

---

저자 (Authors)	윤동희, 김수민, 김도현 Donghee Yoon, Suemin Kim, Dohyun Kim
출처 (Source)	<a href="#">신뢰성응용연구 19(2)</a> , 2019.6, 167-178(12 pages) <a href="#">Journal of Applied Reliability 19(2)</a> , 2019.6, 167-178(12 pages)
발행처 (Publisher)	<a href="#">한국신뢰성학회</a> The Korean Reliability Society
URL	<a href="http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08745761">http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08745761</a>
APA Style	윤동희, 김수민, 김도현 (2019). 딥러닝을 이용한 시계열데이터 군집화. <a href="#">신뢰성응용연구</a> , 19(2), 167-178
이용정보 (Accessed)	부산도서관 210.103.83.*** 2021/09/24 13:55 (KST)

---

### 저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

### Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

## 딥러닝을 이용한 시계열데이터 군집화\*

윤동희<sup>1</sup> · 김수민<sup>1,2</sup> · 김도현<sup>1†</sup>

<sup>1</sup>명지대학교 산업경영공학과, <sup>2</sup>엔씨소프트 지식AI랩

## Clustering of Time Series Data using Deep Learning\*

Donghee Yoon<sup>1</sup> · Suemin Kim<sup>1,2</sup> · Dohyun Kim<sup>1†</sup>

<sup>1</sup>Department of Industrial & Management Engineering, Myongji University

<sup>2</sup>Knowledge AI Lab, NCSOFT

**Purpose:** This paper presents the clustering results of time series multiple sensor data using deep neural networks based unsupervised learning algorithm without target variables.

**Methods:** Time series data collected from multiple sensors were clustered using two clustering algorithms based on deep learning: Deep Embedding Clustering (DEC) and Jointly Deep Embedding Clustering (JDEC). DEC and JDEC are designed based on the autoencoder and the convolutional neural network, which are representative neural network structures. They allow high-dimensional data to be represented by low-dimensional data and clustered based on their corresponding low-dimensional values.

**Results:** Two data sets, real time series data collected from manufacturing processes and simulated data, were used in the experiments. The simulated data's performance was evaluated for accuracy, while the clustering performance of the real data was visually evaluated by mapping data and their clusters into a two-dimensional space. The experimental results show that the proposed methods were more accurate than K-means clustering.

**Conclusion:** Real time series data collected from manufacturing processes and simulated data were analyzed and meaningful clustering results were obtained. The proposed methods enabled the unsupervised learning of time series multiple sensor data without target variables by a deep neural network and showed good clustering performance.

**Keywords:** Deep Learning , Deep Embedding Clustering , Time Series Data

### 1. 서론

#### 1.1 연구배경 및 목적

최근 음성인식, 자연어처리, 무인자율주행 등의 다양한 분야에서 딥러닝이 활발히 활용되고 있다. 제조업의

현장에서도 딥러닝을 활용한 공정의 개선에 대한 관심이 많아지고 있다. 즉, 제조 공정 데이터를 이용하여 공정을 자동적으로 제어함으로써 생산성을 향상시키고 공정의 지능화를 추구한다. 일반적으로 공정의 자동화가 어떤 정해진 규칙을 바탕으로 사람보다

\* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. 한국연구재단에서 부여한 과제번호: NRF-2017R1E1A1A01077375).

† 교신저자 ftgog@mju.ac.kr

2019년 5월 31일 접수; 2019년 6월 11일 수정본 접수; 2019년 6월 12일 게재 확정.

정확하고 빠른 속도로 반복적인 동작을 수행함으로써 생산의 효율을 증대 시켰다면, 지능화는 상태를 예측하고 예측된 상태에서 취할 수 있는 솔루션을 제공하거나 예측된 문제를 사전에 스스로 해결하도록 하는 것이다. 딥러닝을 통해 공정데이터를 분석함에 있어 가장 어려운 점은 공정으로부터 수집되는 대부분의 데이터가 라벨 없이 수집되고 있어 일반적인 지도 학습 기반의 딥러닝 알고리즘을 활용할 수 없다는 점이다. 수집된 공정데이터를 분석하기 위해서는 비지도 학습 기반의 시계열데이터 분석 알고리즘이 필요하다.

따라서 본 논문에서는 종속변수에 대한 상황 정의가 없는 상황에서 딥러닝을 활용하여 고차원의 다중 센서의 시계열데이터를 저차원으로 표현하고, 이를 바탕으로 군집분석을 수행함으로써 공정 상태를 예측하고자 한다.

## 1.2 관련연구

딥러닝은 음성인식, 시각적 객체인식, 객체탐지 및 약물발견 및 게놈학 같은 많은 분야에서 성능을 획기적으로 향상시킨 알고리즘이다[13]. 딥러닝은 인간의 신경의 전달과정을 토대로 만들어진 인공신경망을 기반으로 학습되는 방법으로, 뉴런의 시냅스결합 정도를 학습하는 것과 같이 인공신경망의 파라미터를 학습함으로써 비선형문제를 해결할 수 있도록 한다[13]. 일반적인 딥러닝의 학습은 입력 값이 주어졌을 때, 원하는 결과값이 나올 수 있도록 종속변수가 존재하는 데이터를 활용하며 인공신경망에서 최적의 파라미터를 찾는 교사학습이다. 최적의 파라미터로 학습된 모델은 고차원 데이터의 복잡한 비선형 구조를 발견하는데 뛰어난 성능을 보이게 된다.

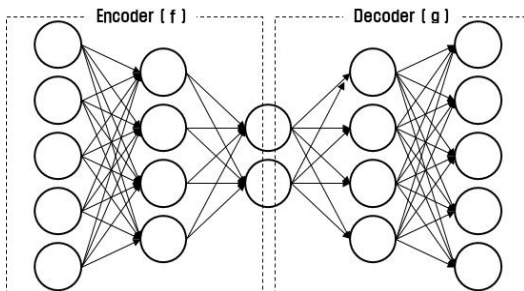


Fig. 1 Autoencoder model

최근들어 독립변수만을 이용하여 학습을 수행하는 비지도학습기반의 딥러닝 알고리즘들이 개발되고 있다. 대표적으로 자기부호화기(Autoencoder) 기반의 군집화 알고리즘이 있다. 자기부호화기는 종속변수 없이 독립변수만 가지고 데이터의 내재된 특징을 학습하는 알고리즘으로 <Fig. 1>과 같이 입력층의 노드의 수와 출력층의 노드의 수가 동일한 구조를 지닌다. 다시 말해 입력데이터가 마지막 출력층에서 동일하게 복원되도록 신경망의 파라미터를 학습하게 된다. 입력층에서 은닉층으로 가는 과정을 부호기(encoder)( $f$ )라고 하고 은닉층에서 출력층으로 가는 과정을 복호기(decoder)( $g$ )라고 하면, 부호기를 통과하여 축약된 데이터( $h(x_i)$ )는 복호기를 통해 복원한다. 복원된  $g(h(x_i))$ 는 실제 입력데이터  $x_i$ 와의 차이가 최소가 되도록 신경망의 파라미터를 학습하고, 실제값과 복원된 값과의 차이는 일반적으로 수식 (1)의 MSE(mean squared error)를 통해 계산된다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - g(h(x_i)))^2 \quad (1)$$

학습이 된 후 부호기의 결과는 PCA(principal component analysis)와 유사하게 고차원의 입력데이터를 저차원의 데이터로 축약하게 된다. PCA가 선형결합을 통해 데이터를 축약하는 반면, 자기부호화기는 비선형 결함을 통해 데이터를 축약하기 때문에, 데이터가 지닌 비선형적인 특징을 보다 잘 표현할 수 있다.

자기부호화기 기반의 군집화 알고리즘은 비지도 학습의 대표적인 구조인 자기부호화기를 이용하여 저차원으로 표현된 데이터를 군집화하는 방법이다. 대표적인 자기부호화기 기반의 군집화 알고리즘인 Deep Embedding Clustering(DEC)는 2016년 Junyuan Xie 가 제안하였다[15]. DEC는 자기부호화기의 부호기에서 추출된 저차원으로 표현된 데이터를 군집화하는 알고리즘으로써 KL-Divergence[10]을 이용하여 신뢰도 높은 군집화가 될 수 있도록 신경망을 학습하게 된다. 또한 DEC는 중심값 기반의 군집화 알고리즘으로 KL-Divergence 손실함수를 통해 중심값을 최적화한다.

또 다른 딥러닝 기반의 군집화 알고리즘인 Discriminatively boosted clustering(DBC)[14]는 DEC와 작동 원리는 동일하지만 합성곱신경망(Convolutional Neural Network)을 사용하여 DEC보다 데이터의 표현력을 높게 된다. 딥러닝의 대표적인 구조인 합성곱신경망은

LeCun, Bengio, and Hinton(1988)이 제안하였는데[12], 합성곱신경망은 일반적으로 이미지 데이터 분석에 많이 사용되지만, 최근에는 시계열데이터, 텍스트 데이터와 같이 다양한 형태의 데이터 분석에도 많이 활용되고 있다. 합성곱신경망은 필터라는 파라미터 행렬을 이용하여 다음 층으로 정보를 전달하게 된다. 이로 인해 합성곱신경망은 이미지의 공간정보를 유지하면서 인접이미지를 고려한 특징을 효과적으로 인식하고, 복수의 필터로 이미지의 특징을 추출할 수 있다. 또한 필터를 공유파라미터로 사용하기 때문에 일반 인공신경망과 비교하여 학습파라미터가 적다. 또한 DBC는 boosting 방법을 활용하여 더 신뢰도 높은 군집화를 수행한다. boosting 방법은 각 군집에 포함될 확률을  $n$ 번 곱해 확률의 상대적인 차이를 극대화하여 군집의 중심으로 데이터가 더 밀집되도록 학습하는 방법이다. 이를 통해 군집의 신뢰도를 향상시킬 수 있다.

Elie Aljalbout에 의해 제안된 Jointly Deep Embedding Clustering[5]는 자기부호화기의 복원 손실함수와 KL-divergence 손실함수[10]을 동시에 고려한 알고리즘으로 앞서 제안된 모델들에 비해 안정적인 학습이 가능하고, 경사하강법을 통해 군집의 최적화가 이루어진다. DEC 알고리즘의 경우, 군집화 손실함수만을 고려하여 학습이 진행되어 선택되는 하이퍼-파라미터에 따라 지역최적화가 나타날 수 있지만 JDEC의 경우 자기부호화기의 복원 손실함수를 동시에 고려하기 때문에 데이터의 표현력을 고려한 안정적인 군집화가 수행된다. <Table 1>은 세 가지 알고리즘을 비교하고 있다. 세 가지 알고리즘은 기본적으로 이미지를 군집화하기 위한 목적으로 개발되었다.

**Table 1** Comparison of clustering algorithms based on autoencoder

Method	structure	Clustering algorithm	Non-clustering loss
DEC Xie, Girshick, and Farhadi[15]	MLP	Network estimates centroid	-
DBC Li, Qiao, Zhang, and Xi[14]	CNN	K-means	-
JDEC Aljalbout, Golkov, Siddiqui, Strobel, and Cremers[5]		Network estimates centroid	Autoencoder reconstruction loss

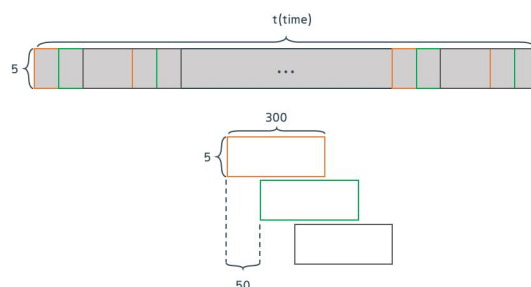
본 논문에서는 원래는 이미지 군집분석을 위해 개발된 DEC(Deep Embedding Clustering)과 JDEC(Jointly Deep Embedding Clustering)를 활용하여 시계열데이터를 군집화하고자 한다. 즉, 딥러닝을 이용해 시계열데이터의 특징을 잘 표현할 수 있는 잠재변수를 도출하고, 이를 바탕으로 시계열데이터를 군집화하고자 한다. DEC와 JDEC를 시계열에 적용하기 위해 시간창(time window)의 형태로 전처리하여 분석하였으며, 합성곱신경망의 필터의 크기를 가로가 긴 형태의 직사각형으로 설계하여 시간의 흐름을 반영하였다.

## 2. 딥러닝 기반 시계열 군집화 알고리즘

### 2.1 시계열데이터 전처리 및 사전학습

딥러닝을 통해 시계열데이터를 군집화하기 위해 연속적인 시계열데이터를 시간창의 형태로 전처리하여 분석하였다. 5개 센서로부터 수집된 시계열데이터를 300프레임(6초) 단위로 분할하고, 50프레임씩 이동하면서  $5 \times 300$ 프레임 형태의 데이터로 변환하였다. 연속적인 값에 대해 시간창의 형태로 전처리를 수행한 이유는 시간이 흐름에 따라서 특정 구간의 상태를 예측하기 위함이다. 연속적인 시계열데이터에서 특정 패턴이 짧은 순간 발생할 수 있기 때문에 이를 탐지하기 위해 <Fig. 2>와 같이 일정폭만큼 중첩된 형태로 전처리를 수행한다.

딥러닝 기반의 군집화 알고리즘을 학습하기 전에 자기부호화기 사전학습을 진행하였다. 사전학습의 목적은 좋은 초기 값을 통해 지역 최적화를 피하기 위해서 진행된다. 만약 초기 값이 좋지 않으면 데이터의 특징을 제대로 추출하지 못하고 군집화를 수행하여 결과적으로 의미가 없는 군집이 만들어질 수 있다.



**Fig. 2** Pre-treatment of time series data

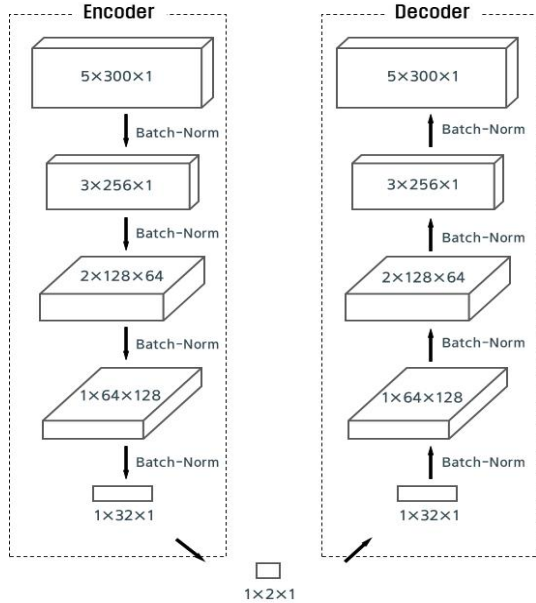


Fig. 3 Autoencoder structure for initial parameter learning

자기부호화기 모델의 구조는 군집화된 데이터의 시각화를 위해 인코딩된 차원은 2차원으로 설정하여 <Fig. 3>과 같이 설계하였다. 특별히 주변의 값의 영향을 많이 받고, 동일한 패턴이 자주 보이는 시계열데이터의 특징 때문에 합성곱신경망을 사용하여 모델을 설계하였다. 합성곱신경망의 필터의 크기를 일반적인 이미지 분석에 활용되는 정사각형 형태의 필터가 아닌 가로가 긴 형태의 직사각형으로 설계하여 시간의 흐름을 반영하였다. 또한 레이어 마다 필터의 수와 크기를 달리하여 다양한 표현으로 데이터를 설명할 수 있도록 하였으며, 각 층 사이에는 과적합을 방지하기 위한 배치정규화[8] 층을 쌓았다. 배치정규화는 일반적으로 파라미터가 점점 작아져 출력층의 정보가 입력층까지 전달되지 못하는 문제(gradient vanishing)와 파라미터가 무한대로 발산하는 문제(gradient exploding)가 일어나지 않도록 하는 아이디어 중의 하나이다. 배치정규화는 각 레이어의 입력 데이터의 분포가 달라지는 공변량의 이동을 방지하여 위와 같은 문제를 해결하게 된다. 이를 통해 학습을 빠르게 하고 안정화시키는 효과를 얻게 된다.

## 2.2 시계열데이터 군집화 알고리즘

본 논문에서는 이미지 데이터를 위해 개발된 두 가지

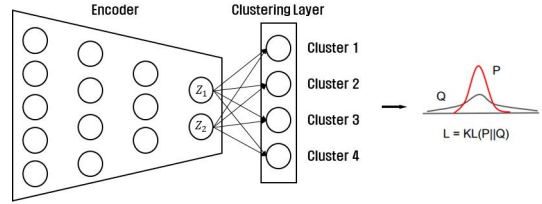


Fig. 4 DEC clustering layer structure

방법론(DEC, JDEC)을 시계열데이터에 적용함으로써 각 구간의 상태를 군집화를 하는 것을 목표로 한다. 두 가지 알고리즘 모두 자기부호화기 기반의 방법론으로 데이터의 잠재적인 특징을 찾고, 이를 기반으로 군집화를 진행한다.

자기부호화기 기반의 대표적인 알고리즘인 DEC는 사전 학습된 자기부호화기의 부호기 부분과 군집화 층을 결합하여 모델을 설계한다. 또한 DEC는 자기부호화기와 Kullback-Leibler(KL) divergence 기법을 사용하여 차원을 축소하면서 잘 표현된 벡터를 통해 군집화를 수행하게 된다.

<Fig. 4>와 같이 군집화 층의 노드의 개수는 군집화하고자 하는 군집의 수와 동일하게 구성하고, 각 군집은  $z$ 의 차원을 가지는 군집의 중심으로 표현한다. 또한 개별 데이터  $i$ 가 군집  $j$ 에 포함될 확률인  $q_{ij}$ 는 수식 (2)를 통해 계산된다. 단,  $q_{ij}$ 를 계산할 때 사용되는 하이퍼-파라미터  $\alpha$ 는 비지도학습 상황에서 검증할 수 없으므로 DEC 논문[15]에서와 같이 1로 고정하고 실험을 진행하였다.

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_j (1 + \|z_i - \mu_j\|^2 / \alpha)^{-\frac{\alpha+1}{2}}} \quad (2)$$

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_j q_{ij}^2 / f_j} \quad (3)$$

$$L_{DEC} = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (4)$$

학습을 진행하기 위해 두 분포의 차이를 계산할 수 있는 KL-Divergence를 손실함수로서 활용한다. KL-Divergence는 어떤 이상적인 분포와 그 분포를 근사하는 다른 분포로부터 사용해 샘플링하여 정보 엔트로피 차이를 계산한다. 이를 위해 먼저 부호기의 결과값인 잠재변수  $z$ 를 이용하여 수식 (2)와 같이 각 데이터가 구해진 군집에

포함될 확률  $q_{ij}$ 를 계산한다. 또한 목표확률  $p_{ij}$ 는 군집에 포함될 확률인  $q_{ij}$ 를 제공하여 정의 하게 되는데, 이를 통해 각 데이터별로 가능성이 높은 군집에 더 높은 가중치를 주어서 데이터를 군집화할 수 있게 되고, 순수도 높은 군집을 얻을 수 있게 된다.  $f_j$ 는 군집에 포함된 데이터의 수를 의미한다.  $p_{ij}$ 와  $q_{ij}$ 의 정보 엔트로피 차이를 최소화 하도록 학습을 하게 되면 관측치들이 군집의 중심에 밀집되어 있는 순수도 높은 군집화를 수행할 수 있다. 목표분포와 근사분포로부터 계산된 KL-Divergence값 즉  $L_{DEC}$ 가 최소가 되도록 모델을 학습한다. 그리고 수식 (5), 수식 (6)과 같이 경사하강법 방법을 이용하여  $L_{DEC}$ 을 최소화하는 군집의 중심  $\mu_j$ 와 잠재 변수  $z_i$ 를 구한다.

$$\frac{\partial L_{DEC}}{\partial z_i} = \frac{\alpha+1}{\alpha} \sum_j (1 + \frac{\|z_i - \mu_j\|^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \quad (5)$$

$$\frac{\partial L_{DEC}}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \sum_i (1 + \frac{\|z_i - \mu_j\|^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \quad (6)$$

언어진 최적의 군집의 중심  $\mu_j$ 를 통해 DEC는 중심 값 기반의 군집화 기법을 수행한다. 단, 빠른 수렴과

지역최적화를 피하기 위해 초기의 군집의 중심은 K-means 알고리즘[3]을 통해 계산한다.

2018년 Elie Aljalbout가 제안한 JDEC[5]는 군집화를 하는 방법이 DEC와 유사하다. 하지만 표현학습을 하는 부호기를 학습하는 과정에서 자기부호화기의 복원 손실함수를 추가적으로 계산하여 학습한다.

JDEC는 <Fig. 6>과 같은 구조를 통해 학습을 진행한다. DEC와의 차이점은 부호기뿐만 아니라 복호기를 통해 나오는 복원에 대한 손실함수도 고려하여 표현을 학습한다는 점이다. DEC의 경우 군집화에 대한 오류만을 고려해 학습을 진행하기 때문에 학습과정 중 데이터의 표현력을 잃어버릴 수 있다. 즉, 자기부호화기의 복원 손실함수가 크게 증가할 수 있다. 하지만 JDEC의 경우 아래의 수식 (7)과 같이 복호기를 통해 나오는 복원에 대한 손실함수도 함께 고려하기 때문에 데이터에 대한 표현력을 잃지 않고 최적의 군집화를 수행할 수 있다.  $L_{JDEC}$ 의 첫 번째 항은 DEC의 손실함수와 동일하며 두 번째 항은 자기부호화기를 통해 계산되는 복원손실함수를 의미한다.

$$L_{JDEC} = \alpha \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1-\alpha) \frac{1}{n} \sum_{i=1}^n (x_i - g(h(x_i)))^2 \quad (7)$$

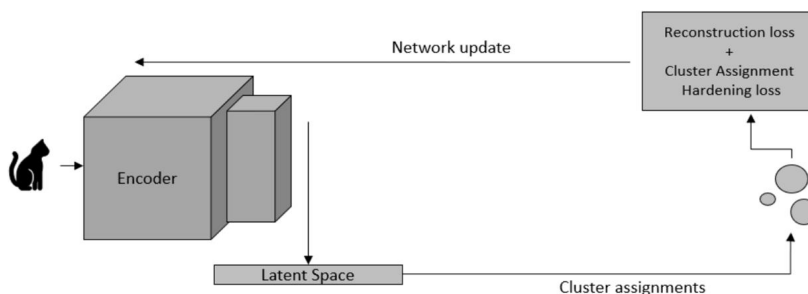


Fig. 5 DEC structure[5]

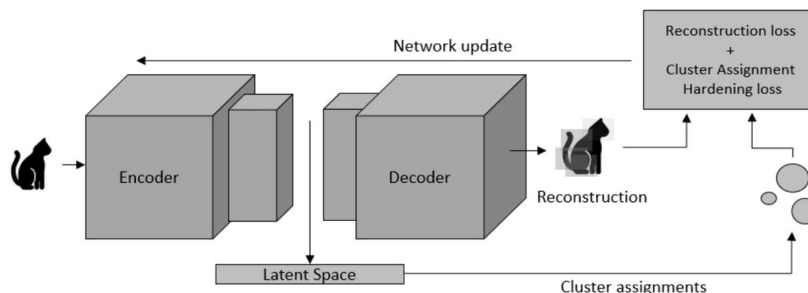


Fig. 6 JDEC structure[5]

원활한 군집화를 위해 복원손실함수와 KL-divergence의 손실함수의 반영비율  $\alpha$ 를 변화시켜 가면서 학습하게 된다.

### 3. 딥러닝 기반 시계열 군집화 알고리즘 성능평가

#### 3.1 실험 데이터

실험에 사용한 데이터는 두 종류이다. 첫 번째는 알고리즘의 성능을 평가하기 위해서 가상의 시계열 데이터를 생성하고 이를 바탕으로 분석을 진행한다. 종속변수가 없는 상황에서 군집화를 수행했을 때 같은 신호가 같은 군집에 포함되는 지에 대한 평가를 진행하고자 한다.

수식 (8)을 이용하여 cosine함수와 tangent함수를 조합한 파동의 폭과 너비가 다른 4가지의 시계열 데이터를 생성하였고 실제 데이터와 유사한 환경을 만들어 주기위해서 잡음을 추가하였다. 잡음은 수식 (9)와 같이 일양분포에서 랜덤하게 추출하였다. 생성된 시계열 데이터는 <Table 2>와 같다.

$$\text{signal} = H \cdot \cos(w\pi) + H \cdot \tan(w\pi) \quad (8)$$

$$\text{noise} = \text{uniform}(-3H, 3H) \quad (9)$$

두 번째 실험 데이터는 실제 공정의 5개의 다중 센

Table 2 Simulation data generation rules

Signal	Combination	H(height)	W(width)
A	signal+noise	3	0.05
B	signal+noise	2	0.01
C	signal+noise	5	0.01
D	noise	1	-

서로부터 나온 시계열데이터이다. 실제 데이터의 경우 가상의 데이터와는 달리 정의된 군집이나 종속변수가 없는 것이 특징이다. 따라서 분류의 정확도를 계산하는 것은 불가능하기 때문에 군집화된 결과를 정성적으로 평가하였다.

#### 3.2 성능평가 측도

성능평가는 앞서 설명한 바와 같이 가상의 데이터에 대해서 진행된다. 비지도 학습인 군집화는 특정 종속변수를 예측하는 것이 아닌 데이터의 유사도를 바탕으로 데이터를 분류하는 것이다. 이러한 특징 때문에 각 관측치들의 정확한 군집을 파악하기 어렵다. 따라서 다수의 군집 중 포함된 관측치의 수가 가장 많은 군집을 해당 관측치의 군집으로 가정하고, 예측된 군집과 실제 군집과의 차이를 통해 정확도를 계산한다.

$$\text{Accuracy} = \max_m \frac{\sum_{i=1}^n 1\{l_i = m(c_i)\}}{n} \quad (10)$$

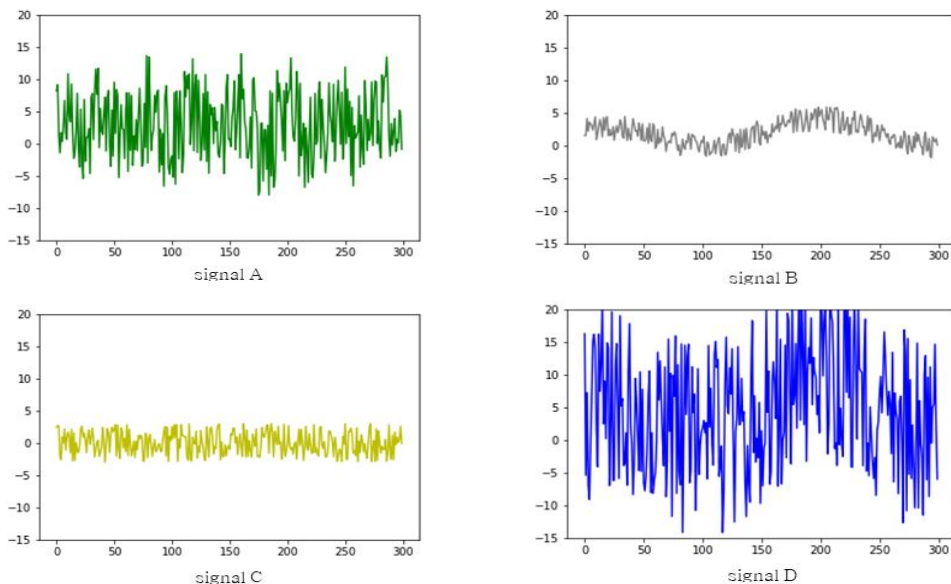


Fig. 7 Signal A, B, C, D



군집분석의 경우, 데이터별 실제 라벨값이 존재하지 않으므로, 성능 평가가 어렵다. 따라서 전체 데이터에 대한 실험결과, 실제 라벨과 일치하는 데이터가 가장 많은 군집을 해당 라벨로 정의하고 정확도를 계산하여 성능을 평가한다. 수식 (10)[15]에서  $n$ 은 총 관측치의 수,  $c_i$ 는 학습모델로부터 예측된  $i$ 번째 관측치의 군집을 나타낸다.  $m$ 은 예측된 군집을 실제 라벨로 변환하는 함수이며,

실제 군집  $l_i$ 가 예측된 군집  $m(c_i)$ 와 일치하면 1, 일치하지 않으면 0으로 계산하여 모델의 정확도를 계산한다.

### 3.3 실험 방법

DEC 모델 실험의 전반적인 프로세스는 <Fig. 8>과 같다. 시계열데이터를 시간창의 형태로 전처리한 후, MSE

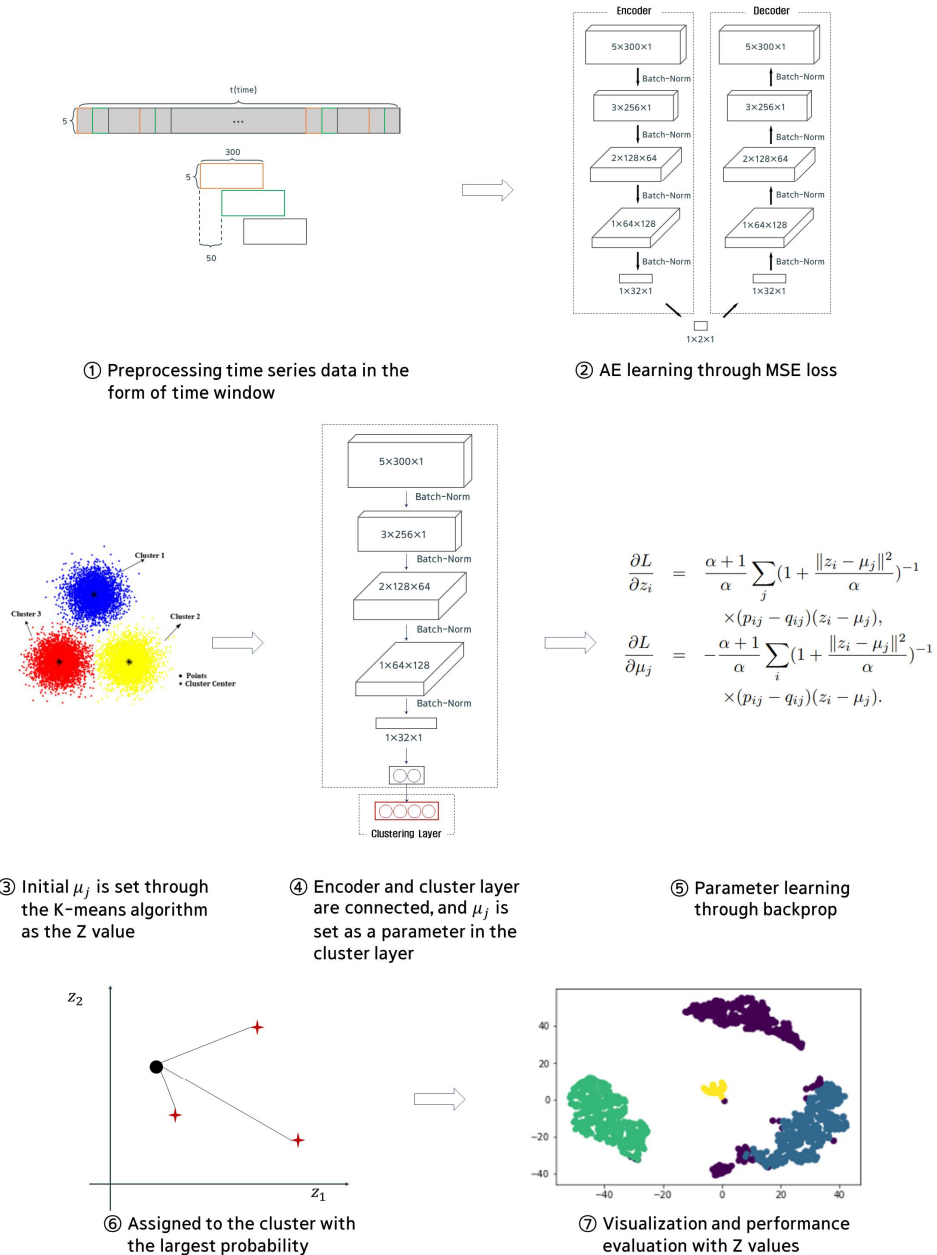


Fig. 8 Experimental procedure for DEC



손실함수를 통해 자기부호화기를 이용하여 사전학습을 진행한다. 그리고 학습된 부호기를 통해 구해진 잠재변수  $z$  값을 바탕으로 K-means 분석을 수행하여 초기의 중심값  $\mu_j$  을 구한다. 구해진 중심값으로 부호기와 군집층을 연결시키고 KL-divergence 손실함수를 최소화 하도록 DEC 모델을 학습시킨다. 학습의 결과로 각 군집의 중심값이 도출되며, 중심값을 기반으로 가장 가까운 군집에 데이터를 할당시키고 시각화하여 모델의 성능을 평가한다.

터할당시키고 시각화하여 모델의 성능을 평가한다.

JDEC의 경우는 DEC와 달리 사전에 학습된 전체 자기부호화기 모델을 사용하게 된다. 아래의 <Fig. 9>와 같이 JDEC 학습은 KL-divergence와 MSE 손실함수를 모두 사용하게 되고, 두 손실함수의 가중치를 변화시켜가며 학습을 진행한다. 이외의 모든 프로세스는 DEC 모델과 동일하다.

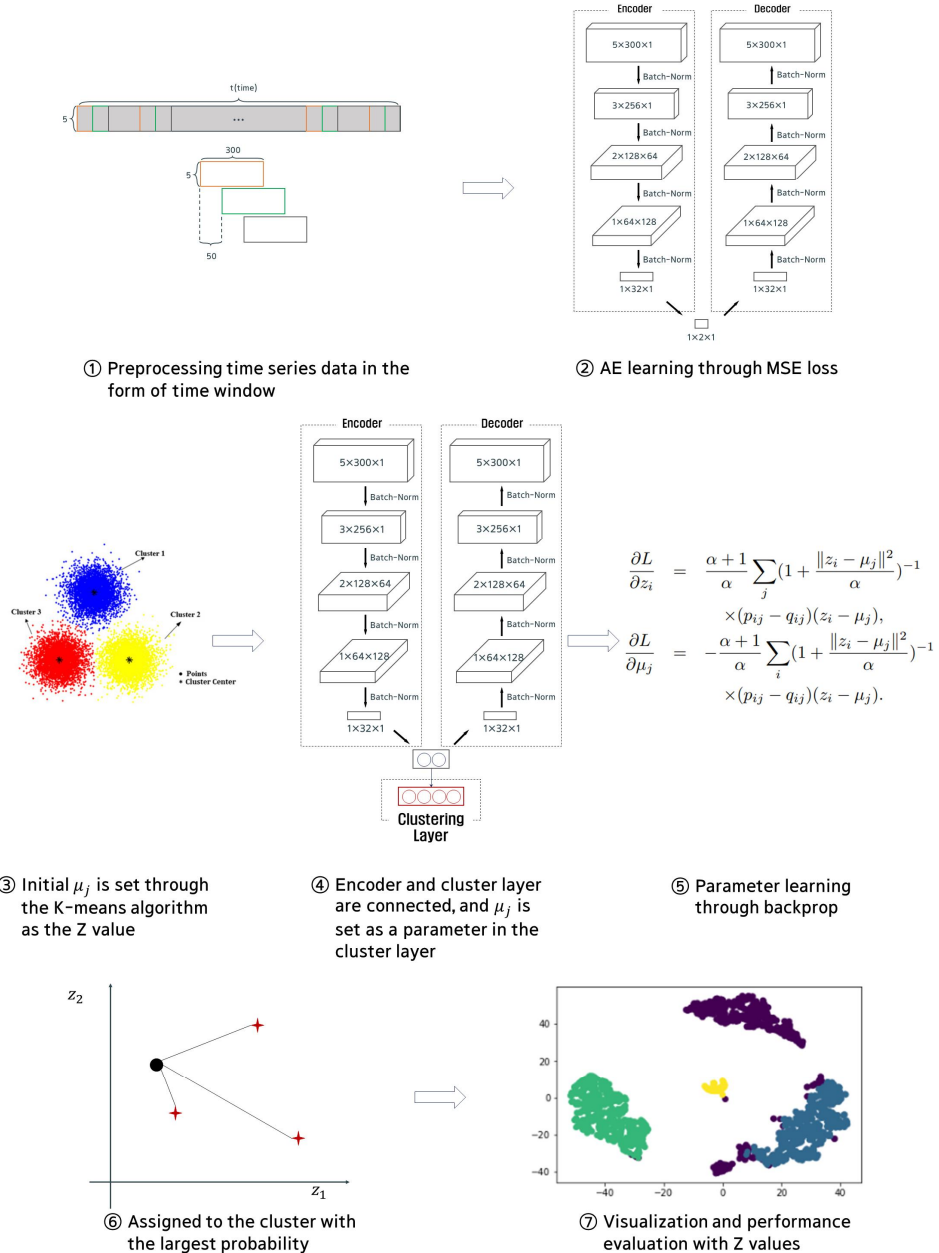


Fig. 9 Experimental procedure for JDEC

성능평가는 제안된 알고리즘과 일반적인 군집화 알고리즘인 K-means의 결과를 비교하여 진행하였다. K-means 알고리즘은 클러스터의 개수 K를 사전에 설정하고, K개의 중심값을 랜덤하게 선택하고 중심값까지의 거리가 최소인 군집에 각 데이터를 할당한다. 새롭게 할당된 데이터를 기반으로 새로운 중심값을 계산하고, 이를 바탕으로 각 데이터를 K개의 군집에 할당한다. 위와 같은 과정을 중심값이 수렴할 때까지 반복하여 최적의 군집을 찾게 된다[3]. K-means 군집화 알고리즘은 딥러닝을 활용한 군집화와 달리 데이터의 차원축소의 과정이 없기 때문에 군집화 결과를 시각적으로 보여주기 위해 t-SNE 알고리즘을 사용하였다. t-SNE 알고리즘은 고차원 데이터를 탐색하는데 사용되는 비선형 차원감소 알고리즘으로, 고차원 데이터를 시각화하기 위해 사용되는 대표적인 방법이다[11].

모델을 학습하기에 앞서 하이퍼-파라미터를 선택하여야 한다. 하이퍼-파라미터란 모델의 학습을 통해 최적화되는 파라미터가 아닌 분석하기 전에 직접 설정해야 하는 변수를 나타낸다. 선택해야 하는 하이퍼-

파라미터는 학습 반복수인 에포크, 업데이트의 반영 비율인 학습률, 한번 학습되어지는 데이터의 크기인 배치의 크기, 최적화 함수, 레이어의 수, 필터의 크기, 잠재변수의 차원 등의 파라미터가 있다. 하지만 현실적으로 모든 파라미터를 학습 할 수 없기 때문에 일부 파라미터를 고정하고 성능에 크게 작용하는 파라미터를 변화시켜가며 실험을 진행하였다.

## 4. 실험결과

### 4.1 시뮬레이션 데이터 성능평가

가상데이터의 실험결과는 다음과 같다. 성능이 가장 좋은 알고리즘은 JDEC이고, DEC, JDEC와 같이 딥러닝을 활용한 알고리즘이 더 좋은 성능을 보인다. 군집의 형태 또한 DEC와 JDEC가 K-means보다 관측치들을 명확히 분류한 것을 확인할 수 있다. 정확도는 JDEC가 97.8%로 가장 높았으며, K-means 알고리즘이 71.1%로 가장 낮은 정확도를 보였다.

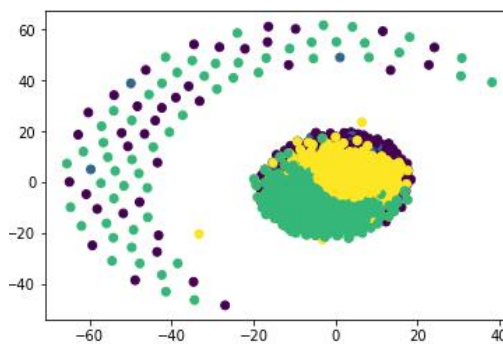
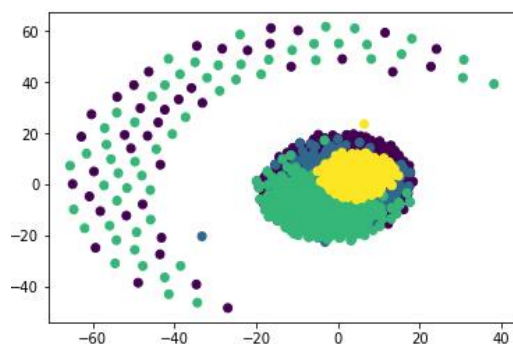
<Fig. 10>은 K-means 알고리즘의 결과이다. 고차원의 데이터를 시각화하기 위해 t-SNE를 이용하여 2차원으로 데이터를 맵핑하였다. 그 결과 실제라벨과 예측된 군집화 결과 차이가 큰 것을 확인할 수 있다.

**Table 3** Hyper parameters for DEC and JDEC

Hyper parameter	Value	Method
Hidden layer	9	Fix
Optimization function	Adam optimizer	Fix
Dimension of latent variable	2~5	Random
Number of clusters	2~6	Random
Learning rate	0.01~0.00001	Random
Batch size	$2^6 \sim 2^{10}$	Random
Epoch	0~10000	Random

**Table 4** Comparison of simulation data performance

Algorithm	Accuracy(%)
K-means	71.1
DEC	92.3
JDEC	97.8



**Fig. 10** t-SNE actual labels(left) and predicted labels(right) by K-means

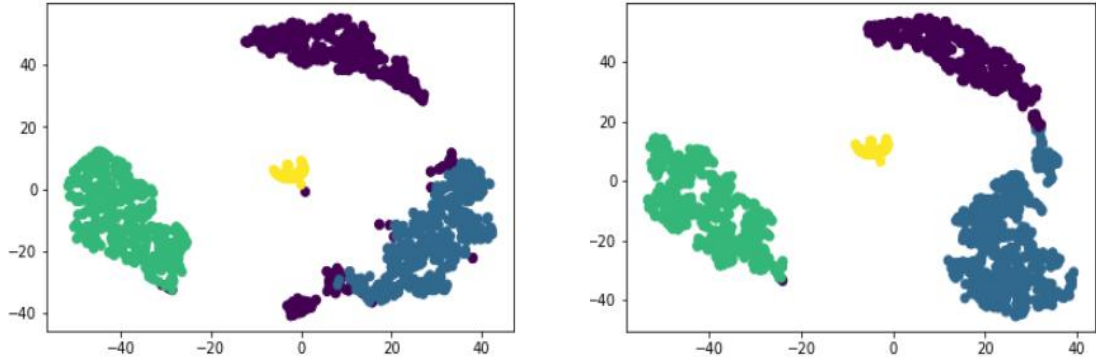


Fig. 11 Actual labels(left) and predicted labels(right) by DEC

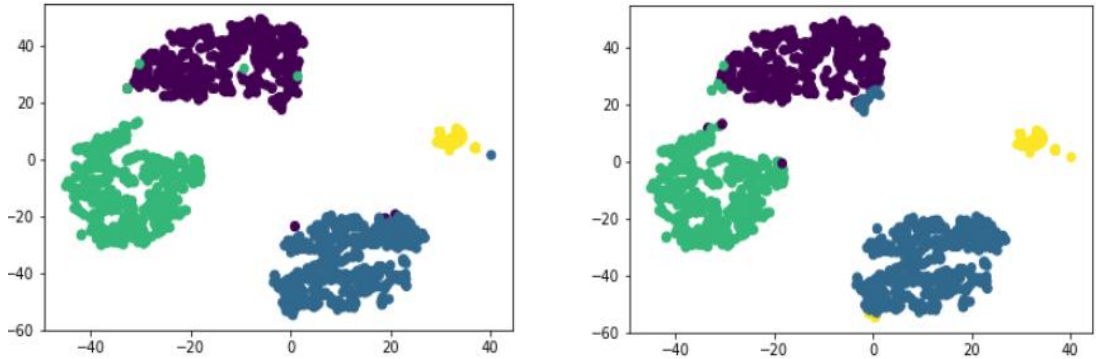


Fig. 12 Actual labels(left) and predicted labels(right) by JDEC

<Fig. 11>과 <Fig. 12>는 DEC와 JDEC알고리즘의 결과이다. 부호기를 통해 구해진 저차원의 z값을 시각화 하였다. K-means 알고리즘에 비해 DEC와 JDEC에 의한 4개의 군집이 명확히 분류되는 것을 확인할 수 있다. DEC의 결과에서는 일부 군집이 중첩되는 모습을 보이기도 하나 JDEC는 중첩 없이 군집을 명확히 구별하고 있다.

#### 4.2 실제 센서데이터를 통한 성능평가

공정에서 나온 다중 센서를 통해 수집된 데이터를 분석하여 제안 알고리즘의 성능을 평가하기 위한 실험을 진행하였다. 가상의 데이터와 같이 K-mean, DEC, JDEC 알고리즘의 성능을 비교하였다. 수집된 센서는 총 5가지이다. 사전학습된 자기부호화기의 파라미터는 두 알고리즘에 동일하게 적용하였으며, 자기부호화기는 최적의 MSE 손실함수를 가지는 모델로 선택하였다. 도출된 최적의 하이퍼-파라미터는 <Table 5>와 같다.

**Table 5** Parameter selection for process data analysis

	Hyper parameter	Value
Common	Number of data	Train2123/ Validation200
	Dimension of latent variable	2
	Optimization function	Adam
AE	Batch size	128
	Learning rate	0.0001
	Epoch	10000
DEC	Batch size	128
	Learning rate	0.0001
	Number of clusters	4
JDEC	Epoch	5000
	Batch size	64
	Learning rate	KLD 0.6/ MSE 0.4
	Number of clusters	4
	Epoch	3000

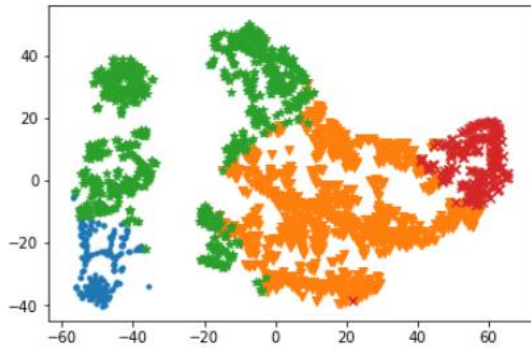


Fig. 13 K-means result for real sensor data

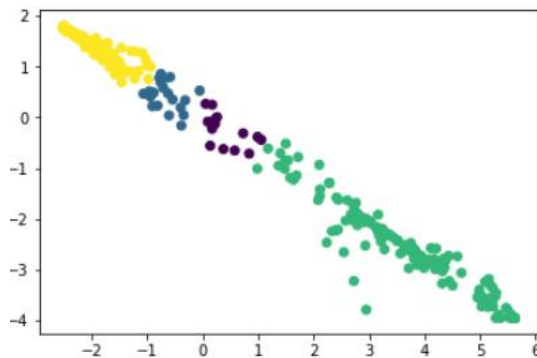


Fig. 14 DEC result for real sensor data

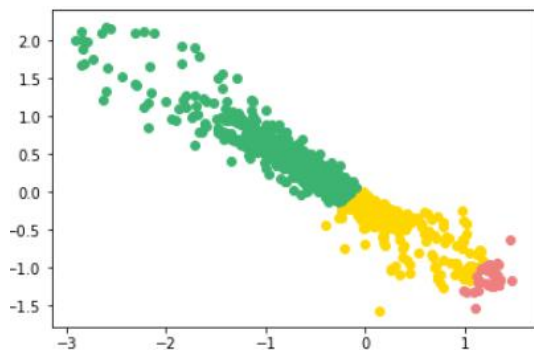


Fig. 15 JDEC result for real sensor data

시계열 센서데이터를 분석한 결과는 아래와 같다. <Fig. 13>~<Fig. 15>와 같이 K-means의 경우 군집화가 확실히 이루어지지 않았지만 딥러닝을 활용한 DEC와 JDEC는 2차원 공간에 관측치들을 명확히 분리해 내는 것을 확인하였다.

결론적으로 첫 번째 가상의 데이터를 분석한 결과 딥러닝을 활용한 모델들(DEC, JDEC)이 K-means 알고리즘보다 더 좋은 성능을 보이는 것을 수치적으로 확

인할 수 있었으며, JDEC가 가장 좋은 성능을 보여주었다. 또한 군집을 시각화 하였을 때, 딥러닝을 활용한 모델에서 데이터들이 각 군집의 중심에 밀집되어 있는 형태를 보였다. 그리고 JDEC는 다른 알고리즘에 비해 군집별로 데이터수가 좀 더 균일하게 할당되는 특징을 보였다. 두 번째 실제 공정의 센서데이터를 분석한 결과에서도 K-means 알고리즘에 비해 딥러닝을 활용한 모델이 더 명확한 군집화를 수행하는 것을 확인하였으며, 공정의 현재 상태를 명확히 파악할 수 있는 의미있는 군집화를 수행하였다.

## 5. 결론 및 향후 연구과제

본 논문은 이미지 분류를 위해 제안된 딥러닝 기반의 군집화 모델인 DEC와 JDEC를 기반으로 다중센서 시계열데이터를 군집화하는 방법을 제안하였다. 연속적인 시계열데이터 특징을 반영하기 위해 시간창의 형태로 전처리하여 분석하였으며, 합성곱신경망의 필터 크기를 가로가 긴 형태의 직사각형으로 설계하여 시간의 흐름을 반영하였다. 제안된 방법에서 합성곱신경망 자기부호화기를 통해 추출된 저차원의 시계열 특징 데이터를 군집화에 활용하였으며, 저차원에서 각 군집의 중심에 데이터가 밀집될 수 있도록 목표분포를 정의하고, 이를 바탕으로 모델을 학습하여 제안된 방법이 우수한 군집화 성능을 보여주었다.

제안된 방법의 우수성을 검증하기 위해서 가상의 데이터와 실제 공정에서 수집된 센서데이터를 통해 분석을 진행하였으며, 모델에서 추출된 잠재변수를 2차원으로 시각화하여 검증하였다. 또한 제안된 DEC와 JDEC 기반의 모델들과 딥러닝을 활용하지 않은 K-means 모델과 성능을 비교하였다. 그 결과 딥러닝을 활용한 모델이 더 우수한 군집화 성능을 보이는 것을 확인하였다. 결론적으로 제안한 방법을 통해 비지도 학습 상황에서도 시계열데이터를 군집화할 수 있었으며, 다중 채널의 센서데이터에 대해서도 제안된 방법이 우수한 군집화 성능을 보이는 것을 확인하였다.

추후 연구주제는 다음과 같다. 첫째, 본 논문에서는 공정에서 수집된 시계열 센서데이터를 활용하여 성능 평가를 하였으나 향후에는 다른 도메인의 데이터셋이나 다수의 채널 센서 데이터에 대한 실험을 통해 제안된 방법의 우수성을 추가적으로 검증할 필요가 있다.

둘째, 딥러닝의 특성상 배치의 크기, 학습률, 에포크, 초기 파라미터 등에 따라 학습 성능이 안정적이지 못한 측면이 있다. 또한 군집 중심의 초기 값에 따라 K-means 알고리즘 결과가 많은 차이를 보인다. 이를 보완하여 초기값이나 하이퍼-파라미터의 영향을 덜 받는 안정적인 모델을 연구할 필요가 있다.

셋째, 본 논문에서는 자기부호화기를 활용하여 데이터 특징을 추출하였으나, 최근 연구되고 있는 다양한 특징추출 기법을 적용하여 추가적인 연구가 진행될 필요가 있다. 뿐만 아니라 시계열 센서데이터의 연속적인 특성을 반영한 모델에 대한 연구 또한 필요하다.

## References

- [1] Krizhevsky, A., and Hinton, G. (2009). "Learning multiple layers of features from tiny images". Technical Report, University of Toronto.
- [2] Bengio, Y., Courville, A., and Vincent, P. (2013). "Representation learning: A review and new perspectives". *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 35, No. 8, pp. 1798-1828.
- [3] Ding, C., and He, X. (2004). "K-Means clustering via principal component analysis". *Proceedings of the 21<sup>st</sup> International Conference on Machine Learning (ICML)*.
- [4] Dizaji, K. G., Herandi, A., Deng, C., Cai, W., and Huang, H. (2017). "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization". *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5747-5756.
- [5] Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., and Cremers, D. (2018). "Clustering with deep learning: taxonomy and new methods". *arXiv:1801.07648*.
- [6] Geoffrey, H., and Ruslan, S. (2006). "Reducing the dimensionality of data with neural networks". *Science*, Vol. 313, No. 5786, pp. 504-507.
- [7] Goodfellow, I., Pouget-Abadie, J., Mirza, M. Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). "Generative adversarial nets". *arXiv:1406.2661*.
- [8] Ioffe, D., and Szegedy, C. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *arXiv:1502.03167*.
- [9] Kingma, D., and Ba, J. (2014). "Adam: A method for stochastic optimization". *arXiv:1412.6980*.
- [10] Kullback, S. (1959). "Information theory and statistics". John Wiley & Sons.
- [11] Maaten, L. V. D., and Hinton, G. (2008). "Visualizing data using t-SNE". *Journal of Machine Learning Research*, Vol. 9, pp. 2579-2605.
- [12] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). "Backpropagation applied to handwritten zip code recognition". *Neural Computation*, Vol. 1, No. 4, pp. 541-551.
- [13] LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep learning". *Nature*, Vol. 521, No. 7533, pp. 436-444.
- [14] Li, F., Qiao, H., Zhang, B., and Xi, X. (2017). "Discriminatively boosted image clustering with fully convolutional auto-encoders". *arXiv:1703.07980*.
- [15] Xie, J., Girshick, R., and Farhadi, A. (2016). "Unsupervised deep embedding for clustering analysis". *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 478-487.