

# Hubness Change Point Detection

Ikumi Suzuki<sup>1</sup>, Kazuo Hara<sup>1</sup>, Eiji Murakami<sup>2</sup>

<sup>1</sup>Yamagata University

<sup>2</sup>Azbil Kimmon Co.,Ltd., Keio University

suzuki.ikumi@gmail.com, kazuo.hara@gmail.com, e.murakami.y8@azbil.com

## Abstract

This study proposes a new change detection method that leverages hubness. Hubness is a phenomenon that occurs in high-dimensional spaces, where certain special data points, known as hub data, tend to be closer to other data points. Hubness is known to degrade the accuracy of methods based on nearest neighbor search. Therefore, many studies in the past have focused on reducing hubness to improve accuracy. In contrast, this study utilizes hubness to detect changes. Specifically, if there is no change, suppressing the hubness occurring in the two datasets obtained by dividing the time-series data will result in a uniform data distribution. However, if there is a change, even if we try to reduce the hubness in the two datasets obtained by dividing the time-series data before and after the change, the hubness will not be reduced, and the data distribution will not become uniform. We use this finding to detect changes. Experiments with synthetic data show that the proposed method achieves accuracy comparable to or exceeding that of existing methods. Additionally, the proposed method achieves good accuracy with real-world data from hydraulic systems and gas sensors, along with excellent runtime performance.

## Introduction

This study proposes a novel change detection method. The changes targeted for detection in this study are those that occur in time-series data obtained by monitoring the state of machines. Examples of time-series data include data obtained from sensors monitoring the state of machines operating in factories (Yi, Huang, and Li 2017; Lei et al. 2018), and data obtained from meters measuring the usage of water or gas in households (Gungor et al. 2011). These data typically change due to intentional state changes, such as switching operating modes. However, they can also change due to unintentional state changes caused by machine failures. Missing these unintentional state changes can threaten the safety and reliability of the machines. Furthermore, it is desirable to perform change detection through edge computing on-site, without sending data to a cloud data center, and to trigger alarms as quickly as possible when changes occur. Therefore, developing a change detection algorithm that does not require extensive computation time is a crucial task.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Existing change detection methods can be broadly classified into two categories: those that assume probabilistic models and those that are model-free. Methods based on probabilistic models first estimate a probabilistic model that generates data using past data. When data that is significantly unlikely according to the estimated probabilistic model appears consecutively, a change is considered to have occurred. Model-free methods, on the other hand, involve dividing time-series data into two groups before and after each point in time to determine whether a change has occurred. If a significant difference between the two groups is observed, a change is deemed to have occurred at that point.

The new change detection method we propose is categorized as a model-free method. We focus on the hubness phenomenon that occurs in the two datasets obtained by dividing the time-series data. When using entries from one dataset as queries to perform nearest neighbor search on the entries from the other dataset, certain entries from the other dataset frequently appear in the search results for many queries, indicating the occurrence of the hubness phenomenon. To manage this hubness, we apply hubness reduction methods, specifically centering and vector length normalization. Importantly, if no changes occur and, as a result, there is no difference between the two datasets, hubness reduction will succeed, resulting in a uniform data distribution. Conversely, if changes occur and there is a difference between the two datasets, hubness will not be reduced, and the data distribution will not become uniform.

Our contributions are as follows.

- Propose a novel method that applies the concept of hubness to change detection.
- Introduce a novel method for hubness suppression in detecting differences between two datasets and provide analytical proofs to support its effectiveness.
- Propose employing a uniformity test to measure hubness phenomena between two datasets, which advances effective change point detection.
- Develop the Hubness Change Point Detection (Hub-CPD) algorithm and present comparative experiments against state-of-the-art approaches. Due to the absence of complex computations, our method achieves faster computation times and accuracy that is either comparable to or superior to existing methods.

## Related Work

There is a large amount of research on change detection, as seen in the literature such as (Aminikhanghahi and Cook 2017). The objectives of past research on change detection can be divided into two main types: segmentation of time-series data, which involves dividing the given time-series data into contiguous intervals with no changes; and detecting changes in time-series data, which involves sounding an alarm when a change occurs in the incoming data. This study falls under the latter type, and therefore, we will describe it in detail below. Additionally, there is an intermediate type of research that aims to predict the distribution of the number of time points with no changes since the most recent change, known as the run length distribution, but these studies do not aim to sound an alarm (Adams and MacKay 2007).

A traditional method for sounding an alarm in response to changes in time-series data is the CUSUM control chart (Page 1954). This method assumes a probabilistic model, that is, Gaussian distribution, for data generation under normal conditions, and it triggers an alarm when the cumulative sum of scores representing deviations from the normal state at each point exceeds a threshold. Following the CUSUM, change detection methods that assume data generation models have been developed, including autoregressive models (Yamanishi and Takeuchi 2002), state-space models (Kawahara, Yairi, and Machida 2007), and spatio-temporal models (Alanqary, Alomar, and Shah 2021). In contrast, model-free change detection methods that do not require assumptions about the data generation model have been developed. These methods divide the recent time-series data into two subsequences and trigger an alarm when a difference between them is detected, indicating a change has occurred. Methods for identifying differences between the two subsequences include those based on density ratio (Liu et al. 2013), maximum mean discrepancy (MMD) (Chang et al. 2019), and random partitioning (Hooi and Faloutsos 2019).

The proposed method in this study is a model-free change detection approach that leverages hubness to identify differences between two subsequences. It should be noted that the method based on density ratio requires matrix inversions, and the method based on MMD uses neural networks, both of which result in high computational costs for change detection.

### Hubness Suppression in Single Dataset

Hubness is the phenomenon where special data points, those that are particularly close to many other data points, emerge in a dataset (Radovanović, Nanopoulos, and Ivanović 2010a,b; Radovanović 2015). Formally, let  $D$  be dataset consisting of  $d$ -dimensional vectors and  $N_k(x)$  be the count of how many times  $x \in D$  is in the  $k$ -nearest neighbors of  $q \in D \setminus \{x\}$ . The presence of a right-skewed tail in the distribution of  $N_k(x)$  is referred to as the occurrence of hubness. The data points with large values of  $N_k(x)$  are termed hub data.

Hubness is known to occur when there is a gradient in data density, and it becomes particularly prominent in high-dimensional spaces (Low et al. 2013; Radovanović 2015;

Hara et al. 2016). When measuring the proximity between data points using (squared) Euclidean distance  $dist(x, q) = \|x - q\|^2$ , hub data tends to appear in areas where the data density is higher compared to the surrounding regions. For instance, in a dataset generated from a Gaussian distribution, data close to the center of the dataset becomes hub data. In a dataset with multiple clusters, data close to the cluster centers becomes hub data.

When hubness occurs, it is known to degrade the accuracy of tasks such as classification and recommendation based on nearest neighbor search. To address this issue, methods for suppressing hubness have been developed (Schnitzer et al. 2012; Suzuki et al. 2012, 2013; Hara et al. 2015a,b, 2016; Flexer 2016). The centering methods modify the metric for measuring proximity between data points to prevent the emergence of hub data. This modification eliminates areas where data density is higher compared to the surroundings, leading to a more uniform data density. According to (Hara et al. 2016), the formal definition of centering squared distance is given below.

**Definition 1** (Centering the squared distance). *Given  $x, q \in D$  and  $c = \frac{1}{|D|} \sum_{x \in D} x$  is the centroid of dataset  $D$ . The dissimilarity between data points  $x$  and  $q$  is defined as,*

$$dissim(x, q) = \|x - q\|^2 - \|x - c\|^2 - \|q - c\|^2. \quad (1)$$

Using Eq. (1) to measure the proximity between data points, the data density is modified to approach more uniformly, suppressing hubness (Hara et al. 2016). In the next section, we propose how to suppress hubs between two datasets.

### Hubness Suppression Between Two Datasets

Techniques for suppressing hubness in single dataset already exist, but extending them to address hubness arising between two datasets has not been explored. In this section, we introduce the concept of hubness that arises between two distinct datasets consisting of  $d$ -dimensional vectors, and we propose a method to suppress it. The algorithm of change point detection we will propose in the next section leverages the approach described in this section.

Previous studies on hubness assumed that the database sample  $x$  and the query sample  $q$  belong to the same dataset  $D$ . However, here, we introduce a new scenario where  $x$  belongs to the dataset  $D_x$  and  $q$  belongs to the dataset  $D_q$ , assuming that these are two distinct datasets. Under this scenario, let  $kNN(q)$  denote the set of  $x \in D_x$  that fall within the  $k$ -nearest neighbors of  $q \in D_q$ . We define  $\mathbb{1}(x \in kNN(q)) = 1$  if  $x$  is within the  $k$ -nearest neighbors of  $q$ , and  $\mathbb{1}(x \in kNN(q)) = 0$  otherwise. In this context, let  $N_k(x) = \sum_{q \in D_q} \mathbb{1}(x \in kNN(q))$ . We refer to the phenomenon where  $N_k(x)$  becomes particularly large for some data  $x$  as hubness, and such data points are called hub data.

When hubness occurs, hub data  $x$  is more likely to be included in the  $k$ -nearest neighbors of data  $q$ . In other words, on average, the dissimilarity between data  $x$  and data  $q$  is small, or the similarity is large. Therefore, hubness is considered to be suppressed when, on average, the dissimilarity

or similarity between any two data  $a, b \in D_x$  and  $q \in D_q$  becomes equal.

Next, we define the following metric for measuring the proximity between data points to suppress hubness:

**Definition 2** (Centering the squared distance for hubness suppression between two datasets). *Given  $x \in D_x$ ,  $q \in D_q$  and  $c_x = \frac{1}{|D_x|} \sum_{x \in D_x} x$  and  $c_q = \frac{1}{|D_q|} \sum_{q \in D_q} q$  are the centroids of datasets  $D_x$  and  $D_q$ , respectively. We define the dissimilarity between data points  $x$  and  $q$  as,*

$$\text{dissim}(x, q) = \|x - q\|^2 - \|x - c_x\|^2 - \|q - c_q\|^2 + \|c_x - c_q\|^2. \quad (2)$$

Note that if  $D_x$  and  $D_q$  are the same dataset, then  $c_x = c_q$ , and hence, Eq. (2) coincides with Eq. (1).

Importantly, Eq. (2) is simplified through a straightforward mathematical transformation to:

$$\begin{aligned} \text{dissim}(x, q) &= \langle x - q, x - q \rangle - \langle x - c_x, x - c_x \rangle \\ &\quad - \langle q - c_q, q - c_q \rangle + \langle c_x - c_q, c_x - c_q \rangle \\ &= -2\langle x, q \rangle + 2\langle x, c_x \rangle + 2\langle q, c_q \rangle - 2\langle c_x, c_q \rangle \\ &= -2\langle x - c_q, q - c_x \rangle. \end{aligned}$$

Furthermore, by reversing the sign and dividing by 2, we convert it into the similarity measure as shown in the following definition :

**Definition 3** (Centering the inner product similarity for hubness suppression between two datasets). *Given  $x \in D_x$ ,  $q \in D_q$  and  $c_x = \frac{1}{|D_x|} \sum_{x \in D_x} x$  and  $c_q = \frac{1}{|D_q|} \sum_{q \in D_q} q$  are the centroids of datasets  $D_x$  and  $D_q$ . We define the similarity between data points  $x$  and  $q$  as,*

$$\text{sim}(x, q) = \langle x - c_q, q - c_x \rangle. \quad (3)$$

Eq.(3) calculates the similarity between  $x$  and  $q$  by transforming them into the difference vectors with respect to the centroids of the datasets  $D_q$  and  $D_x$ , to which they do not belong, and then computing the inner product of these difference vectors. We show in Proposition 1 that using the similarity in Eq. (3) suppresses hubness. That is, hubness is suppressed when the differences in similarity between any two data points  $a, b \in D_x$  and  $q \in D_q$ , become equal on average.

**Proposition 1.** *Applying Eq.(3), if  $D_x$  and  $D_q$  are datasets generated from the same distribution, hubness is suppressed. Conversely, if  $D_x$  and  $D_q$  are datasets generated from different distributions, there is no guarantee that hubness will be suppressed.*

*Proof.* Let  $x \in D_x$  and  $q \in D_q$  be generated according to the probability distributions  $P(x)$  and  $P(q)$ , respectively. We denote the means of  $P(x)$  and  $P(q)$  by  $\mu_x$  and  $\mu_q$ , and the sample means of the datasets  $D_x$  and  $D_q$  by  $c_x$  and  $c_q$ , respectively.

For two distinct data points  $a, b \in D_x$ , when the similarity with  $q \in D_q$  is calculated using the standard inner product instead of the similarity defined in Eq. (3), the difference  $\Delta$  between them is expressed by the following formula:

$$\Delta = \langle a, q \rangle - \langle b, q \rangle = \langle a - b, q \rangle.$$

The difference  $\Delta$  is a function of  $q$ . Since  $q$  is generated according to the probability distribution  $P(q)$ , the expectation of the difference  $\Delta$  is calculated as follows:

$$\mathbb{E}_q[\Delta] = \mathbb{E}_q[\langle a - b, q \rangle] = \langle a - b, \mu_q \rangle.$$

From this equation, it shows that if  $\mu_q$  is not the zero vector, the difference  $\Delta$  is, on average, not zero. In other words, this implies that either  $a$  or  $b$  could become a hub data, as the similarity with  $q \in D_q$  tends to be larger on average.

Now, we will show that when using the similarity defined in Eq. (3) instead of the standard inner product, hubness is suppressed if  $D_x$  and  $D_q$  are generated according to the same probability distribution. First, let  $\Delta^{\text{cent}}$  denote the difference in similarity between  $q \in D_q$  and  $a, b \in D_x$  when computed using Eq. (3). Then,  $\Delta^{\text{cent}}$  is given by the following equation:

$$\begin{aligned} \Delta^{\text{cent}} &= \langle a - c_q, q - c_x \rangle - \langle b - c_q, q - c_x \rangle \\ &= \langle a - b, q - c_x \rangle = \langle a - b, q \rangle - \langle a - b, c_x \rangle \\ &= \Delta - \langle a - b, c_x \rangle. \end{aligned}$$

Then, the expectation of  $\Delta^{\text{cent}}$  is given by:

$$\begin{aligned} \mathbb{E}_q[\Delta^{\text{cent}}] &= \mathbb{E}_q[\Delta] - \langle a - b, c_x \rangle \\ &= \langle a - b, \mu_q \rangle - \langle a - b, c_x \rangle = \langle a - b, \mu_q - c_x \rangle. \end{aligned}$$

If the datasets  $D_x$  and  $D_q$  are generated from the same distribution, then the sample mean  $c_x$  of  $D_x$  and the mean  $\mu_q$  of  $P(q)$  become approximately equal. Therefore, we obtain the following:

$$\mathbb{E}_q[\Delta^{\text{cent}}] = \langle a - b, \mu_q - c_x \rangle \approx 0.$$

From this equation, the difference  $\Delta^{\text{cent}}$  is approximately equal to zero on average. In other words, using the similarity defined in Eq. (3), the similarity between  $q \in D_q$  and the two data points  $a, b \in D_x$  becomes approximately equal on average, indicating that hubness is suppressed.

Conversely, if the datasets  $D_x$  and  $D_q$  are generated from different distributions,  $c_x$  and  $\mu_q$  are not necessarily equal. As a result, the difference  $\Delta^{\text{cent}}$  is not guaranteed to be zero on average. This implies that hubness is not necessarily suppressed.  $\square$

## Uniformity Test for Measuring Hubness

In this section, we propose the uniformity test for measuring hubness. In previous studies (Radovanović, Nanopoulos, and Ivanović 2010a), the occurrence of hubness was measured by the skewness of the distribution of the frequency with which a data point falls into the  $k$ -nearest neighbors of other data points (denoted as  $N_k(x)$ ). Smaller skewness indicates that hubness is suppressed. However, skewness, as a third-order moment, tends to be unstable.

In the previous section, we have shown that using the similarity measure defined in Eq. (3) can reduce hubness. Here, we further mitigate hubness by normalizing the vector lengths to 1. It is known that when using inner product similarity, hub data tends to appear at the boundary of the dataset, rather than at the center, in the hubness that arises

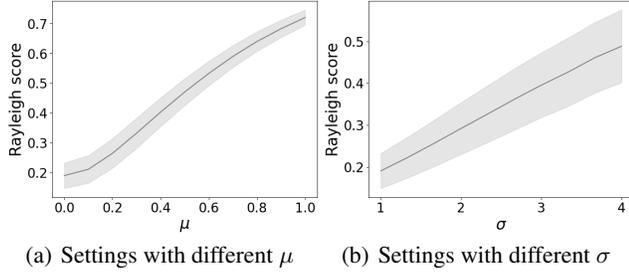


Figure 1: Rayleigh score obtained by repeating the generation of  $D_x$  and  $D_q$  10,000 times for each setting. The gray shading represents the range of the mean  $\pm$  standard deviation.

within a single dataset (Low et al. 2013; Suzuki and Hara 2017). Here, the boundary refers to the edge of the region supported by the data distribution. For example, in datasets generated from Gaussian or t-distribution, data that is notably distant from the center of the dataset and is essentially considered outliers becomes hub data. In contrast, hubness does not occur when a dataset does not have a boundary and density gradient. The situation without a boundary and density gradient is obtained when the dataset is distributed on a unit hyper-sphere. Therefore, we propose to normalize the length of the difference vectors,  $x - c_q$  and  $q - c_x$  to 1 in Eq. (3), as shown below:

**Definition 4** (Normalizing the centered inner product similarity for hubness suppression between two datasets). *Given  $x \in D_x$ ,  $q \in D_q$  and  $c_x = \frac{1}{|D_x|} \sum_{x \in D_x} x$  and  $c_q = \frac{1}{|D_q|} \sum_{q \in D_q} q$  are the centroids of datasets  $D_x$  and  $D_q$ . We define the similarity between data points  $x$  and  $q$  as,*

$$\text{sim}(x, q) = \left\langle \frac{(x - c_q)}{\|x - c_q\|}, \frac{(q - c_x)}{\|q - c_x\|} \right\rangle. \quad (4)$$

Using Eq. (4), if  $D_x$  and  $D_q$  are datasets generated from the same distribution, hubness is suppressed. For instance, if  $D_x$  and  $D_q$  follow the same isotropic Gaussian distribution,  $(x - c_q)/\|x - c_q\|$  becomes a uniform distribution on a unit hyper-sphere without density gradient. Similarly,  $(q - c_x)/\|q - c_x\|$  also becomes a uniform distribution on a unit hyper-sphere. Therefore, in this case, hubness is suppressed. In contrast, if  $D_x$  and  $D_q$  are datasets generated from different distributions, it is not assured that  $(x - c_q)/\|x - c_q\|$  and  $(q - c_x)/\|q - c_x\|$  become uniform distributions on a unit hyper-sphere, and thus hubness is not suppressed.

The above mentioned relation between hubness suppression and uniformity can be observed through the following experiment. Consider the case where  $x \in D_x$  ( $|D_x| = 50$ ) is generated from a 10-dimensional Gaussian distribution  $N(0, I)$ , and  $q \in D_q$  ( $|D_q| = 50$ ) is generated from  $N(\mu \mathbf{1}, I)$ . Here,  $I$  is the  $10 \times 10$  identity matrix, and  $\mathbf{1}$  is a 10-dimensional all-ones vector. In this case, we investigate whether  $(x - c_q)/\|x - c_q\|$  is uniformly distributed on a unit hyper-sphere using the score calculated by

Rayleigh’s method, namely, the length of the mean vector of  $(x - c_q)/\|x - c_q\|$ . The results are plotted with  $\mu$  on the horizontal axis and the Rayleigh score on the vertical axis in Figure 1(a). Additionally, we illustrate the results when  $x \in D_x$  is generated from  $N(0, I)$  and  $q \in D_q$  is generated from  $N(0, \sigma^2 I)$  in Figure 1(b). The results show that the larger  $\mu$  or  $\sigma$  is, indicating a greater divergence between the distributions of  $D_x$  and  $D_q$ , the Rayleigh score increases, that is, hubness is not suppressed.

Note that in this section, we have explained the proposal using the centroids of the datasets,  $c_x$  and  $c_q$ . However, by replacing them with local centroids,  $c_x(q) = \frac{1}{k} \sum_{x \in D_x^{knn}(q)} x$  and  $c_q(x) = \frac{1}{k} \sum_{q \in D_q^{knn}(x)} q$ , where  $D_x^{knn}(q)$  is the set of  $x \in D_x$  that are  $k$ -nearest neighbors of  $q$ , and  $D_q^{knn}(x)$  is the set of  $q \in D_q$  that are  $k$ -nearest neighbors of  $x$ , we can adapt to a more general case where the dataset is generated from a non-unimodal distribution.

## Proposed Hub-CPD Algorithm

In this section, we propose a novel change point detection algorithm, which we refer to as Hub-CPD, consisting of three steps: preprocessing, hubness suppression, and uniformity test, utilizing sliding window approach.

### Step 1: Preprocessing

Let us denote multivariate time-series data with  $m$  variables as  $y_1, y_2, \dots, y_t \in \mathbb{R}^m$ . Our proposed algorithm, as well as the state-of-the-art non-parametric methods for change point detection (the RuLSIF and KL-CPD methods), converts time-series data into sub-series that overlap using a sliding window of size  $w$ . As a result, we obtain a set of  $mw$ -dimensional vectors.

For time-series data from time  $t - (n + w - 2)$  to the current time  $t$ , denoted as  $y_{t-(n+w-2)}, \dots, y_t$ , the Hankel matrix  $Y$  is constructed as follows:

$$Y = \begin{pmatrix} y_{t-(n-1)} & y_{t-(n-2)} & \dots & y_t \\ y_{t-(n)} & y_{t-(n-1)} & \dots & y_{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t-(n+w-2)} & y_{t-(n+w-3)} & \dots & y_{t-(w-1)} \end{pmatrix}.$$

We then obtain  $n$  column vectors of the matrix  $Y$ , that is,  $v_{t-(n-1)}, \dots, v_t$ , which are  $n$  sub-series represented as  $mw$ -dimensional vectors:

$$v_{t-(n-1)} = \begin{pmatrix} y_{t-(n-1)} \\ y_{t-(n)} \\ \vdots \\ y_{t-(n+w-2)} \end{pmatrix}, \dots, v_t = \begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-(w-1)} \end{pmatrix}.$$

Given time-series data, this conversion is useful because we are interested in sub-series rather than the whole: To detect a change in time-series data, we detect the difference between two sets of sub-series, each of which corresponds to those before and after the time when the change occurred. To be more precise, by converting time-series data to  $mw$ -dimensional vectors using a sliding window, we reduce the problem of detecting changes in time-series data to the problem of finding difference between two sets of

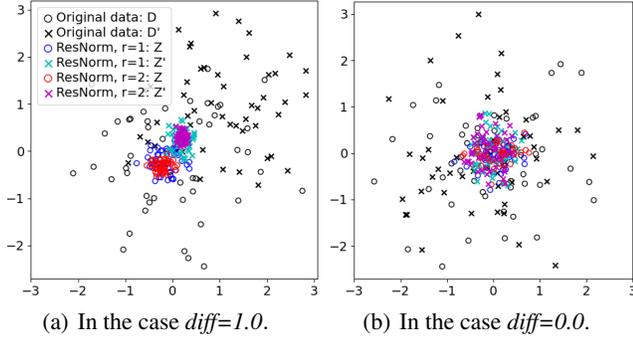


Figure 2: An example of transforming data using the Residual Normalization module. The original data,  $D$  and  $D'$ , are generated from two Gaussian distributions with a mean difference of  $diff$  on each axis. (a) There is a difference between  $D$  and  $D'$ . (b) There is no difference between  $D$  and  $D'$ .

$mw$ -dimensional vectors, which we will denote as  $D$  and  $D'$  hereafter.

## Step 2: Hubness Suppression

To investigate whether  $D = \{v_1, \dots, v_n\}$  and  $D' = \{v'_1, \dots, v'_n\}$  are generated from the same distribution, we utilize the hubness suppression technique described in Eq.(4). Initially, for each  $v_i \in D$ , we calculate the local centroid in  $D'$ , denoted as  $v_i^{(knn)}$ . This local centroid is the mean vector of  $v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(k)} \in D'$ , representing the  $k$ -nearest neighbors of  $v_i$ . Next, we compute the residual vector  $r_i = v_i - v_i^{(knn)}$  obtained by subtracting the local centroid from  $v_i$ . Furthermore, we normalize the residual vector to length 1, obtaining  $z_i = r_i / \|r_i\|$ . The above process is applied similarly to  $v'_i \in D'$ , obtaining  $z'_i = r'_i / \|r'_i\|$ .

We present the module, which we call Residual Normalization, in Algorithm 1. Aside from the parameter  $k$ , the module takes two input datasets, each consisting of  $n$  vectors, and transforms them through  $k$ -nearest neighbor search, residual vector calculation, and vector normalization, producing the corresponding output. This module can be used repeatedly by connecting the output back to the input, allowing for iterative use.

By repeating this process, the original datasets  $D$  and  $D'$  undergo transformations, as demonstrated in Figure 2. In this example, dataset  $D$  consists of  $n = 50$  vectors following a 10-dimensional standard Gaussian distribution, while dataset  $D'$  consists of  $n = 50$  vectors following a modified distribution in which each axis of the 10-dimensional standard Gaussian distribution is increased by  $diff$ . Figure 2 depicts a plot using the first two dimensions of the 10-dimensional vectors. By applying the module, the residual vectors are normalized to a length of 1. When there is a difference between the two datasets (i.e.,  $diff = 1.0$ ), applying the module once results in  $Z$  (blue plot) and  $Z'$  (light blue plot) being separated. Applying the module twice further separates  $Z$  (red plot) and  $Z'$  (magenta plot). However, when there is no difference between the two datasets (i.e.,

## Algorithm 1: Residual Normalization

---

**Input:**  $D = \{v_1, \dots, v_n\}, D' = \{v'_1, \dots, v'_n\}, k$   
**Output:**  $Z = \{z_1, \dots, z_n\}, Z' = \{z'_1, \dots, z'_n\}$

- 1: **for each**  $v_i \in D$  **do**
- 2:   (i) Find  $k$ -nearest neighbors  $v_i^{(1)}, \dots, v_i^{(k)} \in D'$
- 3:   (ii) Compute the residual vector  $r_i = v_i - v_i^{(knn)}$
- 4:   where  $v_i^{(knn)} = \frac{1}{k} \sum_{j=1}^k v_i^{(j)}$
- 5:   (iii) Normalize the residual vector as  $z_i = r_i / \|r_i\|$
- 6: **end for each**
- 7: **for each**  $v'_i \in D'$  **do**
- 8:   (i) Find  $k$ -nearest neighbors  $v_i^{(1)}, \dots, v_i^{(k)} \in D$
- 9:   (ii) Compute the residual vector  $r'_i = v'_i - v_i^{(knn)}$
- 10:   where  $v_i^{(knn)} = \frac{1}{k} \sum_{j=1}^k v_i^{(j)}$
- 11:   (iii) Normalize the residual vector as  $z'_i = r'_i / \|r'_i\|$
- 12: **end for each**
- 13: **return**  $Z = \{z_1, \dots, z_n\}, Z' = \{z'_1, \dots, z'_n\}$

---

## Algorithm 2: Proposed Hub-CPD Algorithm

---

**Input:**  $D = \{v_1, \dots, v_n\}, D' = \{v'_1, \dots, v'_n\}, k, r$   
**Output:**  $\max(R, R')$

- 1: **For each**  $i = 1, \dots, r$  **do**
- 2:    $Z, Z' = \text{ResidualNormalization}(D, D', k)$
- 3:    $D, D' = Z, Z'$
- 4: **end For each**
- 5: Compute the Rayleigh score  $R = \|\frac{1}{n} \sum_{i=1}^n z_i\|$
- 6: Compute the Rayleigh score  $R' = \|\frac{1}{n} \sum_{i=1}^n z'_i\|$
- 7: **return**  $\max(R, R')$

---

$diff = 0.0$ ), applying the module does not result in the separation of  $Z$  and  $Z'$ , and both remain evenly distributed on the unit hyper-sphere.

As can be seen from this observation, the role of Residual Normalization is to extract the differences between the two datasets. By repeating this process, finer differences can be extracted. However, if the number of repetitions is too high, it may extract differences even when the two datasets are generated from the same distribution. Determining the appropriate number of repetitions is a subject for future work.

## Step 3: Uniformity Test

To measure the uniformity of the vectors on a unit hyper-sphere, the proposed change detection algorithm uses the Rayleigh score (Rayleigh 1919), which is calculated as the norm of the mean of the normalized residual vectors:

$$R = \|\frac{1}{n} \sum_{i=1}^n z_i\|, \quad R' = \|\frac{1}{n} \sum_{i=1}^n z'_i\|. \quad (5)$$

If the vectors  $\{z_i\}_{i=1}^n$  and  $\{z'_i\}_{i=1}^n$  are uniformly distributed on a unit hyper-sphere,  $\sum_{i=1}^n z_i$  and  $\sum_{i=1}^n z'_i$  will be close to zero vectors, and hence the values of  $R$  and  $R'$  will be close to zero, respectively. Using the Rayleigh score, if at least one of  $R$  and  $R'$  takes a large value, that is, if  $\max(R, R')$  is large, then the proposed method determines that there

is a difference between  $D$  and  $D'$ . Note that the proposed method, shown in Algorithm 2, can have different variants depending on the number of iterations, denoted as  $r$ , for the Residual Normalization module.

## Experiment on Artificial Data

Using artificially generated time-series data, we compare the proposed Hub-CPD with four state-of-the-art methods, namely RuLSIF (Liu et al. 2013), KL-CPD (Chang et al. 2019), BnB (Hooi and Faloutsos 2019), and mSSA-MW (Alanqary, Alomar, and Shah 2021). We provide details on the parameter settings of these comparison methods in the Appendix A2.<sup>1</sup>

### Generation of Artificial Time-Series Data

We use following two models to generate artificial datasets, both characterized by a mean  $\mu$  and variance  $\sigma^2$ . By adjusting the ShiftValue, we control the magnitude of fluctuations in the generated data.

- (1) **Jumping Mean:** We fixed the variance  $\sigma^2 = 1$  and changed the mean  $\mu$  alternately at each time of  $t_{\text{change}}$ , i.e.,  $\mu = 0 \rightarrow \text{ShiftValue} \rightarrow 0 \rightarrow \text{ShiftValue}, \dots$ .
- (2) **Scaling Variance:** We fixed the mean  $\mu = 0$  and changed the variance  $\sigma^2$  alternately at each time of  $t_{\text{change}}$ , i.e.,  $\sigma^2 = 1 \rightarrow \text{ShiftValue} + 1 \rightarrow 1 \rightarrow \text{ShiftValue} + 1, \dots$ .

The values of the model parameters were changed at every 200-time step intervals, that is, at  $t_{\text{change}} = 201, 401, 601, \dots, 19801$ . Given the parameter values, 20000 data points,  $\{x_t\}_{t=1}^{20000}$ , were generated by the following probabilistic models: (i) i.i.d. Gaussian  $N(\mu, \sigma^2)$ , (ii) i.i.d. t-distribution with location parameter  $\mu$ , scale parameter  $\sigma$ , and 3 degrees of freedom, and (iii) AR model with  $x_t = 0.9x_{t-1} + \epsilon_t$ , where  $\epsilon_t \sim N(\mu, \sigma^2)$ .

In Appendix A1 we present figures plotting six types of time-series data generated by three data generation models (Gaussian, t-distribution, and AR model) with two change patterns (Jumping Mean and Scaling Variance).

### Evaluation Metric

All the compared methods output scores at each time point  $t$ , which we denote as  $Score(t)$ . These methods are expected to output higher scores in response to changes occurring at time  $t_{\text{Change}(i)} = 200i + 1$  for  $i = 1, \dots, 99$ . To capture the response to the changes, we recorded the maximum score up to 100 time points after each  $t_{\text{Change}(i)}$ , denoted as  $A(i) = \max_{t_{\text{Change}(i)} \leq t < t_{\text{Change}(i)} + 100} Score(t)$ . On the other hand, no change occurred at time  $t_{\text{NoChange}(i)} = 200i + 101$  for  $i = 1, \dots, 99$ . We also recorded the maximum score up to 100 time points after  $t_{\text{NoChange}(i)}$ , denoted as  $B(i) = \max_{t_{\text{NoChange}(i)} \leq t < t_{\text{NoChange}(i)} + 100} Score(t)$ . If these methods have the power of change detection,  $A(i)$  should be high values and  $B(i)$  should be low values for  $i = 1, \dots, 99$ . Using these 198 scores, we drew the receiver

<sup>1</sup>The Appendix and code are available at <https://analogy.sakura.ne.jp/AAAI25/appendix.pdf> and [https://analogy.sakura.ne.jp/AAAI25/code\\_and\\_data.zip](https://analogy.sakura.ne.jp/AAAI25/code_and_data.zip).

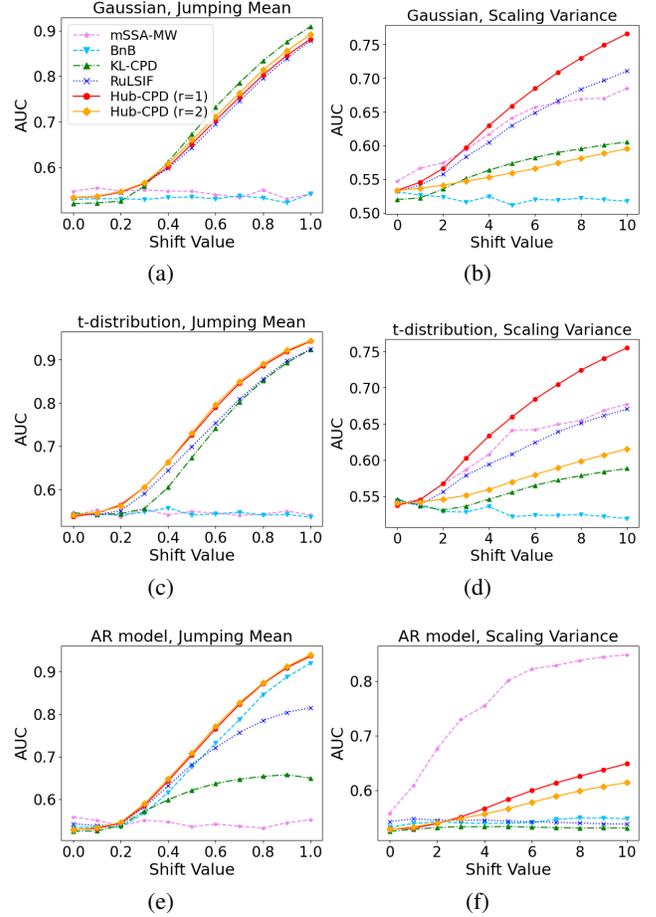


Figure 3: Power of change detection measured by AUC. The higher the AUC, the better. The artificial time-series data were generated under the following settings: (a) Gaussian with jumping mean. (b) Gaussian with scaling variance. (c) t-distribution with jumping mean. (d) t-distribution with scaling variance. (e) AR model with jumping mean. (f) AR model with scaling variance.

operating characteristic (ROC) curve by plotting false positive rate and true positive rate with varying thresholds. We evaluated each of the compared methods using area under the ROC curve (AUC) as the evaluation metric.

### Results

The results are presented in Figure 3. In the setting of Gaussian with jumping mean, KL-CPD achieved the highest AUC. However, the  $r = 1$  variant of the proposed Hub-CPD outperformed existing methods when applied to more realistic time-series data generated by a heavy-tailed t-distribution, where sporadic outliers were included as noise. For time-series data generated by the AR model, the model-based mSSA-MW showed high accuracy under the scaling variance setting. However, among the model-free methods, the proposed method outperformed the others, namely RuLSIF, KL-CPD, and BnB.

	Hub-CPD ( $r = 1$ )	Hub-CPD ( $r = 2$ )	RuLSIF	KL-CPD	BnB	mSSA -MW
PS1	<b>1.000</b>	<b>1.000</b>	0.993	0.653	<b>1.000</b>	0.511
PS2	<b>1.000</b>	<b>1.000</b>	0.999	0.786	<b>1.000</b>	0.531
PS3	0.858	0.832	0.830	0.549	<b>0.902</b>	0.551
EPS1	0.802	<b>0.838</b>	0.560	0.637	0.755	0.541
FS1	0.966	0.985	0.970	0.616	<b>0.987</b>	0.547
SE	0.933	0.945	<b>0.996</b>	0.591	0.867	0.531
Average	0.927	<b>0.933</b>	0.891	0.639	0.919	0.535

Table 1: Experimental results for hydraulic system data (AUC). The bold red numbers represent the highest AUC.

As for the comparison between the  $r = 1$  and  $r = 2$  variants of the proposed method, the  $r = 2$  variant did not achieve higher AUC in the scaling variance setting. However, in the jumping mean setting, it showed slightly higher AUC compared to the  $r = 1$  variant. This suggests that the choice between the  $r = 1$  and  $r = 2$  variants may provide advantages depending on the specific characteristics of the time-series data.

We also provide a sensitivity analysis of the parameters in the proposed Hub-CPD in Appendix A3–A5. A summary of the findings is as follows: For both the  $r = 1$  and  $r = 2$  variants of the proposed method, the following trends are observed:

- The behavior with respect to  $k$  differs: a larger  $k$  is better for jumping mean, while a smaller  $k$  works better for scaling variance.
- The accuracy deteriorates when  $w$  is small, and decreases similarly to other methods (RuLSIF, KL-CPD) as  $n$  decreases.
- When  $n$  decreases, the detection delay becomes shorter, but in some cases changes are not detected.

## Experiment on Real-World Data

In this section, we compare the proposed method with existing methods using real-world data. The real-world data consists of multivariate time-series data obtained from sensors monitoring the state changes of hydraulic system valves (Helwig and Eliseo Pignanelli 2015)<sup>2</sup> and the concentration changes of carbon monoxide detected by gas sensors (Burgués, Jiménez-Soto, and Marco 2018).<sup>3</sup>

In the experiment on hydraulic data, we detected changes in valve condition and measured the accuracy (AUC). There are six types of sensors (PS1, PS2, PS3, EPS1, FS1, SE). Each sensor data is a bivariate time-series of length 1300. In the experiment on gas data, we detected changes in CO concentration and measured the accuracy (AUC). There are two types of sensors (Figaro, FIS). Each sensor data is a multivariate time-series of length 2019 with 7 variables. The parameters of the change detection methods were determined using validation data. The detailed description of the experimental setup can be found in the Appendix B1.

The results are shown in Table 1. Except for the case when the sensor is EPS1, the RuLSIF, BnB, and proposed method

<sup>2</sup><https://archive.ics.uci.edu/dataset/447>

<sup>3</sup><https://archive.ics.uci.edu/dataset/487>

	Hub-CPD ( $r = 1$ )	Hub-CPD ( $r = 2$ )	RuLSIF	KL-CPD	BnB	mSSA -MW
Figaro	<b>0.905</b>	0.842	0.882	0.789	0.754	0.639
FIS	<b>0.999</b>	<b>0.999</b>	<b>0.999</b>	0.916	0.920	0.638
Average	<b>0.952</b>	0.920	0.940	0.852	0.837	0.639

Table 2: Experimental results for gas sensor data (AUC). The bold red numbers represent the highest AUC.

	Hub-CPD ( $r = 1$ )	Hub-CPD ( $r = 2$ )	RuLSIF	KL-CPD	BnB	mSSA -MW
Hydraulic	0.5	0.6	11.8	212.8	1.2	2.1
Gas	0.5	0.6	12.3	216.9	1.1	4.1

Table 3: Runtime comparison (seconds).

achieved an AUC of 0.8 or higher. In particular, the proposed method achieved an AUC of 0.8 or higher for all sensor data. The average AUC across the six sensor data showed that the proposed method with  $r = 2$  had the highest value. However, for PS3, the proposed method with  $r = 1$  outperformed the method with  $r = 2$ . Recalling that in the experiment on artificial data with the scaling variance setting, the proposed method with  $r = 1$  was superior to the method with  $r = 2$ , it is possible that the data of PS3 have the characteristic of variance shifting in response to changes.

The results for gas sensor time-series data are presented in Table 2. In the case of Figaro sensor data, only the proposed method with  $r = 1$  achieved an AUC value exceeding 0.9. For FIS sensor data, all methods except mSSA-MW achieved an AUC value of 0.9 or higher, but both the proposed method and RuLSIF achieved nearly 1.0. Overall, when averaging the results for Figaro and FIS, the proposed method with  $r = 1$  achieved the highest performance.

## Runtime Comparison

Table 3 shows the runtimes of each method for change detection in hydraulic system data and gas sensor data. The table provides the average time each method took to process test data. From the table, the proposed method has shorter execution times compared to the existing methods, indicating that it is a lightweight approach. The execution was performed on an Intel Xeon Gold 6134 CPU with 526GB RAM and an NVIDIA Quadro P5000 GPU with 16GB SGRAM. KL-CPD was run using GPU. RuLSIF was run using MATLAB’s built-in multithreading.

## Conclusion

We introduced a novel method that applies the concept of hubness to change detection. This method includes a new approach for hubness suppression in detecting differences between datasets, supported by analytical proofs. We proposed using a uniformity test on the unit hyper-sphere to advance change point detection by identifying hubness phenomena. Additionally, we developed the Hubness Change Point Detection (Hub-CPD) algorithm and show through comparative experiments that it achieves faster computation times and comparable or superior accuracy to state-of-the-art methods.

## References

- Adams, R. P.; and MacKay, D. J. C. 2007. Bayesian Online Changepoint Detection.
- Alanqary, A.; Alomar, A.; and Shah, D. 2021. Change Point Detection via Multivariate Singular Spectrum Analysis. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 23218–23230. Curran Associates, Inc.
- Aminikhanghahi, S.; and Cook, D. J. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2): 339 – 367.
- Burgués, J.; Jiménez-Soto, J. M.; and Marco, S. 2018. Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models. *Analytica Chimica Acta*, 1013: 13–25.
- Chang, W.; Li, C.; Yang, Y.; and Póczos, B. 2019. Kernel Change-point Detection with Auxiliary Deep Generative Models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 1–14. OpenReview.net.
- Flexer, A. 2016. An Empirical Analysis of Hubness in Unsupervised Distance-Based Outlier Detection. In *Proceedings of 4th International Workshop on High Dimensional Data Mining (HDM), in conjunction with the IEEE International Conference on Data Mining (IEEE ICDM 2016) Barcelona, Spain*.
- Gungor, V. C.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; and Hancke, G. P. 2011. Smart Grid Technologies: Communication Technologies and Standards. *IEEE Transactions on Industrial Informatics*, 7(4): 529–539.
- Hara, K.; Suzuki, I.; Kobayashi, K.; and Fukumizu, K. 2015a. Reducing Hubness: A Cause of Vulnerability in Recommender Systems. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, 815–818. ACM.
- Hara, K.; Suzuki, I.; Kobayashi, K.; Fukumizu, K.; and Radovanovic, M. 2016. Flattening the Density Gradient for Eliminating Spatial Centrality to Reduce Hubness. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 1659–1665. AAAI Press.
- Hara, K.; Suzuki, I.; Shimbo, M.; Kobayashi, K.; Fukumizu, K.; and Radovanović, M. 2015b. Localized Centering: Reducing Hubness in Large-Sample Data. In *Proceedings of the Twenty-Nine AAAI Conference on Artificial Intelligence*, 2645–2651. AAAI Press.
- Helwig, N.; and Eliseo Pignanelli, A. S. 2015. condition monitoring of a complex hydraulic system using multivariate statistics. In *IEEE Instrumentation and Measurement Technology Conference*, 210–215.
- Hooi, B.; and Faloutsos, C. 2019. Branch and Border: Partition-Based Change Detection in Multivariate Time Series. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, 504–512.
- Kawahara, Y.; Yairi, T.; and Machida, K. 2007. Change-Point Detection in Time-Series Data Based on Subspace Identification. In *Proceedings of the 7th IEEE International Conference on Data Mining ICDM*, 559–564.
- Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; and Lin, J. 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104: 799 – 834.
- Liu, S.; Yamada, M.; Collier, N.; and Sugiyama, M. 2013. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43: 72–83.
- Low, T.; Borgelt, C.; Stober, S.; and Nürnberger, A. 2013. The Hubness Phenomenon: Fact or Artifact? In *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, volume 285 of *Studies in Fuzziness and Soft Computing*, 267–278. Springer Berlin Heidelberg.
- Page, E. S. 1954. Continuous Inspection Schemes. *Biometrika*, 41: 100 – 115.
- Radovanović, M. 2015. Hubs in nearest-neighbor graphs: Origins, applications and challenges. NII Shonan Meeting on “Dimensionality and Scalability II: Hands-On Intrinsic Dimensionality”, National Institute of Informatics, Tokyo, Japan, June 28–July 2, 2015.
- Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010a. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research*, 11: 2487–2531.
- Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010b. On the existence of obstinate results in vector space models. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, 186–193. ACM.
- Rayleigh, L. 1919. On the problem of random vibrations, and of random flights in one, two, or three dimensions. *Philosophical Magazine*, 37(220): 321–347.
- Schnitzer, D.; Flexer, A.; Schedl, M.; and Widmer, G. 2012. Local and Global Scaling Reduce Hubs in Space. *Journal of Machine Learning Research*, 13(1): 2871–2902.
- Suzuki, I.; and Hara, K. 2017. Centered kNN Graph for Semi-Supervised Learning. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, 857–860. ACM.
- Suzuki, I.; Hara, K.; Shimbo, M.; Matsumoto, Y.; and Saerens, M. 2012. Investigating the Effectiveness of Laplacian-Based Kernels in Hub Reduction. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 1112–1118. AAAI Press.
- Suzuki, I.; Hara, K.; Shimbo, M.; Saerens, M.; and Fukumizu, K. 2013. Centering Similarity Measures to Reduce Hubs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 613–623.
- Yamanishi, K.; and Takeuchi, J. 2002. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the Eighth*

*ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, 341–349.

Yi, T.; Huang, H.; and Li, H. 2017. Development of sensor validation methodologies for structural health monitoring: A comprehensive review. *Measurement*, 109: 200 – 214.