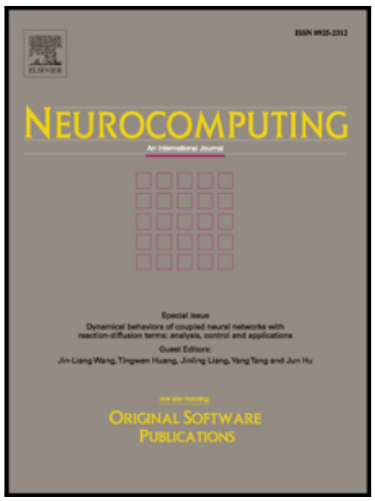


Extreme Learning Machine based Mutual Information Estimation with Application to Time-series Change-points Detection

Beom-Seok Oh, Lei Sun, Chung Soo Ahn, Yong Kiang Yeo,
Yan Yang, Nan Liu, Zhiping Lin

PII: S0925-2312(17)30218-7
DOI: 10.1016/j.neucom.2015.11.138
Reference: NEUCOM 18015

To appear in: *Neurocomputing*

Received date: 22 September 2015
Revised date: 4 November 2015
Accepted date: 7 November 2015

Please cite this article as: Beom-Seok Oh, Lei Sun, Chung Soo Ahn, Yong Kiang Yeo, Yan Yang, Nan Liu, Zhiping Lin, Extreme Learning Machine based Mutual Information Estimation with Application to Time-series Change-points Detection, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2015.11.138

Extreme Learning Machine based Mutual Information Estimation with Application to Time-series Change-points Detection

Beom-Seok Oh[a], Lei Sun[b,a], Chung Soo Ahn[a], Yong Kiang Yeo[a], Yan Yang[c], Nan Liu[d], Zhiping Lin[a,*]

[a] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*
[b] *School of Information and Electronics, Beijing Institute of Technology, China*
[c] *Center of Intelligent Acoustics and Immersive Communications, School of Marine Technology, Northwestern Polytechnical University, China*
[d] *Department of Emergency Medicine, Singapore General Hospital, Singapore*

Abstract

In this paper, we propose an efficient parameter tuning-free Mutual Information (MI) estimator in a form of a Radial Basis Function (RBF) network. The input layer of the proposed network propagates a sample pair of two random variables to the hidden layer. The propagated samples are then transformed by a set of Gaussian RBF kernels with randomly determined kernel centers and widths similar to that in an extreme learning machine. The output layer adopts a linear weighting scheme which can be analytically estimated. Our empirical results show that the proposed estimator outperforms the competing state-of-the-art MI estimators in terms of computational efficiency while showing the comparable estimation accuracy performance. Moreover, the proposed model achieves promising results in an application study of time-series change-points detection and driving stress analysis.

Keywords: Density Ratio Approximation, Mutual Information Estimation, Extreme Learning Machine, Change-Points Detection, Electrocardiogram, Driving Stress

*Corresponding author
Email address: ezplin@ntu.edu.sg (Zhiping Lin)

1. Introduction

The Shannon's Mutual Information (MI): $I(\mathbf{X};\mathbf{Y}) = E\left[\log \frac{p_{xy}(\mathbf{x},\mathbf{y})}{p_x(\mathbf{x})p_y(\mathbf{y})}\right]$ has been widely deployed in measuring underlying dependencies between two random variables $\mathbf{X}$ and $\mathbf{Y}$. The $I(\mathbf{X};\mathbf{Y})$ vanishes if the evaluated random variables are independent to each other. These are important traits in measuring similarity between distributions, enabled MI to be deployed in a wide range of applications: pattern recognition (e.g. event detection [1, 2] and feature selection [3, 4, 5]) and time-series analysis [6, 7] to name just a few.

Estimating MI, however, is a difficult problem [8]. The difficulty comes from its algorithmic dependence on Probability Density Function (PDF) calculation. Particularly, a PDF for each of the marginal densities $p_x(\mathbf{x})$ and $p_y(\mathbf{y})$, and the joint probability density $p_{xy}(\mathbf{x},\mathbf{y})$ should be estimated from the given data samples [9, 6]. A Parzen window density estimation technique [10] (a.k.a. Kernel Density Estimator (KDE) [9, 6]) is a common choice for the estimation. Despite the determination of kernel parameters (e.g. kernel width in Gaussian kernel) which is generally nontrivial, we emphasize here that KDE produces a reliable estimation accuracy only when enough samples are available. Unfortunately, the computational efficiency is significantly degraded if the data size increases.

Recently, attempts were made to enhance the computational efficiency of MI estimation while maintaining the estimation accuracy [11, 12, 13]. The core idea of the attempts was to directly estimate the density ratio $r(\mathbf{x},\mathbf{y}) = \frac{p_{xy}(\mathbf{x},\mathbf{y})}{p_x(\mathbf{x})p_y(\mathbf{y})}$ instead of separately estimating the individual densities [14]. Based on this idea, the following estimators have been proposed: Least-Squares Mutual Information (LSMI) [11], multiplicative LSMI (mLSMI) [12] and Maximum-Likelihood Mutual Information (MLMI) [13]. These estimators with appropriate parameter settings outperformed the KDE in terms of estimation error and computational efficiency [11, 12, 13].

Technically, we believe these LSMI, mLSMI and MLMI methods are still somewhat limited. Their major drawback comes from the grid search-based hyper-parameter determination process. Particularly, these methods select a value for a parameter among predefined candidates based on a cross-validation [11, 12, 13]. If we have more than

one parameter, the number of iterations is multiplicatively increased. Moreover, the predefined candidates may not encompass the entire searching range. This is particularly true for non-stationery inputs. These technical issues motivate us to investigate into the proposal of a parameter tuning-free MI estimator which would be computationally more efficient.

In this paper, we propose an Extreme Learning Machine based model for Mutual Information (ELM-MI) estimation. The core idea is to approximate the density ratio $r(\mathbf{x},\mathbf{y})$ by a simple random RBF network. Essentially, the input layer receives and propagates a pair of random variables to the hidden layer. The propagated sample pair is then processed by a set of RBF kernels with randomly determined kernel centers and widths similar to that in [15, 16]. The hidden node outputs are finally incorporated with a deterministically learned output weights to yield the final output.

As an application, the proposed ELM-MI is applied to a time-series change-points detection problem. ELM-MI is then utilized in an empirical analysis where the relationships between driving a car and human stress are investigated. According to [17, 18], Electrocardiogram (ECG) time-series can effectively capture the non-linear dynamics of changes [19, 20] in human stress. ELM-MI can be a computationally efficient and effective tool for the stress analysis. For the analysis, in-house driving ECG data were collected.

The main contributions of this paper can be enumerated as follows: i) proposal of a computationally efficient parameter-tuning free mutual information estimator, ii) proposal of a novel framework for time-series change-points detection, iii) proposal of information theory-based driving stress analysis, iv) provision of extensive empirical results to validate the proposed model.

The remaining of this paper is organized as follows: brief descriptions of MI and one of its estimators are presented in Section 2 for immediate reference. The proposed ELM-based MI estimator is then presented in Section 3. Section 4 presents our experimental setting and numerical results with discussions. In Section 5, we present protocols for ECG data collection, driving scenarios and empirical analysis results. Some concluding remarks are presented in Section 6.

2. Preliminaries

In this section, we provide a brief description of the Mutual Information (MI) between two random variables [21], followed by a brief summary of a least-squares approach [11] is provided for immediate reference.

2.1. Mutual information

Suppose we have n independent and identically distributed (*i.i.d.*) sample pairs $\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathcal{D}_{\mathbf{X}}, \mathbf{y}_i \in \mathcal{D}_{\mathbf{Y}}\}_{i=1}^{n}$ drawn from a joint distribution with density $p_{xy}(\mathbf{x}, \mathbf{y})$, where $\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}} \subset \mathbb{R}^d$ respectively denotes the data domain. The mutual information between the two continuous random variables $\mathbf{X}$ and $\mathbf{Y}$ is defined as follows [21, 11]:

$$\begin{aligned} I(\mathbf{X};\, \mathbf{Y}) &= \iint p_{xy}(\mathbf{x}, \mathbf{y}) \log\left(\frac{p_{xy}(\mathbf{x}, \mathbf{y})}{p_x(\mathbf{x})\, p_y(\mathbf{y})}\right) d\mathbf{x} d\mathbf{y} \\ &= \iint p_{xy}(\mathbf{x}, \mathbf{y}) \log r(\mathbf{x}, \mathbf{y})\, d\mathbf{x} d\mathbf{y}, \end{aligned} \tag{1}$$

where $r(\mathbf{x}, \mathbf{y}) = \frac{p_{xy}(\mathbf{x}, \mathbf{y})}{p_x(\mathbf{x}) p_y(\mathbf{y})}$ indicates the density ratio, and $p_x(\mathbf{x})$ and $p_y(\mathbf{y})$ respectively denotes the marginal density of $\mathbf{x}$ and $\mathbf{y}$.

2.2. Least-Squares Mutual Information (LSMI)

The main idea of LSMI [11, 22] is to directly estimate the density ratio $r(\mathbf{x}, \mathbf{y})$ in (1) without going through estimation of the individual distributions: $p_x(\mathbf{x}), p_y(\mathbf{y})$ and $p_{xy}(\mathbf{x}, \mathbf{y})$ [23]. Particularly, the ratio $r(\mathbf{x}, \mathbf{y})$ is approximated by a linear model as:

$$g(\mathbf{x}, \mathbf{y}) = \boldsymbol{\beta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}), \tag{2}$$

where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_N]^T$ consists of model parameters to be learned from data samples, T denotes the transpose, and $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = [\phi_1(\mathbf{x}, \mathbf{y}), \cdots, \phi_N(\mathbf{x}, \mathbf{y})]^T$ contains basis functions. To capture the inter-correlation between two random variables, LSMI deploys Gaussian kernels located on paired data samples [11, 22] as follows:

$$\phi_l(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_l^x\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\mathbf{y} - \boldsymbol{\mu}_l^y\|^2}{2\sigma^2}\right), l = 1, \cdots, N,$$

where $\boldsymbol{\mu}_l^x, \boldsymbol{\mu}_l^y \in \mathbb{R}^d$ denotes the l-th kernel center points and σ indicates the kernel width.

The criterion function for estimating the optimal $\boldsymbol{\beta}$ is [11, 22]:

$$\begin{aligned} J(g) &= \frac{1}{2}\iint \left(g(\mathbf{x},\mathbf{y}) - r(\mathbf{x},\mathbf{y})\right)^2 p_x(\mathbf{x})\, p_y(\mathbf{y})\, d\mathbf{x}d\mathbf{y} \\ &= \frac{1}{2}\iint g(\mathbf{x},\mathbf{y})^2\, p_x(\mathbf{x})\, p_y(\mathbf{y})\, d\mathbf{x}d\mathbf{y} - \iint g(\mathbf{x},\mathbf{y})\, p_{xy}(\mathbf{x},\mathbf{y})\, d\mathbf{x}d\mathbf{y} + C \\ &= \frac{1}{2}\boldsymbol{\beta}^T \mathbf{H}\boldsymbol{\beta} - \mathbf{h}^T\boldsymbol{\beta} + C, \end{aligned} \tag{3}$$

where $\mathbf{H} = \iint \boldsymbol{\phi}(\mathbf{x},\mathbf{y})\, \boldsymbol{\phi}(\mathbf{x},\mathbf{y})^T\, p_x(\mathbf{x})\, p_y(\mathbf{y})\, d\mathbf{x}d\mathbf{y}$, $\mathbf{h} = \iint \boldsymbol{\phi}(\mathbf{x},\mathbf{y})\, p_{xy}(\mathbf{x},\mathbf{y})\, d\mathbf{x}d\mathbf{y}$ and $C = \frac{1}{2}\iint r(\mathbf{x},\mathbf{y})\, p_{xy}(\mathbf{x},\mathbf{y})\, d\mathbf{x}d\mathbf{y}$ is a constant term. By approximating the expectations in $\mathbf{H}$ and $\mathbf{h}$ by the empirical averages, including a regularization term $\lambda \geq 0$, excluding the constant term C and solving the linear optimization, the weights can be learned as:

$$\widehat{\boldsymbol{\beta}} = \left(\widehat{\mathbf{H}} + \lambda \mathbf{I}\right)^{-1} \widehat{\mathbf{h}}, \tag{4}$$

where $\widehat{\mathbf{H}} = \frac{1}{n^2}\sum_{i,j=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_j)\, \boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_j)^T$, $\mathbf{I}$ is an identity matrix with the same dimension as $\widehat{\mathbf{H}}$, and $\widehat{\mathbf{h}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_i)$. The optimal values for σ and λ can be determined among some predefined candidates based on a cross-validation grid search [11, 22]. Once the output weights vector $\widehat{\boldsymbol{\beta}}$ is learned, LSMI can be computed as follows:

$$\text{LSMI} = \frac{1}{2}\widehat{\boldsymbol{\beta}}^T \widehat{\mathbf{H}}\widehat{\boldsymbol{\beta}} - \widehat{\mathbf{h}}^T\widehat{\boldsymbol{\beta}}. \tag{5}$$

3. Extreme learning machine based mutual information estimation

In this section, we propose a novel random RBF network for efficient mutual information estimation. The key idea is to approximate the density ratio $r(\mathbf{x},\mathbf{y}) = \frac{p_{xy}(\mathbf{x},\mathbf{y})}{p_x(\mathbf{x})p_y(\mathbf{y})}$ in (1) by a simple network with randomly constructed (Gaussian) Radial Basis Function (RBF) kernels similar to that in [15, 16]. Due to the random property, the proposed model is parameter tuning-free which brings us to a high computational efficiency. The following subsections provide details of the proposed network.

3.1. The proposed network architecture

Suppose we have n number of *i.i.d.* sample pairs $\{(\mathbf{x}_i,\mathbf{y}_i) \mid \mathbf{x}_i \in \mathcal{D}_\mathbf{X}, \mathbf{y}_i \in \mathcal{D}_\mathbf{Y}\}_{i=1}^{n}$ drawn from a joint distribution with density $p_{xy}(\mathbf{x},\mathbf{y})$, where $\mathcal{D}_\mathbf{X}, \mathcal{D}_\mathbf{Y} \subset \mathbb{R}^d$ respectively denotes the data domain. With N hidden nodes in which each node consists

of a RBF kernel $\phi_l\left(\boldsymbol{\mu}_l^x, \boldsymbol{\mu}_l^y, \sigma_l, \mathbf{x}_i, \mathbf{y}_j\right)$, $l = 1, 2, \cdots, N$, the proposed ELM-MI model is mathematically defined as follows (see Fig. 1):

$$g\left(\mathbf{x}_i, \mathbf{y}_j\right) = \sum_{l=1}^{N} \beta_l \phi_l\left(\boldsymbol{\mu}_l^x, \boldsymbol{\mu}_l^y, \sigma_l, \mathbf{x}_i, \mathbf{y}_j\right),\ i, j = 1, \cdots, n,$$

where β_l indicates an output weight connecting the l-th hidden node to the output node and $\phi_l\left(\boldsymbol{\mu}_l^x, \boldsymbol{\mu}_l^y, \sigma_l, \mathbf{x}_i, \mathbf{y}_j\right) = \exp\left(-\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_l^x\|^2}{2\sigma_l^2}\right)\exp\left(-\frac{\|\mathbf{y}_j - \boldsymbol{\mu}_l^y\|^2}{2\sigma_l^2}\right)$. $\boldsymbol{\mu}_l^x \in \mathbb{R}^d$ and $\boldsymbol{\mu}_l^y \in \mathbb{R}^d$ respectively indicates the l-th kernel's centers randomly chosen from $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ [11, 15] and $\sigma_l \in \mathbb{R}$ indicates the kernel width (a.k.a. impact width or length scale) which are determined randomly similar to that in [16]. The proposed model is graphically illustrated in Fig. 1.

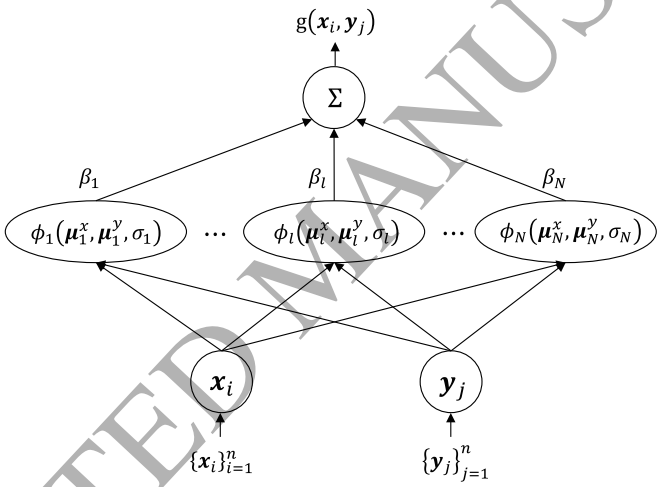


Figure 1: An illustration of the proposed ELM-MI model.

As shown in Fig. 1, the input layer of the proposed network consists of two input nodes. Each node receives and propagates an input sample, either $\mathbf{x}_i$ or $\mathbf{y}_j$, to the hidden nodes at a time in which a non-linear Gaussian kernel mapping is performed. Subsequently, the non-linearly mapped samples are combined by means of the weighted sum in the output layer.

Assume that the density ratio $r(\mathbf{x},\mathbf{y})$ is a continuous function and integrable over the entire range. Then, the ratio can be approximated by the Gaussian kernel (i.e. a universal kernel) without loss of generality [24]. We know that the Gaussian kernel with randomly generated kernel centers and width still satisfies universal approximation capability. Moreover, the authors of [25] have rigorously proven the convergence property of the Gaussian kernel. Based on these observations, we can say that the proposed ELM-MI model for the density ratio estimation satisfies the convergence property. **The Gaussian kernel has also been widely applied in kernel adaptive filters [26, 27, 28, 29].**

3.2. Learning the output weights, $\boldsymbol{\beta}$

Recall that both kernel centers $\boldsymbol{\mu}_l^x$, $\boldsymbol{\mu}_l^y$ and kernel width σ_l are preset to randomly selected input samples and random values, respectively. With these pre-fixed parameters, the problem of mutual information estimation boils down to learning discriminant output weights vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_N]^T$. Similar to that in [11] (see Section 2.2 for details), the output weights $\boldsymbol{\beta}$ is learned by the following closed-form solution (similar to (4)) in which the criterion function in (3) is minimized:

$$\widehat{\boldsymbol{\beta}} = \left(\widehat{\mathbf{H}} + \lambda \mathbf{I}\right)^{-1} \widehat{\mathbf{h}}, \tag{6}$$

where

$$\widehat{\mathbf{H}} = \frac{1}{n^2}\sum_{i,j=1}^{n} \boldsymbol{\phi}_{i,j}\boldsymbol{\phi}_{i,j}^T \in \mathbb{R}^{N\times N}, \;\; \widehat{\mathbf{h}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}_{i,i} \in \mathbb{R}^N, \tag{7}$$

and $\boldsymbol{\phi}_{i,j} = \left[\phi_1\left(\boldsymbol{\mu}_1^x, \boldsymbol{\mu}_1^y, \sigma_1, \mathbf{x}_i, \mathbf{y}_j\right), \cdots, \phi_N\left(\boldsymbol{\mu}_N^x, \boldsymbol{\mu}_N^y \sigma_N, \mathbf{x}_i, \mathbf{y}_j\right)\right]^T \in \mathbb{R}^N$, λ denotes a regularization factor, and $\mathbf{I}$ is an identity matrix with the same dimension as $\widehat{\mathbf{H}}$. After learning, the estimated MI by ELM-MI can be computed as follows (5):

$$\text{ELM-MI} = \frac{1}{2}\widehat{\boldsymbol{\beta}}^T \widehat{\mathbf{H}}\widehat{\boldsymbol{\beta}} - \widehat{\mathbf{h}}^T\widehat{\boldsymbol{\beta}}.$$

3.3. Stabilization of the estimation output by an ensemble

Due to the random setting for the kernel centers $\boldsymbol{\mu}_l^x$, $\boldsymbol{\mu}_l^y$ and kernel width σ_l, the estimation output of ELM-MI may vary over trials. The simplest and yet effective way to remedy such fluctuations is to combine multiple outputs over multiple trials. Such

ensemble approach could effectively handle unpredictable inputs [30]. We refer the readers to [30] for detailed discussions regarding ELM-based voting and ensemble.

In this work, we take an average over M number of ELM-MIs as illustrated in Fig. 2. Particularly, each ELM-MI works based on randomly determined kernel centers and width. **The proposed ELM-MI model with ensemble is summarized in Algorithm 1.** It is worth to note that our experiment (see Section 4.2 and Fig. 6 for details) showed that the overall MI estimation speed of ELM-MI at $M = 10$ is still faster than the other compared MI estimators.

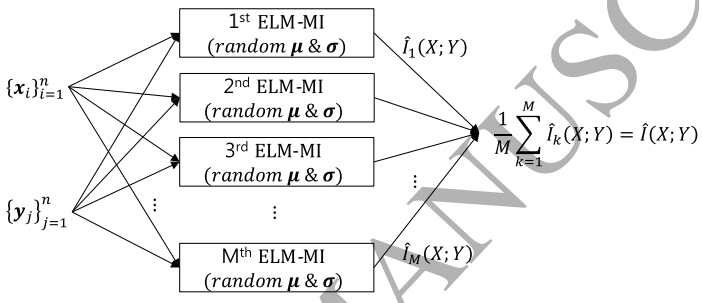


Figure 2: An illustration of the ensemble ELM-MI over M repetitions.

Algorithm 1 ELM-MI

Input: A pair of random variables $\{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathcal{D}_\mathbf{X}, \mathbf{y}_i \in \mathcal{D}_\mathbf{Y}\}_{i=1}^n$ where $\mathcal{D}_\mathbf{X}, \mathcal{D}_\mathbf{Y} \subset \mathbb{R}^d$, RBF Kernel function ϕ, number of hidden nodes N and ensemble size M.
Normalize the $\mathbf{x}_i$ and $\mathbf{y}_i$ respectively using the min-max normalization
for $k \leftarrow 1$ **to** M **do**
  for $l \leftarrow 1$ **to** N **do**
    $\left(\boldsymbol{\mu}_l^x, \boldsymbol{\mu}_l^y\right) \leftarrow$ a pair $\left(\mathbf{x}_i, \mathbf{y}_j\right), i, j \in \{1, \cdots, n\}$ randomly chosen from the input
    $\sigma_l \leftarrow$ uniformly distributed random numbers $\mathcal{U}(0,1)$
  end for
  Estimate $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{H}}$ using (7), and $\widehat{\boldsymbol{\beta}}$ using (6)
  $\hat{I}_k \leftarrow \frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\mathbf{H}}\widehat{\boldsymbol{\beta}} - \widehat{\mathbf{h}}^T\widehat{\boldsymbol{\beta}}$
end for
return $\hat{I} \leftarrow average\left(\hat{I}_k\right)$

3.4. Comparison between ELM-MI and LSMI

We note that the main idea of ELM-MI model was derived from LSMI [11]. Thus, both models share some fundamental idea and structures. However, it is worth to emphasize that the major drawback of LSMI is efficiently resolved by the proposed model. In this section, we briefly discuss about the algorithmic differences between them.

Recall that LSMI selects an optimal value for the kernel width σ and regularization factor λ respectively from some predefined candidates. The adopted selection strategy in [11] was a cross-validation-based grid search (e.g. a five-fold cross-validation). Such grid search increases the computational cost significantly. Assume that for example we have nine candidates respectively for σ and λ, and perform a five-fold cross validation. Under such setting, the best combination of parameters (σ_{opt}, λ_{opt}) can be obtained over 405 $(= 9 \times 9 \times 5)$ repetitions.

Different from the LSMI, the proposed model is free from such tedious grid search. As discussed in Section 3.1, σ_l, $l = 1, \cdots, N$, are prefixed to random values with a fixed small value for λ similar to that in [16]. The universal approximation property of the RBF network with random setting has been proved in [16, 31] which also can be applied to ELM-MI. From our experiment (see Section 4.2), ELM-MI outperforms LSMI in terms of MI estimation error and CPU time performances.

3.5. A MI-based framework for time-series change-points detection

In this subsection, we propose a novel MI-based framework for **retrospective** time-series change-points detection. For the MI measurement, the proposed ELM-MI is deployed thanks to its computational efficiency and reliable estimation accuracy. Although ELM-MI can compute MI between two multi-dimensional $(d > 1)$ random variables, this paper focuses on only a univariate $(d = 1)$ time-series.

Let $y_t \in \mathbb{R}$, $t = 1, \cdots, n$, denotes univariate time-series at time t and $y_{t-\tau}$ denotes its delayed version at time $t - \tau$ where $\tau = 1, 2, \cdots$. Let $p_{yy}(y_{t-\tau}, y_t)$ indicates a joint probability distribution between the time-series y_t measured at time $t - \tau$ and t. Similarly, $p_y(y_t)$ and $p_y(y_{t-\tau})$ respectively denotes the marginal distribution for y_t and $y_{t-\tau}$. With these time-series dependent notations, the formulation for MI computation (1) can be

re-defined for time-series data (a.k.a. Time-Delay Mutual Information (TDMI) [6]) as follows:

$$I(Y_{t-\tau};\, Y_t) = \iint p_{yy}\left(y_{t-\tau}, y_t\right) \log\left(\frac{p_{yy}\left(y_{t-\tau}, y_t\right)}{p_y\left(y_{t-\tau}\right) p_y\left(y_t\right)}\right) dy_{t-\tau} dy_t. \tag{8}$$

TDMI quantifies the co-dependence between the time points $t-\tau$ and t by considering shared previous information content as a function of time [32, 7]. Recall that the proposed ELM-MI model can efficiently capture the second order statistics of non-linear relationships between the time points. Moreover, the non-linear mapping (e.g. Gaussian kernel) of ELM-MI mitigates the effect of τ in TDMI estimation.

The proposed detection framework is essentially a sequential process of TDMI computations as shown in Fig. 3 and **summarized in Algorithm 2**. Particularly, ELM-MI is used to compute TDMI value between two overlapped time intervals (windows) of length k (denoted as $\mathbf{w}_t \in \mathbb{R}^k$ in the figure) with a time delay τ. The computed TDMI values $I\left(Y_{w_1}, Y_{w_2}\right), I\left(Y_{w_2}, Y_{w_3}\right), \cdots$ can be considered as change scores similar to that in [33]. If $I\left(Y_{w_{t-1}}, Y_{w_t}\right)$ is a local peak above a predefined threshold, it can be a candidate for a change-point. Our running example which is shown in Fig. 3 is based on $k = 4$ and $\tau = 1$.

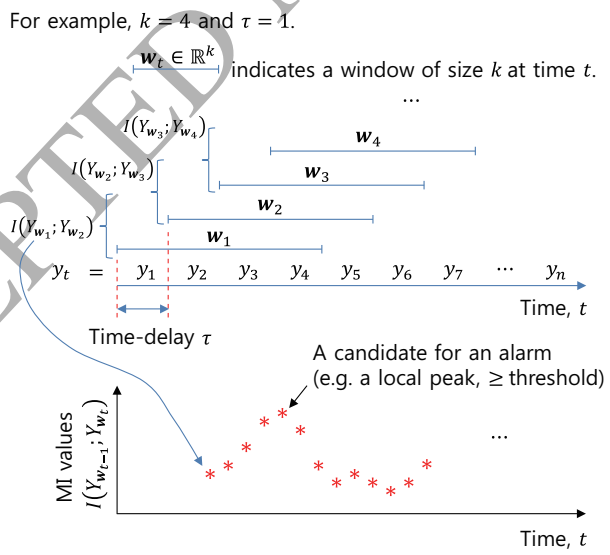


Figure 3: An illustration of windowing for ELM-MI based time-series change-points detection.

Algorithm 2 ELM-MI based time-series change-points detection

Given: Univariate time-series $\{y_t | t = 1, \cdots, n\}$, window size k, time-delay τ, RBF Kernel function ϕ, number of hidden nodes N, ensemble size M and threshold η.
for $t \leftarrow \tau$ **to** $n-k$ **do**
    $\mathbf{w}_{t-\tau} \leftarrow [y_{t-\tau}, y_{t-\tau+1}, \cdots, y_{t-\tau+k}]^T \in \mathbb{R}^k$
    $\mathbf{w}_t \leftarrow [y_t, y_{t+1}, \cdots, y_{t+k}]^T \in \mathbb{R}^k$
    $\hat{I}_{t+\lfloor k/2 \rfloor} \leftarrow$ ELM-MI $(\mathbf{w}_{t-1}, \mathbf{w}_t, \phi, N, M)$ shown in Algorithm 1
    if $\hat{I}_{t+\lfloor k/2 \rfloor} > \eta$ **then**
        A candidate for a change-point detection alarm
    end if
end for
Report only local peaks among the resulted candidates as detected change-points

4. Numerical experiments

The main goal of this experimental study is to verify the effectiveness and computational efficiency of the proposed ELM-MI model for the MI estimation. We will also show its usefulness in an application of time-series change-points detection. **To achieve these two goals, two experiments have been conducted using artificial and real datasets (see Table 1)**. The following subsections provide details of our experimental design, settings and empirical results with discussions.

4.1. Experimental setting

As shown in Table 1, two experiments (denoted respectively as Experiment I and II in the table) have been conducted to evaluate the proposed ELM-MI model. Particularly, under Experiment I, we evaluate ELM-MI model in terms of MI estimation error and CPU time (s) performance using four artificial datasets. These datasets were borrowed from [11, 13] in which competing MI estimators were proposed. For performance benchmarking, the obtained performances (e.g. MI estimation error and CPU time) of ELM-MI are compared with these of KDE [9, 6], LSMI [11], mLSMI [12] and MLMI [13].

Table 1: Two numerical experiments for performance evaluation of the proposed ELM-MI model.

Experiments	Brief descriptions	Dataset	Performance benchmarking with
Experiment I	Performance evaluation for mutual information estimation	Four artificial toy datasets from [11]	LSMI [11], mLSMI [12], MLMI [13], KDE [34]
Experiment II	Performance evaluation for time-series change-points detection	**Artificial dataset [33], Well log dataset [35]**	RuLSIF [33]

* Abbreviation:
LSMI: Least-Squares Mutual Information,
mLSMI: Multiplicative LSMI,
MLMI: Maximum-Likelihood Mutual Information,
KDE: Kernel Density Estimation,
RuLSIF: Relative unconstrained Least-Squares Importance Fitting.

Under Experiment II, the proposed ELM-MI model is applied to an application of time-series change-points detection. **The applicability of ELM-MI to this application is validated using an artificial time-series dataset [33] and a real time-series dataset [35].** For performance comparison, the change-points detection performances of ELM-MI are compared with a competing state-of-the-art method, namely Relative unconstrained Least-Squares Importance Fitting (RuLSIF) [33]. **RuLSIF is included in our comparison study due to its good change-points detection accuracy performance. For example, RuLSIF produced higher detection accuracy than that of unconstrained Least-Squares Importance Fitting (uLSIF) [33] and Kullback-Leibler Importance Estimation Procedure (KLIEP) [36].**

For LSMI, mLSMI, MLMI and RuLSIF, open source codes released by the author group[1] have been utilized in this work. For KDE, an open source code[2] has been utilized. All experiments were performed on a PC of 3.4GHz with 8G RAM under Matlab platform (of version R2014b) [37].

[1] Source code available: http://www.ms.k.u-tokyo.ac.jp/software.html

[2] Source code available: http://www.mathworks.com/matlabcentral/fileexchange/29039-mutual-information-2-variablle/content/MutualInfo.m

4.2. Experiment I: Mutual information estimation

Under this experiment, as shown in Table 1, the proposed ELM-MI model is evaluated in terms of its effectiveness and computational efficiency for MI estimation. The following subsections provide details of datasets, experimental protocols and results related to Experiment I.

4.2.1. Datasets for MI estimation

The following four artificial datasets (see Fig. 4) borrowed from [11, 13] have been utilized in this experiment.

1. *Linear dependence (see Fig. 4 (a)):* y has a linear dependence on x as

$$x \sim \mathcal{N}(x;\, 0,\, 0.5) \;\text{ and }\; y|x \sim \mathcal{N}(y;\, 3x,\, 1),$$

   where $\mathcal{N}\left(x;\, \mu,\, \sigma^2\right)$ denotes the Gaussian distribution x with mean μ and variance σ^2.

2. *Non-linear dependence with correlation (Fig. 4 (b)):* y has a quadratic dependence on x as

$$x \sim \mathcal{N}(x;\, 0,\, 1) \;\text{ and }\; y|x \sim \mathcal{N}\left(y;\, x^2,\, 1\right).$$

3. *Non-linear dependence without correlation (Fig. 4 (c)):* y has a lattice-structured dependence on x as

$$x \sim \mathcal{U}(x;\, -0.5,\, 0.5) \;\text{ and }\; y|x \sim \begin{cases} \mathcal{N}\left(x;\, 0,\, \frac{1}{3}\right) & \text{if } |x| \leq \frac{1}{6}, \\ \frac{1}{2}\mathcal{N}\left(x;\, 1,\, \frac{1}{3}\right) + \frac{1}{2}\mathcal{N}\left(x;\, -1,\, \frac{1}{3}\right) & \text{otherwise,} \end{cases}$$

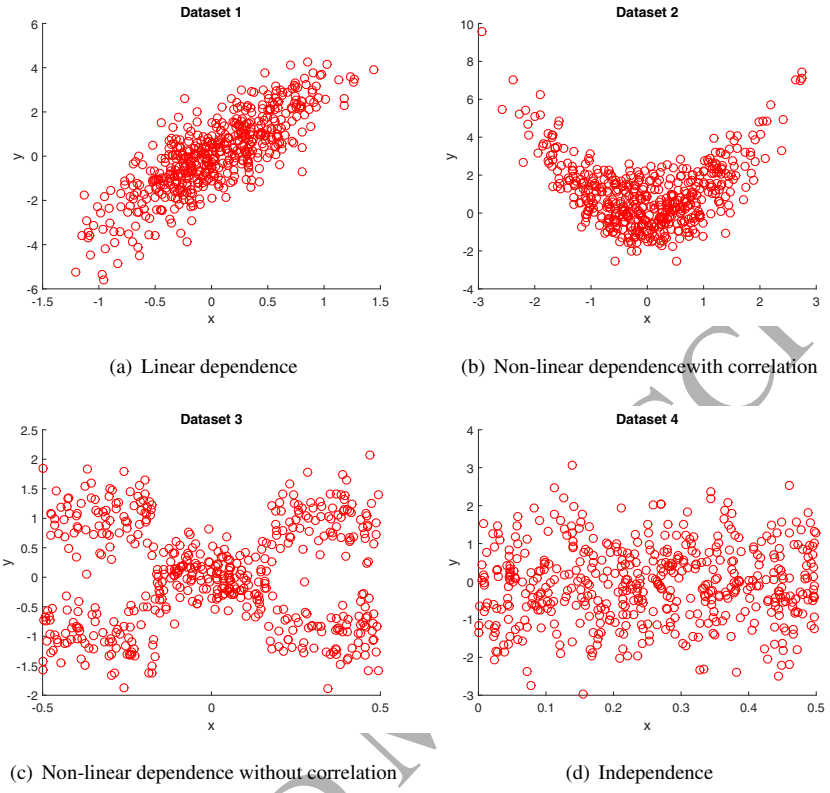
   where $\mathcal{U}(x;\, a,\, b)$ denotes the uniform distribution x over the interval (a, b).

4. *Independence (Fig. 4 (d)):* x and y are independent to each other as

$$x \sim \mathcal{U}(x;\, 0,\, 0.5) \;\text{ and }\; y|x \sim \mathcal{N}(y;\, 0,\, 1).$$

4.2.2. Protocols for MI estimation

The proposed ELM-MI model has three adjustable parameters, namely: the number of hidden nodes N, the regularization factor λ and the ensemble size M. According



(a) Linear dependence

(b) Non-linear dependencewith correlation

(c) Non-linear dependence without correlation

(d) Independence

Figure 4: Artificial datasets (borrowed from [11, 13]) utilized for performance evaluation of the compared MI estimators.

to [16, 38], N is closely related to the model accuracy. To observe its effect on MI estimation error, different numbers of hidden nodes $N \in \{50, 100, 300\}$ are experimented. Each of the settings is respectively denoted as ELM-MI(50), ELM-MI(100) and ELM-MI(300) in results plots (see Fig. 5).

From our preliminary experiments, it was observed that the λ setting does not affect much on MI estimation error. This is particularly true when the input samples are properly normalized, mapped by sufficient number of random RBF kernels, and a small value of λ. Based on these observations, we fix $\lambda = 10^{-3}$ for all subsequent experiments. **Our empirical evaluations also showed that the settings of** $5 \sim 10$ **for** M **is sufficient to achieve a stable performance. In this experiment of MI evaluation,**

we set $M = 10$.

Unlike the proposed model, the compared LSMI, mLSMI and MLMI models locate the best value for λ and/or kernel width σ based on a grid search [11, 12, 13]. Candidates deployed for σ contain nine values of 50 logarithmically spaced points between decades 10^{-2} and 10^{2}. Similarly, candidates for λ include nine values 50 logarithmically spaced points between decades 10^{-3} and 10^{1}. A 5-fold cross-validation was performed for tuning both σ and λ from the candidates.

Similar to that in [11, 13], the MI estimation capability of the compared estimators was measured by an approximation error which is defined as:

$$\text{MI approximation error} = |\hat{I}(X;Y) - I(X;Y)|, \tag{9}$$

where $\hat{I}(X;Y)$ denotes the estimated MI, $I(X;Y)$ indicates the Shannon mutual information (1), and $|\cdot|$ denotes an absolute operation. For statistical evidence, the compared methods are repeated 100 times in which the average values are recorded. **For ELM-MI, we take an average over 10 trials at $M = 10$ to have similar statistics (note: 10 trials $\times$ $M\,(= 10)$ equals to 100 repetitions).**

4.2.3. Results and discussions

Fig. 5 which is plotted over the number of samples $n \in \{20, 40, \cdots, 300\}$, shows the average MI estimation error computed using (9) over 100 trials. Fig. 5 (a), (c), (e) and (g) respectively shows the errors resulted using Dataset 1 to Dataset 4. Their zoom-in view is respectively shown in Fig. 5 (b), (d), (f) and (h).

As shown in Fig. 5 (a) and (b), KDE at $n = 180$ and ELM-MI(100)[3] at $n = 300$ produced the lowest error values among the compared estimators. However, mLSMI produced the best performance when fewer samples (e.g. $n \leq 80$) are available for the estimation. Both LSMI and MLMI produced similar performances over the entire range of n except for $n \leq 60$. Among the three N settings, ELM-MI(100) yielded the lowest MI estimation error values than that of ELM-MI(50) and ELM-MI(300).

Different from Dataset 1 (Fig. 5 (a) and (b)), both mLSMI and MLMI models

[3]The number within the parenthesis indicates the number of utilized hidden nodes N.

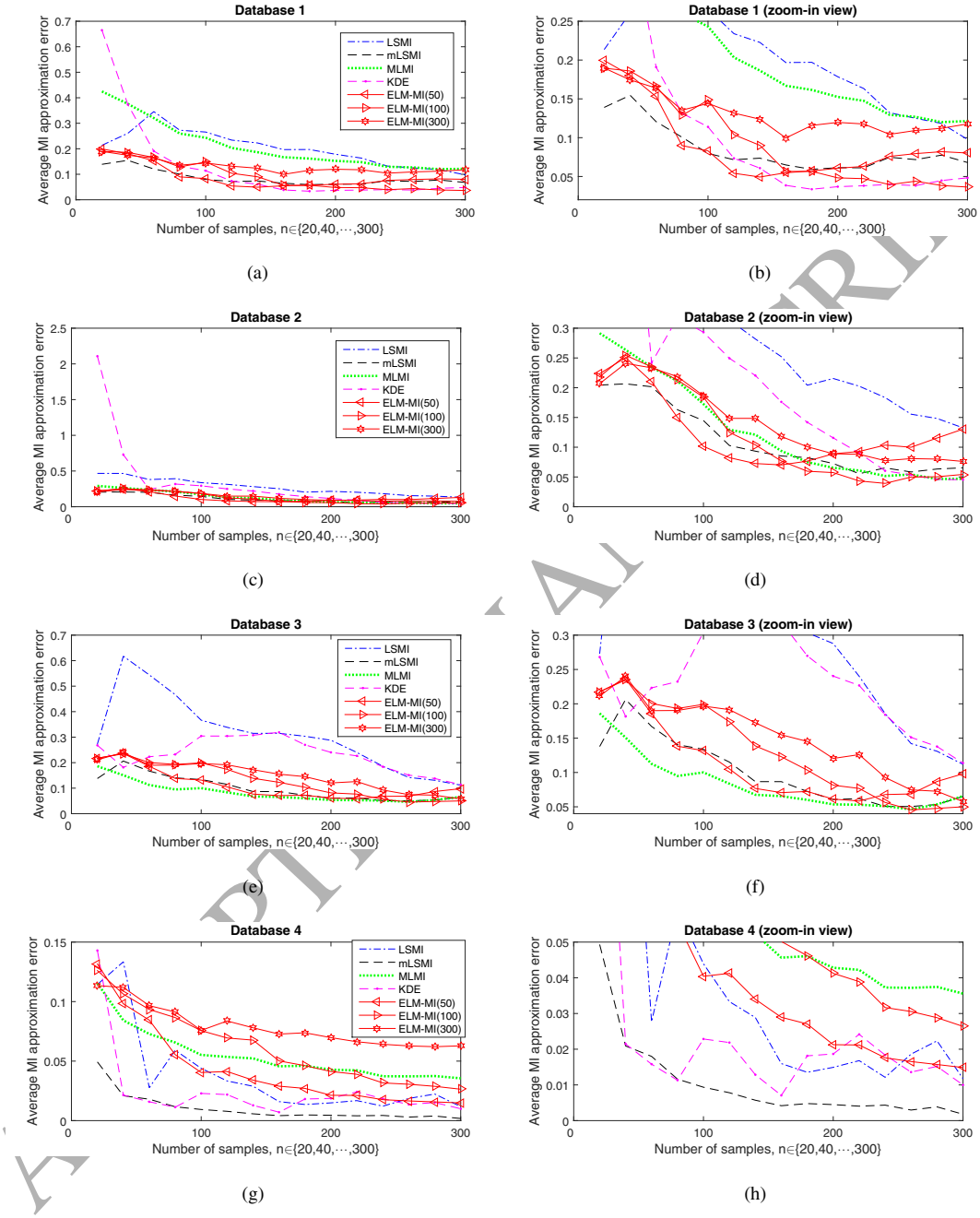
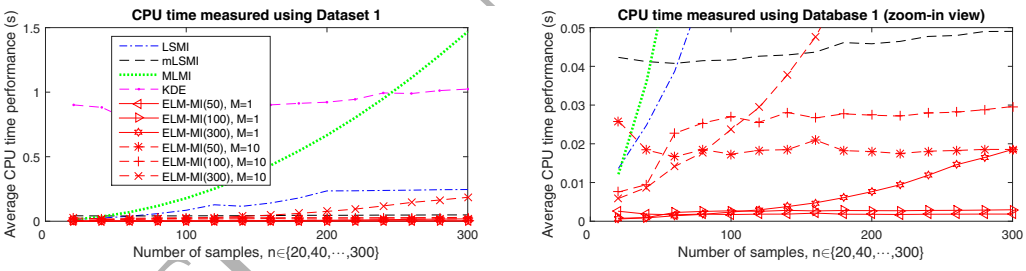


Figure 5: Comparison of the average MI approximation error (over 100 trials) among the compared methods resulted using (a) Dataset 1, (c) Dataset 2, (e) Dataset 3 and (g) Dataset 4. Sub-figures (b), (d), (f) and (h) are a zoom-in view of (a), (c), (e) and (g), respectively.

worked very well for non-linearly dependent datasets (e.g. Dataset 2 shown in Fig. 5 (c) and (d), and Dataset 3 shown in Fig. 5 (e) and (f)). Among the proposed methods, ELM-MI(100) again produced the best estimation performance for both datasets. LSMI produced the highest error values among the compared methods.

When the input random variables are independent to each other (i.e. Dataset 4), mLSMI produced the best MI estimation performance among the compared estimators (see Fig. 5 (g) and (h)). KDE produced the second best estimation performance. Under this dataset, ELM-MI(50) outperformed the other two settings (e.g. ELM-MI(100) and ELM-MI(300)). Interestingly, LSMI produced the third best estimation performance under this dataset.

The investigated estimators are now evaluated and compared in terms of their CPU time performance (measured in seconds) as shown in Fig. 6. Similar to those estimation error plots (see Fig. 5), the measured CPU time values are also plotted over the number of samples $n \in \{20, 40, \cdots, 300\}$. **In order to portray an overview over a range of the computational cost, the CPU time values of ELM-MI measured at $M = 1$ are also included in the same figure apart from those measured at $M = 10$.**



(a) (b)

Figure 6: CPU time (s) elapsed for MI estimation using Dataset 1: (a) an overview and (b) a zoom-in view of (a). **To provide a range of computational cost, CPU time for ELM-MI was measured at $M = 1$ and $M = 10$, respectively.**

Among the compared methods, KDE at $n < 260$ and MLMI at $n > 260$ consumed the most time while **ELM-MI(50) at** $M = 10$ takes the least time. LSMI was faster than mLSMI at $n \leq 80$. However, if the sample size n is increased (e.g. $n > 100$), LSMI spent more time than mLSMI. **Among the three settings of N, as expected,**

ELM-MI(50) and ELM-MI(300) at $M = 10$ have respectively consumed the least and the most time. **The figure also reveals that ELM-MI(300) at** $M = 10$ **is faster than mLSMI when** $n < 150$**.**

To summarize, the proposed ELM-MI model works well for MI estimation regardless of the underlying data distribution. Particularly, ELM-MI has shown MI estimation error similar to or lower than that of competing state-of-the-art methods under linear dependence (see Fig. 5 (a) and (b)), and non-linear dependence with (see Fig. 5 (c) and (d)) and without (see Fig. 5 (e) and (f)) correlation. When the input data are independent of each other, mLSMI, LSMI and KDE outperformed the proposed method. However, the estimation error gap is almost negligible (e.g. < 0.03). Besides the accuracy issue, the proposed estimator significantly outperformed all the compared methods in terms of CPU time performance (see Fig. 6). **For example, ELM-MI(100) at** $M = 10$ **is about 2**$\sim$**4 times faster than that of mLSMI over the entire range of experimented** n**. If we reduce the ensemble size** M **to 1, then ELM-MI(100) is about 2.5**$\sim$**20 times faster than that of mLSMI.**

4.3. Experiment II: Time-series change-points detection

As shown in Table 1, the applicability of ELM-MI model to time-series change-points detection is evaluated under this experiment. In the following subsections, we discuss the utilized datasets, experimental protocols and the obtained results in detail.

4.3.1. Datasets for time-series change-points detection

The following two time-series sets have been utilized in this change-points detection experiment.

1. ***Jumping mean (artificial data)*** **[33]: single dimensional artificial time-series is generated using the following auto-regressive model**

$$y(t) = 0.6y(t-1) - 0.5y(t-2) + \varepsilon_t, \;\; t = 1, \cdots, 5000, \tag{10}$$

where ε_t denotes the Gaussian noise with mean $\mu_{\tilde{N}}$ and standard deviation 1.5. Similar to that in [33], we set $y(1) = y(2) = 0$ and insert a change-point at every

100th time steps by setting the noise mean $\mu_{\tilde{N}}$ at time t as

$$\mu_{\tilde{N}} = \begin{cases} 0 & \tilde{N} = 1, \\ \mu_{\tilde{N}-1} + \frac{\tilde{N}}{16} & \tilde{N} = 2, \cdots, 49, \end{cases}$$

where $\tilde{N}$ is a number which satisfies $100\left(\tilde{N}-1\right)+1 \leq t \leq 100\tilde{N}$.

2. ***Well log***[4] ***(real data) [35, 39]*****: this dataset consists of $n = 4,050$ univariate measurements (time-series) acquired from a geophysical research. The goal of this study is to detect changes in rock stratification. Each measurement is nuclear magnetic response generated while drilling a well. The original signal is shown in the top panel of Fig. 8. Note that no groundtruth of change-points is available.**

4.3.2. Protocols for time-series change-points detection

Recall that as discussed in Section 4.2.2, the proposed ELM-MI model has three adjustable parameters: the number of hidden nodes N, the regularization factor λ and the ensemble size M. Similar to the settings of Experiment I, we set $\lambda = 10^{-3}$. It was observed from Fig. 5 that using 100 hidden nodes is enough to capture both linear and non-linear relations between random variables for MI estimation. Based on this observation, we set $N = 100$ for this change-points detection experiment. Different from Experiment I, ten ELM-MIs (i.e. $M = 10$) have been averaged to stabilize the estimation result. Besides these three parameters, we have two additional parameters, namely: window size k and time delay τ for time-series data processing. The k controls the number of samples to be captured by a window from time-series data. Similar to [33], $k = 50$ samples have been considered at a time. For the delay, we set $\tau = 1$ in this work. We shall explore the effect of τ on detection accuracy in our future work.

RuLSIF has several parameters, namely: window size k, relativeness weighting parameter α, segment size s (note that the authors in [33] have used n for the segment size) and cross-validation size. To reproduce similar performances as reported in [33], we followed the similar settings: $k = 10$, $s = 50$, $\alpha = 0.1$ and 5-fold cross-validation.

[4] Available from: http://mldata.org/repository/data/viewslug/well-log/

In addition, the following candidate parameters have been used for cross-validation based grid search: $\sigma = \{0.6d_{med}, 0.8d_{med}, d_{med}, 1.2d_{med}, 1.4d_{med}\}$ where d_{med} denotes the median distance and $\lambda = \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ as in [33]. From our preliminary results, it was observed that the symmetrized search does not improve detection accuracy while increasing its computational cost a lot. We thus performed only a forward search for RuLSIF.

Similar to [33, 40], a peak of MI values (change-point scores for RuLSIF) is regarded as a detection alarm if the value/score is higher than a pre-defined threshold η. A detection alarm generated at time t is counted as a correct alarm if $t \in [t^* - 10, t^* + 10]$ where t^* is a time stamp with a true alarm. Otherwise, the alarm is regarded as a false alarm. To remove duplicated alarms, the zth alarm generated at time t_z is eliminated if $t_z - t_{z-1} < 20$ as in [33].

The change-points detection performances of both ELM-MI and RuLSIF methods are evaluated quantitatively and qualitatively. For the quantitative performance, we computed the Receiver Operating Characteristic (ROC) curves and Area Under the ROC Curve (AUC) values. According to [33, 36], the true positive rate and false positive rate for generating an ROC curve are defined as follows:

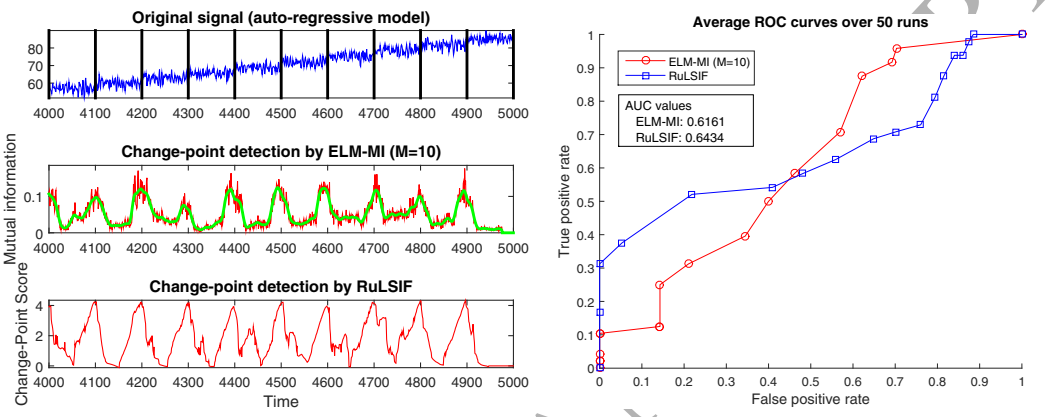
- **True Positive Rate (TPR) =** $n_{correct}/n_{groundTruth}$**,**
- **False Positive Rate (FPR) =** $(n_{all} - n_{correct})/n_{all}$**,**

where n_{all} indicates the number of all alarms generated by the evaluated algorithm, $n_{correct}$ indicates the number of correct alarms among n_{all}, and $n_{groundTruth}$ denotes the entire number change-points embedded in the time-series. Threshold η is decreased from the highest change score or MI value to zero as in [33].

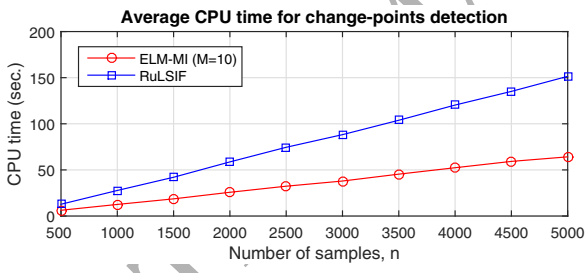
4.3.3. Results and discussions

The upper panel of Fig. 7 (a) shows the input time-series dataset artificially generated using (10). The vertical black lines shown in the panel are those intentionally inserted change points. For clarity, only the last ten change-points are shown in the figure as in [33]. The resulted MI values by ELM-MI and change-point scores by RuLSIF computed from the input signal are shown in the middle and the bottom panel of

Fig. 7 (a), respectively. Due to the random property in parameters setting, the output of ELM-MI model (the red line in the middle panel) is fluctuating over time. For a better visualization of the results, a smoothened line (of green color) is plotted in the plot together.



(a) Qualitative performance (b) Quantitative performance



(c) CPU time performance

Figure 7: (a) Illustrations of the input artificial time-series data (the upper panel), estimated MI values by ELM-MI (the middle panel) and CP scores by RuLSIF (the bottom panel). **(b) shows the average ROC curves over 50 trials. The obtained AUC values for ELM-MI and RuLSIF are 0.6161 and 0.6434, respectively.** The average CPU time performances elapsed for computing the MI values and CP scores from the times-series data of length $n \in \{500, 1000, \cdots, 5000\}$ are shown in (c).

Fig. 7 (b) shows the average ROC curves of ELM-MI and RuLSIF methods, which were taken over 50 trials. In each trial, different random parameters (e.g. kernel centers and widths) were utilized. The average AUC values of the two methods are also shown in the plot. They are finally compared in terms

of their CPU time performance (s) as shown in Fig. 7 (c) which is plotted over $n \in \{500, 1000, \cdots, 5000\}$.

As shown in the middle panel of Fig. 7 (a), ELM-MI produced higher MI values around the change-points while lower values were resulted if no change-point exists. The compared RuLSIF method (see the bottom panel) also produced higher change-point scores (CP-scores) around the change points, while lower CP-scores were resulted when no change exists. From the figure, it is observed that the CP-scores are more intuitive and better correlated to change-points than the MI values. In other words, the CP-scores have no subtle fluctuations, each peak point is sharp enough to indicate a time point and shows clear differences between change- and non-change-points. On the other hand, the output of ELM-MI model needs to be processed by another round to locate candidates for estimated change-points. **However, both methods have produced similar AUC values (0.6161 and 0.6434 respectively for ELM-MI and RuLSIF) as shown in Fig. 7 (b). Moreover,** we emphasize that, as shown in Fig. 7 (c), RuLSIF was about two times slower than the proposed ELM-MI.

Fig. 8 shows the change-points detection performances results on the Well log data. We note here that we report only qualitative performances for this dataset since no groundtruth is available. As shown in the figure, those time stamps corresponding to obvious changes were correctly detected by the two methods. It is observed from the figure that, however, RuLSIF could not detect those changes of a pulse shape (denoted as Sharp peaks 1 to 4 in the top panel of Fig. 8) except for Sharp peak 3. Unlike RuLSIF, the proposed framework yielded a clear indication of plausible changes for these sharp peaks. This result leads us to a partial observation that ELM-MI is more appropriate for an anomaly detection application (with a pulse shape change) than RuLSIF.

5. An empirical analysis in changes of ECG signal while driving

In this section, the proposed ELM-MI model is applied to an analysis of changes in heart rate patterns while driving a simulation car. For the heart rate patterns, an in-house electrocardiogram (ECG) time-series dataset was collected. Similar experi-

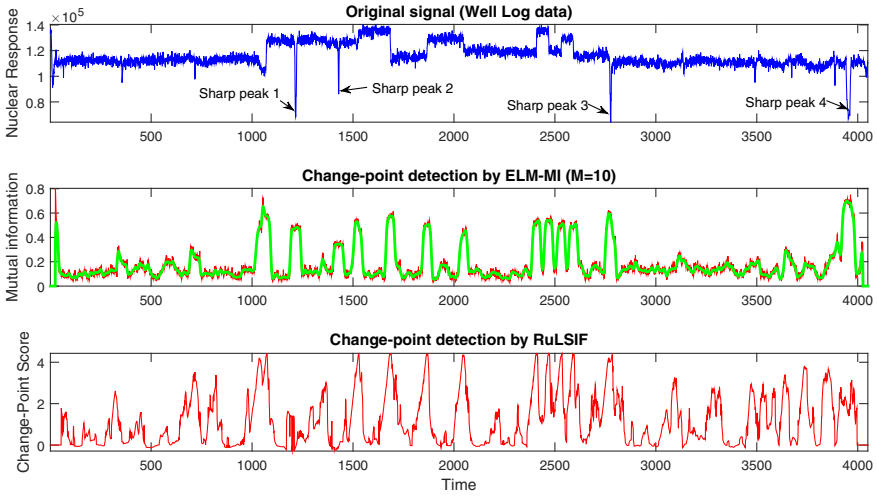


Figure 8: Illustrations of the Well log time-series data (the upper panel), the estimated MI values by ELM-MI (the middle panel) and CP scores by RuLSIF (the bottom panel).

mental protocols discussed in Section 4.3.2 **(e.g.** $\lambda = 10^{-3}$**,** $N = 100$**,** $M = 10$**,** $\tau = 1$ **and** $k = 50$ **for the proposed framework;** $k = 10$**,** $s = 50$**,** $\alpha = 0.1$**, 5-fold cross-validation for** α **and** λ **selection for RuLSIF)** have been adopted for this analysis study. The following subsections present the data collection process and the obtained results.

5.1. (In-house) ECG time-series data

ECG time-series data were collected from two subjects while they were driving a simulation car. The main goal of this data collection was to detect changes in human stress level while driving. As an external source to raise the stress level, an additional workload (e.g. N-back task [41]) was assigned to the driver while driving. The following provides details of the experimental configuration, simulation system setup and preprocessing.

1. *Subjects*: Two healthy male subjects (of age 26 and 36) participated in this data collection. They have more than one year's driving experience with a valid driv-

ing licence. Thus, it was assumed that they were aware of the traffic signs and regulations.

2. *System setup*: Fig. 9 shows (a) an overview of the deployed driving simulator and (b) controller (Logitech G27 racing wheel). A commercial driving simulation software (City Car Driving [42]) was utilized in this work. For ECG signals measurement, we used B20 Patient Monitor of GE Healthcare [43]. This model calculates ECG signal from III-leads electrodes attached on chest at 300 Hz sampling rate. The measured ECG signal is immediately stored on a wiry connected computer. Before the main session of data collection, the participants were given about 40 minutes for driving practice so that they could be familarized to the simulator.

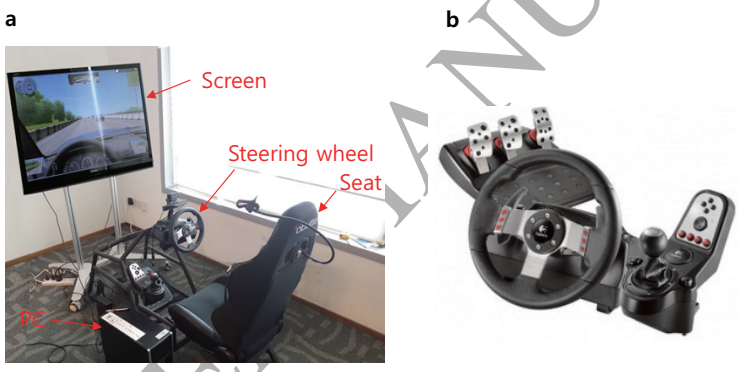


Figure 9: An illustration of driving simulator system: (a) an overview and (b) the utilized controller.

3. *Mental stress (additional workload)*: As an external source of stress, an N-back task [41] was performed while driving. By considering difficulties of the task and connecting it with driving, three 1-back tasks of one minute at driving speed 60km/h, 80km/h and 100km/h were performed.
4. *ECG signal preprocessing and RR intervals*: The stored signal was then fed to the following preprocessing steps for signal quality enhancement and feature extraction:
   (a) Bandpass filter with a passing range [6 Hz, 300 Hz],
   (b) Signal smoothing by Savitzky-Golay finite impulse response filter [44],

(c) RR intervals extraction from the preprocessed ECG signal as features.

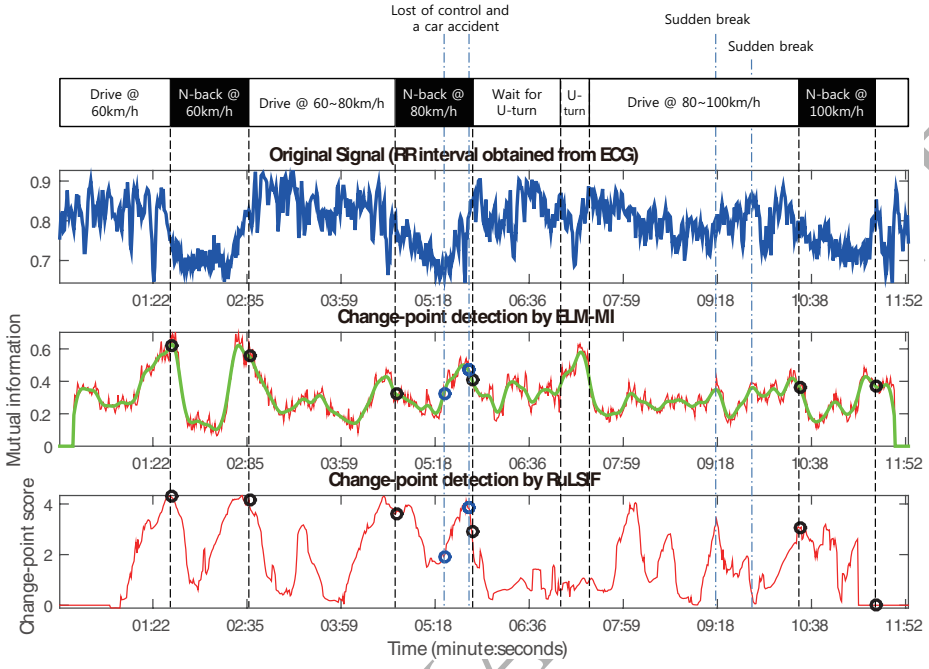
5. *Cash rewards*: In order to encourage the participants to do their best in the driving simulation, different amount of cash (5$∼15$ Singapore Dollar) was rewarded to the participants based on their performance (e.g. no accident, maintain the speed required, good score in N-back task etc.).

5.2. Results and discussions

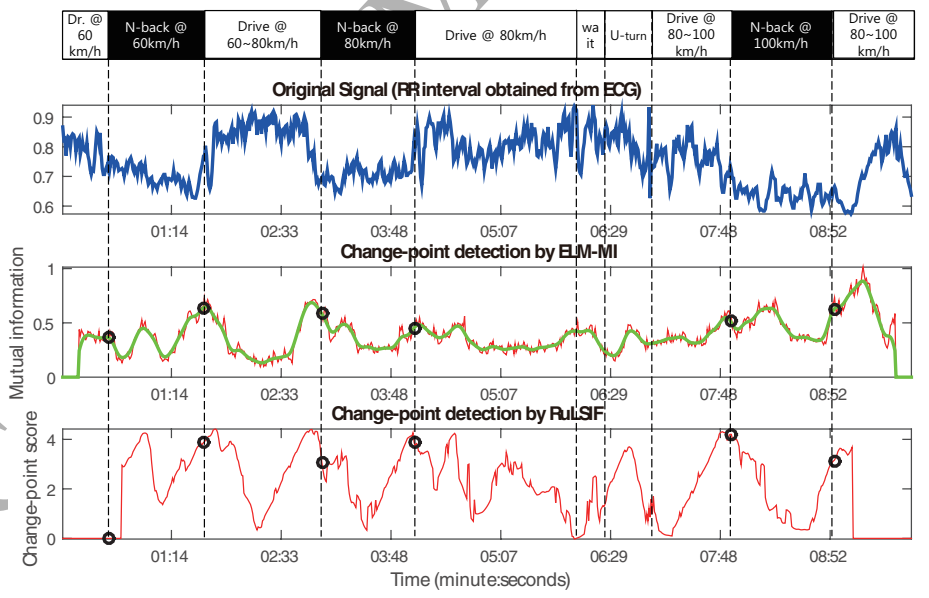
The qualitative results of change-points detection obtained using the real ECG (RR interval) data are shown in Fig. 10. Particularly, Fig. 10 (a) and (b) respectively shows each subject's driving log (manually tagged), the input RR interval signals, resulted MI values and CP-scores. As for the driving log, we only recorded a range of car speed, time points for the N-back tasks and incidents on the roads.

As shown in Fig. 10 (a), the first subject drove a simulation car for about twelve minutes. During his driving, he had a small car accident (e.g. crashed a preceding car) and made sudden break twice because of other cars on the road. The second participant (see Fig. 10 (b)), drove a car for about ten minutes without such incident. From the driving log and RR interval signals, it is observed that the N-back task is enough to cause changes in heart rate patterns. The incidents on the road are also clearly related to the heart rate patterns.

Similar to the results of artificial data (see Fig. 7), CP-scores provide clearer indication of candidate change-points than that of the MI values, while most of the peak points of the MI values are also well correlated to those incidents. Except for some time points, both methods indicated similar time points as possible candidates for change-points. Besides the detection accuracy issue, we emphasize here that ELM-MI is about four times faster than RuLSIF. For example, ELM-MI consumed 5.24 seconds and 5.71 seconds for processing the first and second subjects, while RuLSIF needed 23.74 and 24.17 seconds, respectively. These CPU times are averages over ten repetitions.



(a) Subject 1



(b) Subject 2

Figure 10: Illustrations of the driving log, ECG (RR interval) signals, the estimated MI values by ELM-MI and CP scores by RuLSIF. (a) and (b) respectively shows the data of subject 1 and 2.

6. Conclusion

In this paper, an extreme learning machine-based MI estimator was proposed. Essentially, the density ratio calculation which is the most computationally expensive step in MI estimation was approximated by the proposed learning network. By randomizing the kernel centers and widths, the proposed model could get rid of the tedious parameters selection process. Our empirical results on four artificial datasets showed that the proposed model produced better or comparable estimation error performances than that of the competing MI estimators. However, our model significantly outperformed the compared estimators in terms of computational efficiency. **In applications of time-series change-points detection and driving stress analysis, the proposed model yielded accuracy performance comparable to that of a state-of-the-art method, but with less computational time.**

In our future work, an extension of the proposed change-points detection framework into an online mode would be beneficial in detecting driving stress in real-time.

Acknowledgments

B.-S. Oh' research was supported by the Singapore Academic Research Fund (AcRF) Tier 1 under Project RG 185/14.

References

[1] Y. Kopylova, D. Buell, C.-T. Huang, J. Janies, Mutual Information Applied to Anomaly Detection, Journal of Communications and Networks 10 (1) (2008) 89–97.

[2] T. Drugman, Using Mutual Information in Supervised Temporal Event Detection: Application to Cough detection, Biomedical Signal Processing and Control 10 (2014) 50–57.

[3] M. B. Stojanović, M. M. Božić, M. M. Stanković, Z. P. Stajić, A Methodology for Training Set Instance Selection using Mutual Information in time Series Prediction, Neurocomputing 141 (2014) 236–245.

[4] H. Peng, F. Long, C. Ding, Feature Selection based on Mutual Information Criteria of Max-dependency, Max-relevance, and Min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[5] W. Qian, W. Shu, Mutual Information Criterion for Feature Selection from Incomplete Data, Neurocomputing 168 (2015) 210–220.

[6] A. M. Fraser, H. L. Swinney, Independent coordinates for strange attractors from mutual information, Physical review A 33 (2) (1986) 1134–1140.

[7] W. Endo, F. P. Santos, D. Simpson, C. D. Maciel, P. L. Newland, Delayed mutual information infers patterns of synaptic connectivity in a proprioceptive neural network, Journal of computational neuroscience 38 (2) (2015) 427–438.

[8] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical review E 69 (6) (2004) 066138.

[9] B. W. Silverman, Density Estimation for Statistics and Data Analysis, Vol. 26, CRC press, 1986.

[10] E. Parzen, On Estimation of a Probability Density Function and Mode, The annals of mathematical statistics (1962) 1065–1076.

[11] T. Suzuki, M. Sugiyama, J. Sese, T. Kanamori, A Least-squares Approach to Mutual Information Estimation with Application in Variable Selection, in: Proceedings of Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery, 2008.

[12] T. Sakai, M. Sugiyama, Computationally Efficient Estimation of Squared-Loss Mutual Information with Multiplicative Kernel Models, IEICE TRANSACTIONS on Information and Systems 97 (4) (2014) 968–971.

[13] T. Suzuki, M. Sugiyama, J. Sese, T. Kanamori, Approximating Mutual Information by Maximum Likelihood Density Ratio Estimation, 2008, pp. 5–20.

[14] M. Sugiyama, T. Suzuki, T. Kanamori, Density ratio estimation in machine learning, Cambridge University Press, 2012.

[15] D. S. Broomhead, D. Lowe, Multivariable Functional Interpolation and Adaptive Networks, Complex Systems 2 (1988) 321–355.

[16] G.-B. Huang, C.-K. Siew, Extreme Learning Machine with Randomly Assigned RBF Kernels, International Journal of Information Technology 11 (1) (2005) 16–24.

[17] H. J. Baek, H. B. Lee, J. S. Kim, J. M. Choi, K. K. Kim, K. S. Park, Nonintrusive Biological Signal Monitoring in a Car to Evaluate a Drivers Stress and Health State, Telemedicine and e-Health 15 (2) (2009) 182–189.

[18] G. Rigas, Y. Goletsis, D. I. Fotiadis, Real-time Driver's Stress Event Detection, IEEE Transactions on Intelligent Transportation Systems 13 (1) (2012) 221–234.

[19] M. Owis, A. H. Abou-Zied, A.-B. M. Youssef, Y. M. Kadah, Study of Features based on Nonlinear Dynamical Modeling in ECG Arrhythmia Detection and Classification, IEEE Transactions on Biomedical Engineering 49 (7) (2002) 733–736.

[20] M. Yang, H. Zhang, H. Zheng, H. Wang, Q. Lin, Mutual Information-based Approach to the Analysis of Dynamic Electrocardiograms, Technology and health care 16 (5) (2008) 367–375.

[21] T. M. Cover, J. A. Thomas, Elements of Information Theory, John Wiley & Sons, 2012.

[22] T. Suzuki, M. Sugiyama, T. Kanamori, J. Sese, Mutual information estimation reveals global associations between stimuli and biological processes, BMC bioinformatics 10 (1) (2009) S52.

[23] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, M. Kawanabe, Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, in: Advances in neural information processing systems, 2008, pp. 1433–1440.

[24] X. Nguyen, M. J. Wainwright, M. Jordan, Estimating divergence functionals and the likelihood ratio by convex risk minimization, IEEE Transactions on Information Theory 56 (11) (2010) 5847–5861.

[25] T. Kanamori, T. Suzuki, M. Sugiyama, Statistical Analysis of Kernel-Based Least-Squares Density-Ratio Estimation, Machine Learning 86 (3) (2012) 335–367.

[26] W. Liu, P. P. Pokharel, J. C. Príncipe, The Kernel Least-Mean-Square Algorithm, IEEE Transactions on Signal Processing 56 (2) (2008) 543–554.

[27] B. Chen, S. Zhao, P. Zhu, J. C. Príncipe, Mean Square Convergence Analysis for Kernel Least Mean Square Algorithm, Signal Processing 92 (11) (2012) 2624–2632.

[28] B. Chen, S. Zhao, P. Zhu, J. C. Príncipe, Quantized Kernel Least Mean Square Algorithm, Neural Networks and Learning Systems, IEEE Transactions on 23 (1) (2012) 22–32.

[29] B. Chen, S. Zhao, P. Zhu, J. C. Príncipe, Quantized Kernel Recursive Least Squares Algorithm, IEEE Transactions on Neural Networks and Learning Systems 24 (9) (2013) 1484–1491.

[30] J. Cao, Z. Lin, G.-B. Huang, N. Liu, Voting based extreme learning machine, Information Sciences 185 (1) (2012) 66–77.

[31] G.-B. Huang, L. Chen, C.-K. Siew, Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes, IEEE Transactions on Neural Networks 17 (4) (2006) 879–892.

[32] M. Paluš, V. Komárek, Z. Hrnčíř, K. Štěrbová, Synchronization as Adjustment of Information Rates: Detection from Bivariate Time Series, Physical Review E 63 (4) (2001) 046211.

[33] S. Liu, M. Yamada, N. Collier, M. Sugiyama, Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation, Neural Networks 43 (2013) 72–83.

[34] W. K. Härdle, M. Müller, S. Sperlich, A. Werwatz, Nonparametric and Semiparametric Models, Springer Science & Business Media, 2012.

[35] R. P. Adams, D. J. C. MacKay, Bayesian online changepoint detection, arXiv preprint arXiv:0710.3742.

[36] Y. Kawahara, M. Sugiyama, Sequential Change-Point Detection Based on Direct Density-Ratio Estimation, Statistical Analysis and Data Mining: The ASA Data Science Journal 5 (2) (2012) 114–127.

[37] The MathWorks, MATLAB, `http://www.mathworks.com/`, [Online; accessed 22-October-2015].

[38] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1-3) (2006) 489–501.

[39] J. J. K. O. Ruanaidh, W. J. Fitzgerald, Numerical Bayesian Methods Applied to Signal Processing, Springer Science & Business Media, 2012.

[40] F. Desobry, M. Davy, C. Doncarli, An Online Kernel Change Detection Algorithm, IEEE Transactions on Signal Processing 53 (8) (2005) 2961–2974.

[41] L. R. Zeitlin, Subsidiary Task Measures of Driver Mental Workload: A Long-Term Field Study, Transportation Research Record 1403 (1993) 23–27.

[42] F. Development, City car driving, `http://citycardriving.com/`, [Online; accessed 22-October-2015].

[43] G. Healthcare, B20 patient monitor, http://www3.gehealthcare.co.uk/en-gb/products/categories/patient_monitoring/patient_monitors/b20-patient-monitor, [Online; accessed 22-October-2015].

[44] A. Savitzky, M. J. E. Golay, Smoothing and Differentiation of Data by Simplified Least Squares Procedures, Analytical chemistry 36 (8) (1964) 1627–1639.

Biography

Beom-Seok Oh received the B.S. degree in Computer Science from KonKuk University, South Korea, in 2008, the M.S. degree in Biometrics and the Ph.D degree in Electrical and Electronic Engineering from Yonsei University, South Korea, in 2010 and August 2015, respectively. Since April 2015, he is a research associate in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include biometric, pattern recognition, and machine learning.

Lei Sun received his Ph.D. degree in Electrical and Electronics Engineering from Beijing Institute of Technology, China, in 2007. Since 2007, he has been with the faculty of School of Information and Electronics at Beijing Institute of Technology, where he is currently an Associate Professor. He is the recipient of several awards including Excellent Doctoral Graduate Student of Beijing Institute of Technology in 2007, Science and Technology Progress Award from China Ministry of Industry and Information Technology, China in 2002, 2003, and 2009. He was a Visiting Courtesy professor at Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, in 2011-2012. He was a Senior Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore during March-November, 2015. Currently, his research interests are in statistical machine learning with emphasis on pattern recognition, statistical inference, and their connections to information theory.

Chung Soo Ahn received the B.S. degree in Industrial Engineering from Ajou University, South Korea, in 2014. Since August 2014, he is Ph.D student in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interest include driver state recognition, pattern recognition, and machine learning.

Yong Kiang Yeo is currently a researcher in the School of Electrical and Electronic Engineering at Nanyang Technological University of Singapore working on biomedical wearable technologies. His research interests include computer vision, image processing, machine learning, biomedical signal processing and practical implementation of algorithms.

Yan Yang received the Ph. D. degree in University of Southampton, UK, in 2011, followed by postdoctoral research at Massachusetts Institute of Technology (MIT), USA in 2012 to 2013. She is now working as a professor in Northwestern Polytechical University, China, after the research experience as a Senior Scientist in Nanyang Technological University, Singapore. Prof. Yang's research interests include the transportation engineering, human factor, data mining and machine learning. Yan

is also affiliated with Southeast University, China as an Associate Professor.

Nan Liu received the BEng degree in electrical engineering from University of Science and Technology Beijing, China, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore. He is currently a Principal Research Scientist at the Department of Emergency Medicine, Singapore General Hospital and an Adjunct Assistant Professor at the Centre for Quantitative Medicine, Duke-NUS Graduate Medical School. His research interests include medical informatics, healthcare data analytics, machine learning, statistical analysis, data mining, biomedical signal processing, natural language processing and image processing.

Zhiping Lin received the B.Eng. degree in control engineering from South China Institute of Technology, Canton, China in 1982 and the Ph.D. degree in information engineering from the University of Cambridge, England in 1987. Since Feb. 1999, he has been an Associate Professor at Nanyang Technological University (NTU), Singapore. He is also the Program Director in Distributed Diagnosis at the Valens Centre of Excellence, NTU. Dr. Lin has been in the editorial board of Multidimensional Systems and Signal Processing since 1993 and has served as its Editor-in-Chief since 2011. He was an Associate Editor of Circuits, Systems and Signal Processing for 2000-2007 and IEEE Transactions on Circuits and Systems, Part II, for 2010-2011. He also serves as a reviewer for Mathematical Reviews. His research interests include multidimensional systems and signal processing, statistical and biomedical signal processing, and machine learning. He is a co-author of the 2007 Young Author Best Paper Award from the IEEE Signal Processing Society, Distinguished Lecturer of the IEEE Circuits

and Systems Society for 2007-2008, and the Chair of the IEEE Circuits and Systems Singapore Chapter for 2007-2008.