Source Selection Methodology in Transfer Learning of Time Series Data

2023 . 10 . 13

최종원

자기소개

- **최종원(Choi Jong Won)**
  - 고려대학교 산업경영공학과 재학 중
  - Data Mining & Quality Analytics Lab(김성범 교수님 연구실)
  - 석사 과정(2023.03 ~)

- **연구 관심 분야**
  - Machine learning & Deep learning Algorithms
  - Multivariate Time Series Data

Contents

- **Introduction**
  - Label-efficient Learning Methods in Time Series Data
    - Review on In-domain Representation Learning for Time Series Data
    - Review on Cross-domain Representation Learning for Time Series Data
- **Source Selection Methodology in Transfer Learning of Time Series Data**
  - IDS : Inter-Datasets Similarity
  - SMS : Source Model Selection
- **Conclusion**

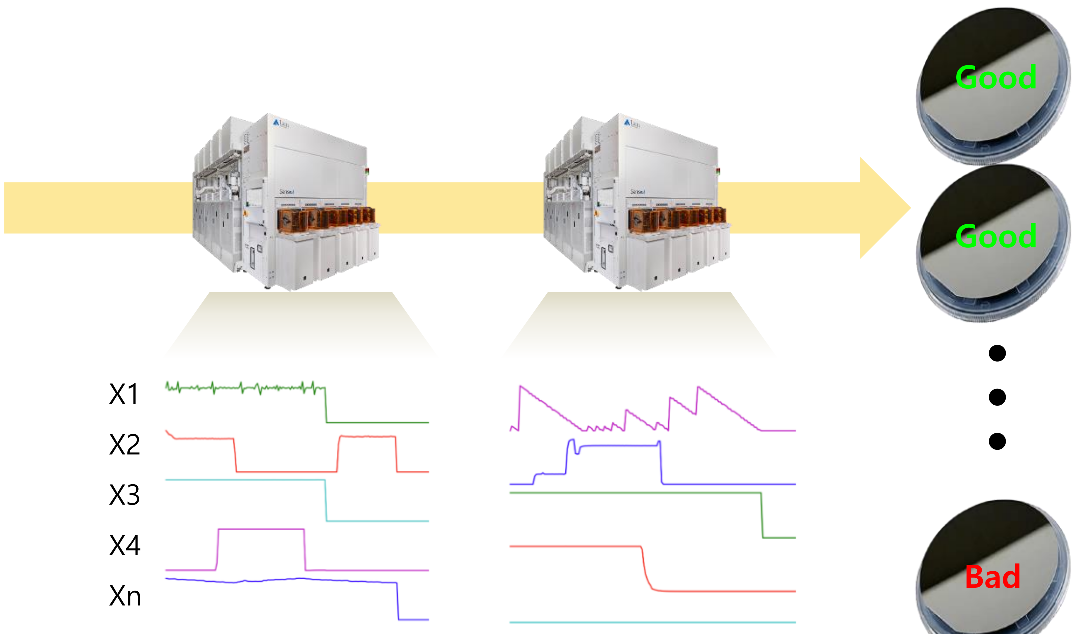
Introduction

Introduction

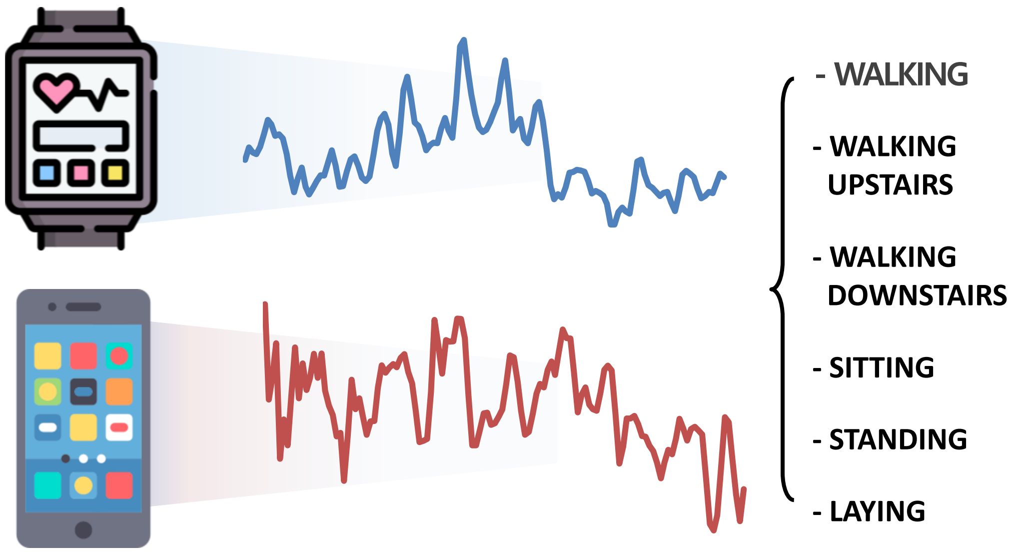
Label-efficient Learning Methods in Time Series Data

❖ **레이블링 된 시계열 데이터의 부족**

- 센서 데이터 : 생산 설비로부터 수집, Domain expert 가 레이블링 한 부분은 극소수
- 헬스케어 분야 (ex : HAR; Human Activity Recognition) : Sample 개수 확보가 어려움



센서 데이터



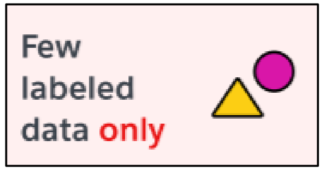
HAR

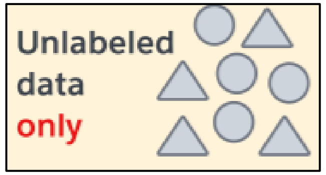
Introduction

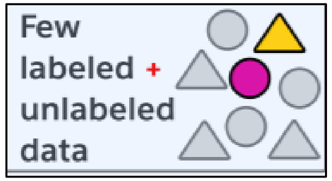
Label-efficient Learning Methods in Time Series Data

❖ **레이블링 데이터 부족 Case**

- Case 1 : Few labeled data only
- Case 2 : Unlabeled data only
- Case 3 : Few labeled data + Unlabeled data







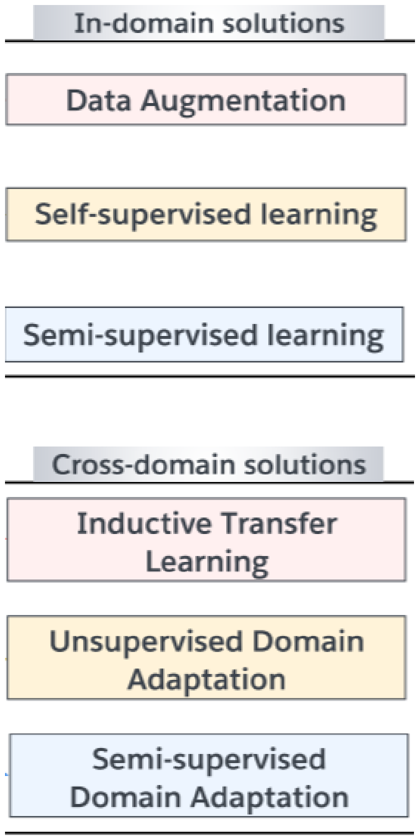
Introduction

Label-efficient Learning Methods in Time Series Data

❖ **레이블링 데이터 부족을 극복 하기 위한 방법론**

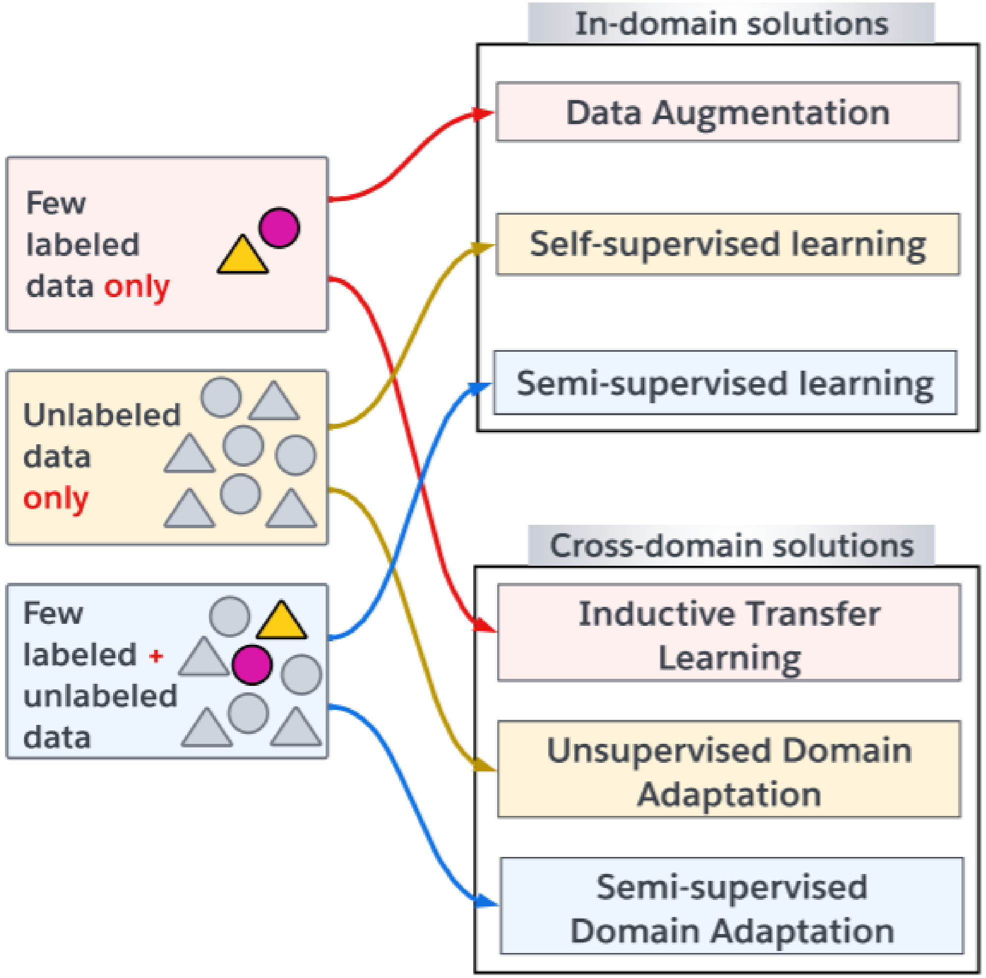
- **In-domain methods**
  - 해당 Task 를 수행하기 위한 데이터로부터 Representation 도출에 중점
  - Data Augmentation, Self-supervised learning, Semi-supervised learning

- **Cross-domain methods**
  - Source domain != Target domain
  - Source domain 과 Target domain 이 유사한 분포와 Class 경계를 가지도록 학습
  - Transfer learning, Unsupervised domain adaptation, Semi-supervised domain adaptation



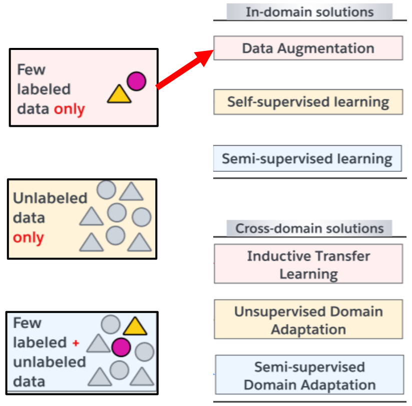
Introduction

Label-efficient Learning Methods in Time Series Data



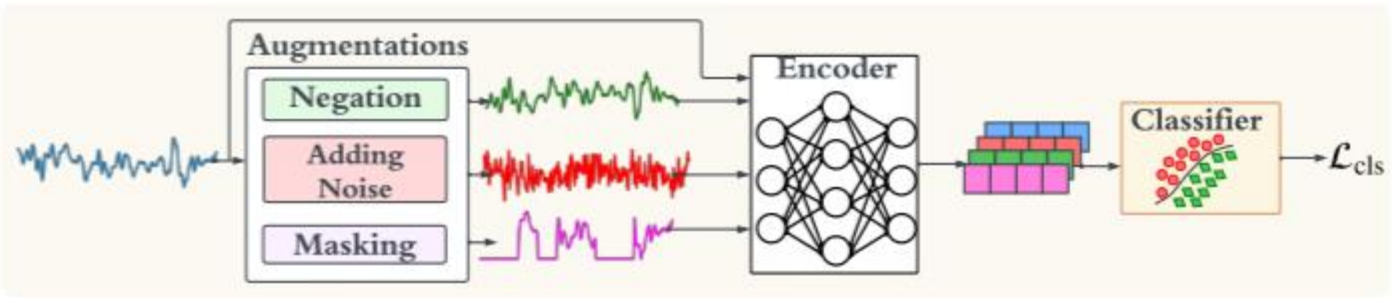
Introduction

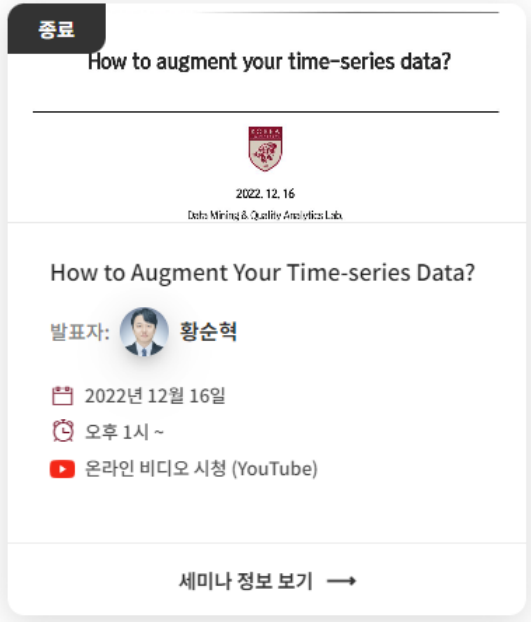
Review on In-domain Representation Learning for Time Series Data



❖ **Data Augmentation**

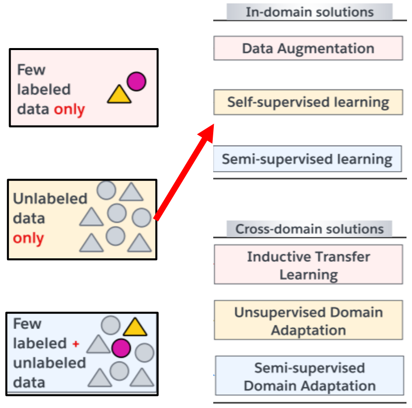
- 데이터 변형 – Add Noise, Masking 등
- 합성 데이터 생성 – GAN
  - ActivityGAN – 1D CNN Network로 Human Activity Recognition 합성 데이터 생성
  - DCGAN – 인공 EEG 데이터 생성





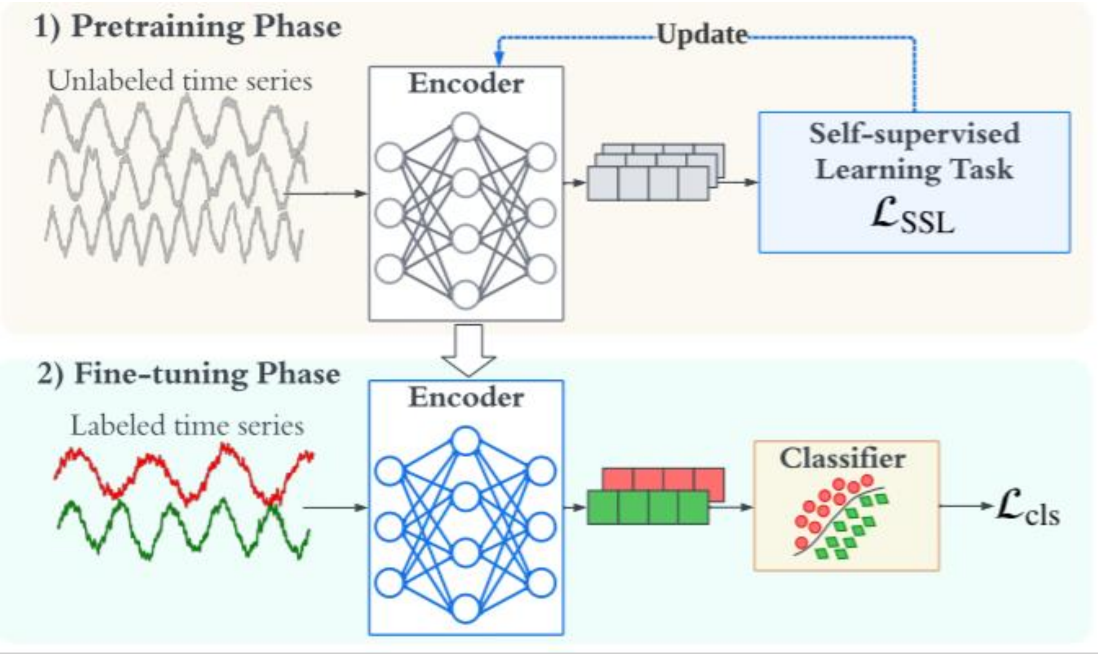
Introduction

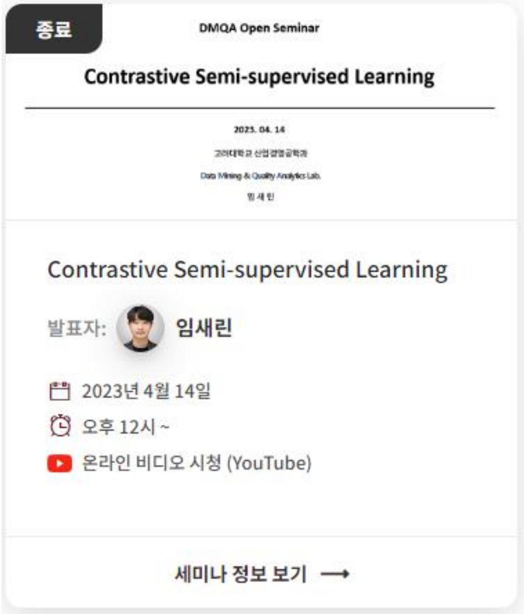
Review on In-domain Representation Learning for Time Series Data



❖ Self-supervised learning

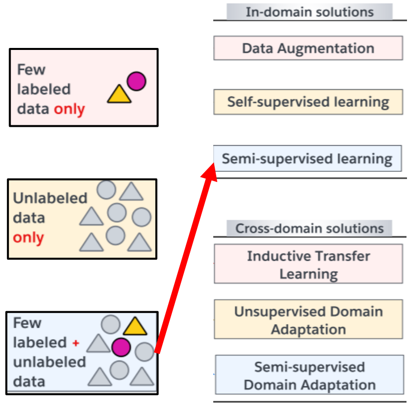
- Pre-training phase 및 Fine-tuning 수행
- 대표적 자가학습 방법론 – Contrastive learning
  - Inter-sample Contrasting : 데이터 증강을 통해 Positive, Negative pair 정의
  - Intra-sample Contrasting : 샘플을 과거, 미래로 분할하여 시간적 관계 학습 (Contrastive Predictive Coding)





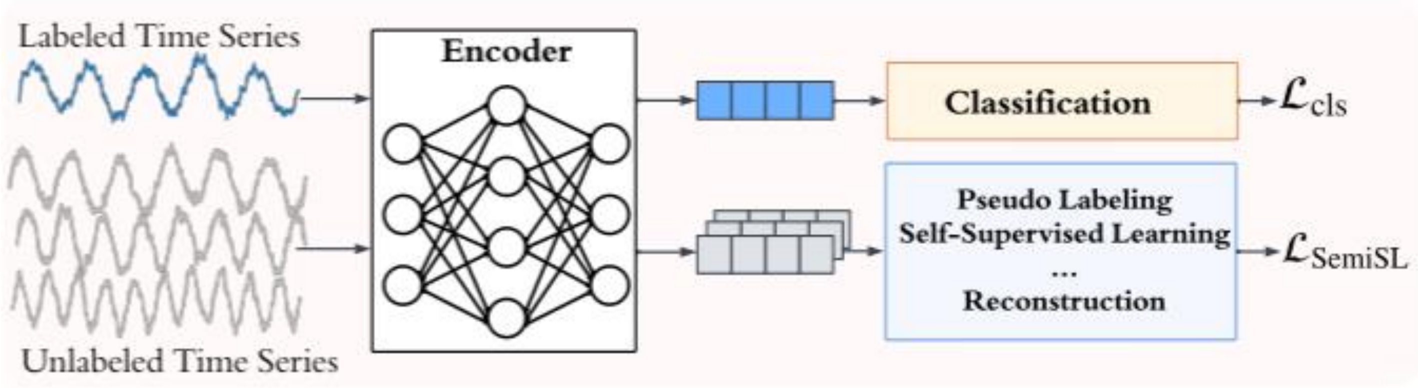
Introduction

Review on In-domain Representation Learning for Time Series Data



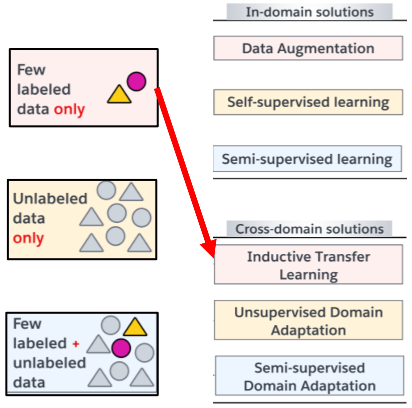
❖ **Semi-supervised Learning**

- Labeled data + Unlabeled data 같이 사용
- Self-training Methods
  - Pseudo labeling : Labeled data로 훈련 → Unlabeled data 의 Label 부여
  - FixMatch like methods : 시계열 데이터 증강 (Weak / Strong) 및 Label 생성
- Auto Encoder
  - Unlabeled data 로 Auto Encoder 학습 하는데 사용



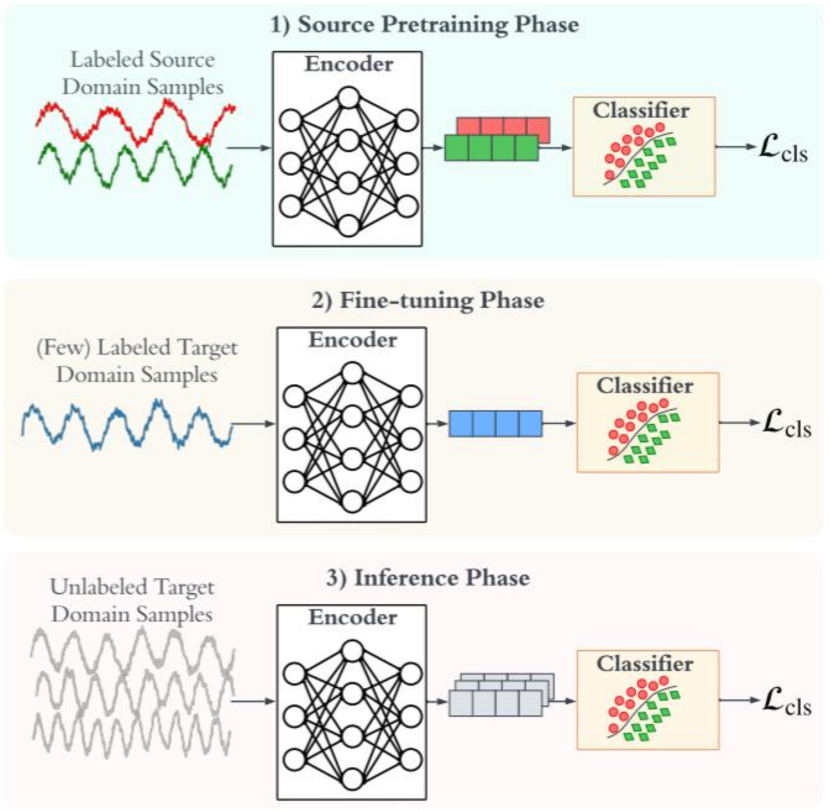
Introduction

Review on Cross-domain Representation Learning for Time Series Data



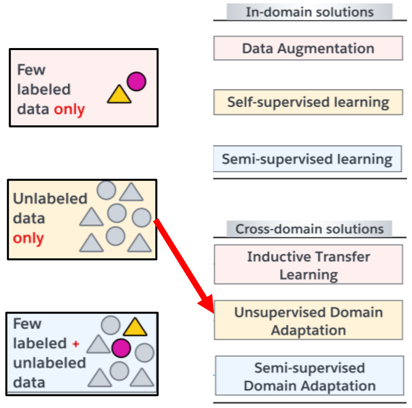
- ❖ **Transfer Learning**
  - Source domain != Target domain
  - Source data 로 Pre-training → Target data 로 Fine-tuning
  - 의료분야와, HAR 분야처럼 같은 Task 이나 Source가 다른 경우 유용함

- ❖ **항상 효과적인 것은 아님 → Source / Target domain 궁합이 중요**
  - IDS : Inter-Datasets Similarity
  - SMS : Source Model Selection
  - 2장에서 자세히 다룰 예정



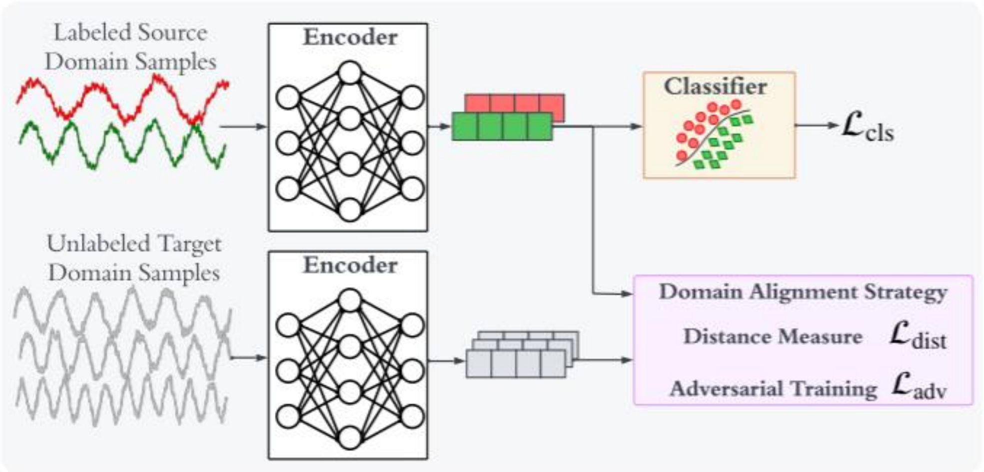
Introduction

Review on Cross-domain Representation Learning for Time Series Data



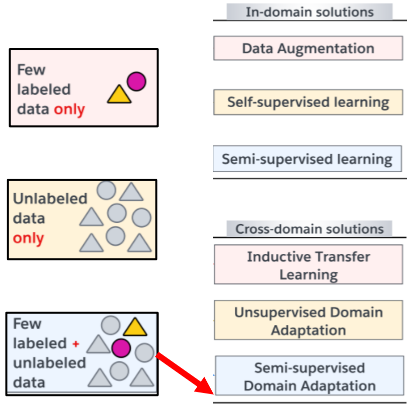
❖ **Unsupervised Domain Adaptation**

- Transfer learning 의 하위개념으로 Target domain 의 label이 없어 Fine tuning 이 불가 한 경우
- Distance-based Methods
  - Maximum Mean Discrepancy(MMD), Correlation Alignment(CORAL) 등 을 Loss Function으로 사용하여 모델의 Source와 Target 분포의 거리를 최소화 함
- Adversarial-based Methods (GAN based)
  - Source-Target 분포 차이가 큰 경우 효과적
  - GAN의 Discriminator network이 Source와 Target을 구분할 수 없도록 훈련



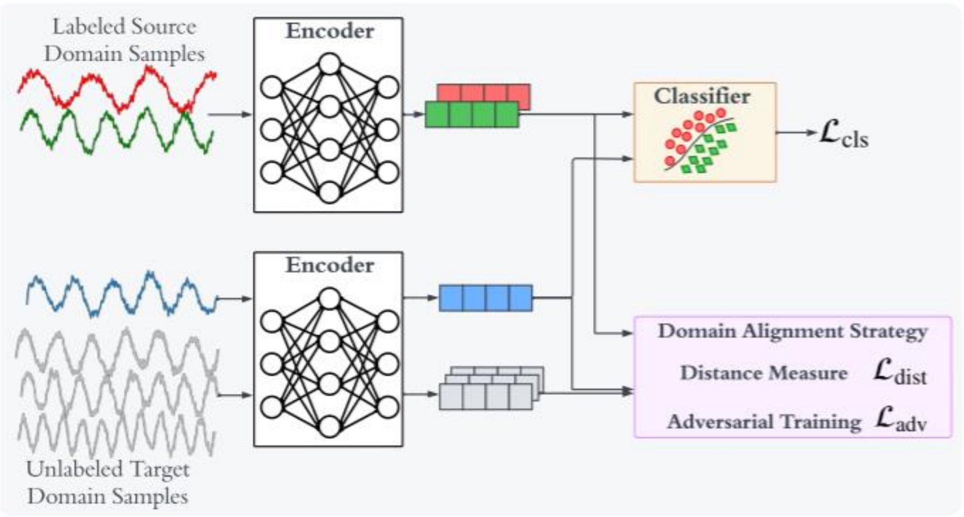
Introduction

Review on Cross-domain Representation Learning for Time Series Data



❖ Semi-supervised Domain Adaptation

- 소량의 Target domain labeled data를 다양하게 활용
  - Classification Loss
  - Pseudo label 생성



Source Selection Methodology in Transfer Learning of Time Series Data

Transfer learning for time series classification

Paper

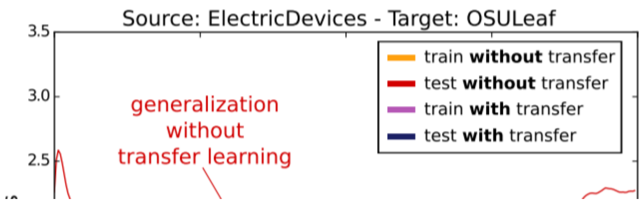
- ❖ Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2018). Transfer learning for time series classification. In 2018 IEEE international conference on big data (263회 인용)
  - **IDS : Inter-Datasets Similarity 개념 제안**

Transfer learning for time series classification

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller
IRIMAS, Université de Haute-Alsace, Mulhouse, France
Email: {first-name.last-name@uha.fr}

***Abstract*—Transfer learning for deep neural networks is the process of first training a base network on a source dataset, and then transferring the learned features (the network's weights) to a second network to be trained on a target dataset. This idea has been shown to improve deep neural network's generalization capabilities in many computer vision tasks such as image recognition and object localization. Apart from these applications, deep Convolutional Neural Networks (CNNs) have also recently gained popularity in the Time Series Classification (TSC) community. However, unlike for image recognition problems, transfer learning techniques have not yet been investigated thoroughly for the TSC task. This is surprising as the accuracy of deep learning models for TSC could potentially be improved if the model is fine-tuned from a pre-trained neural network instead of training it from scratch. In this paper, we fill this gap by investigating how to transfer deep CNNs for the TSC task. To evaluate the potential of transfer learning, we performed extensive experiments using the UCR archive which is the largest publicly available TSC**

problem is known to be mitigated using several techniques such as regularization, data augmentation or simply collecting more data [4], [5]. Another well-know technique is transfer learning [6], where a model trained on a source task is then fine-tuned on a target dataset. For example in Fig. 1, we trained a model on the ElectricDevices dataset [1] and then fine-tuned this same model on the OSULeaf dataset [1], which significantly improved the network's generalization capability.

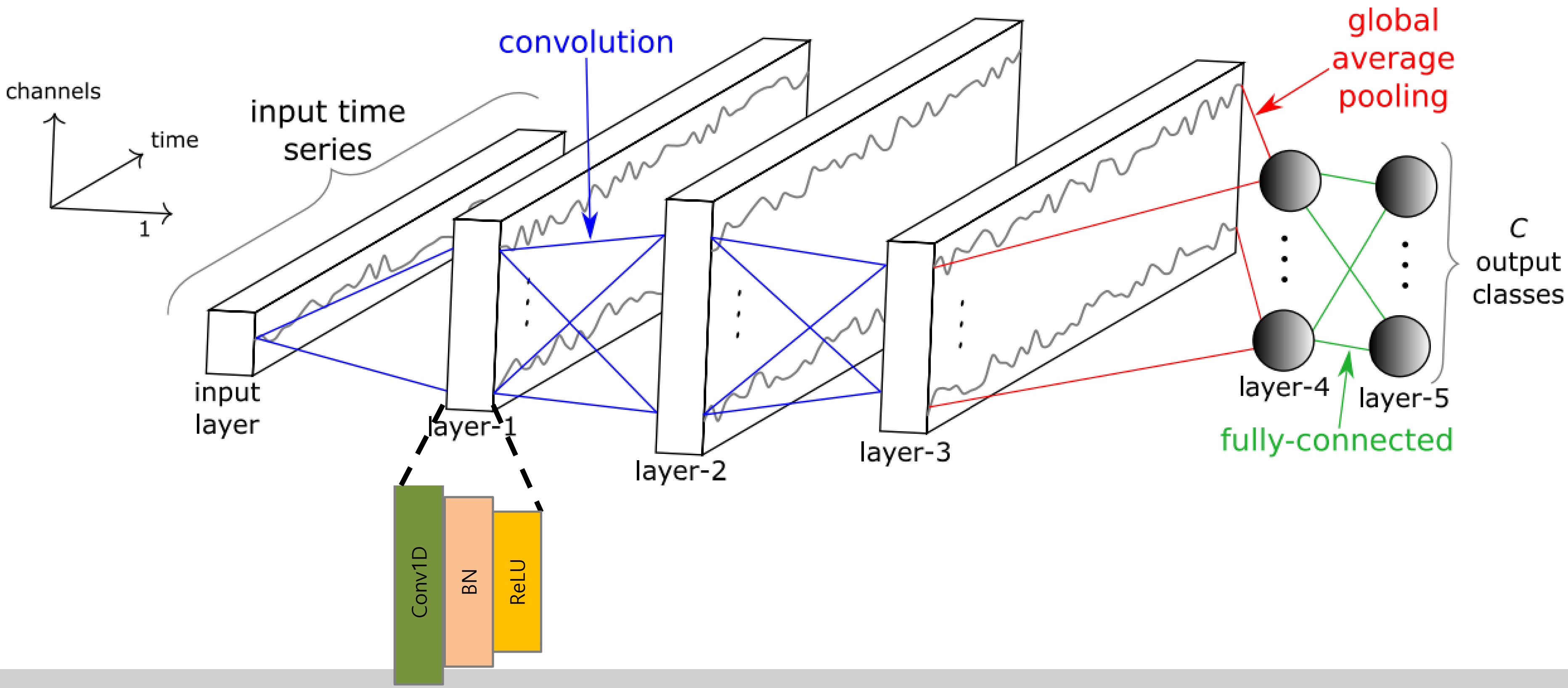


Transfer learning for time series classification

Method

❖ Architecture

- Transfer learning 의 Pre-Train을 위한 모델로 1D Fully Convolutional Network 사용
- Input 은 단변량 시계열 / Output은 Class 확률 분포

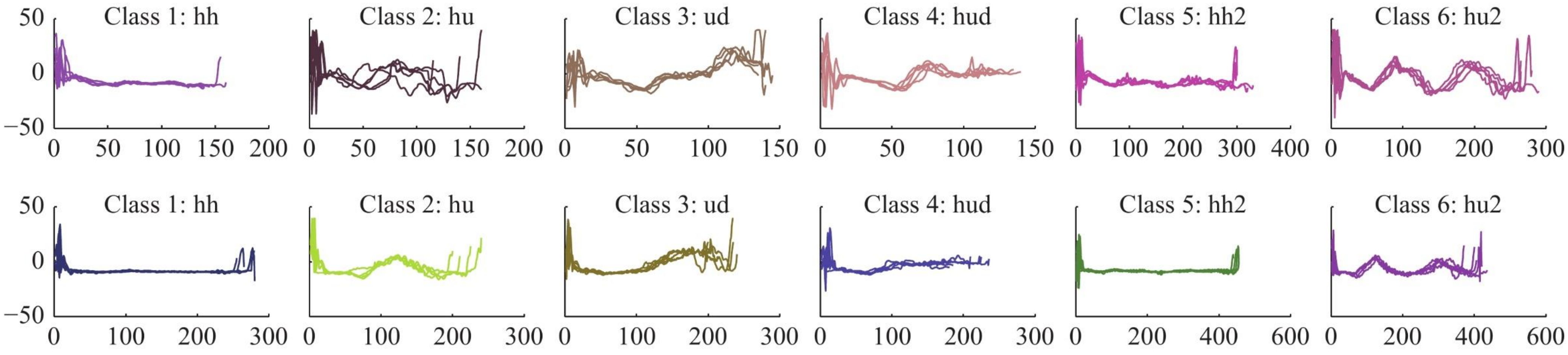


Transfer learning for time series classification

Method

❖ **Dataset**

- UCR Time series classification benchmark
- 85개의 데이터 셋 존재
- 단변량, 다변량 시계열 데이터의 클래스 레이블 존재
- Train data, Test data 이 분리 되어 있음
- 모든 데이터가 정규화 되어있음

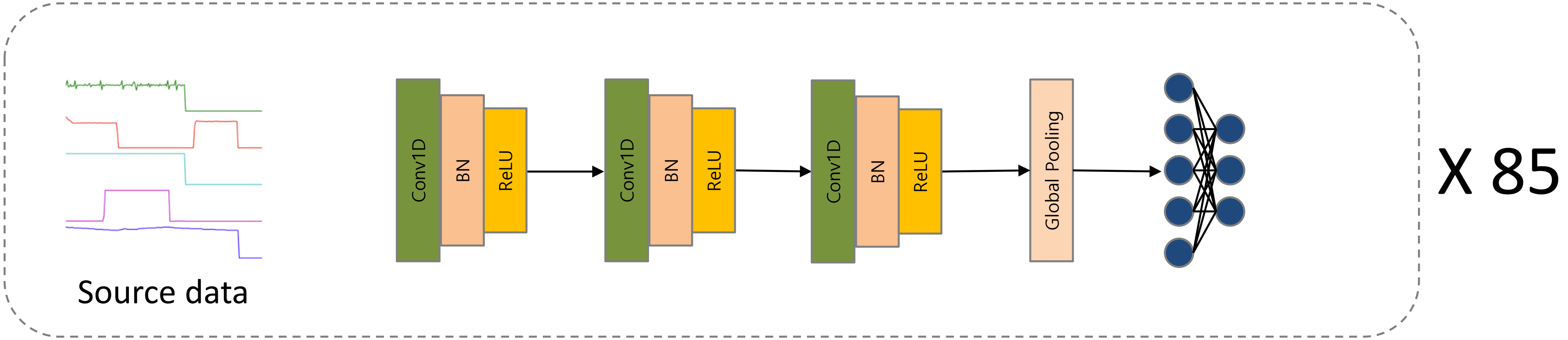


Transfer learning for time series classification

Method

❖ **Network Adaptation (Transfer)**

- UCR Archive 의 85개 데이터셋 각각 모델 훈련 (Pre-train)
- 85개의 Pre-Trained model 아키텍쳐는 출력 계층 (Class 개수에 따라 다름) 외 동일

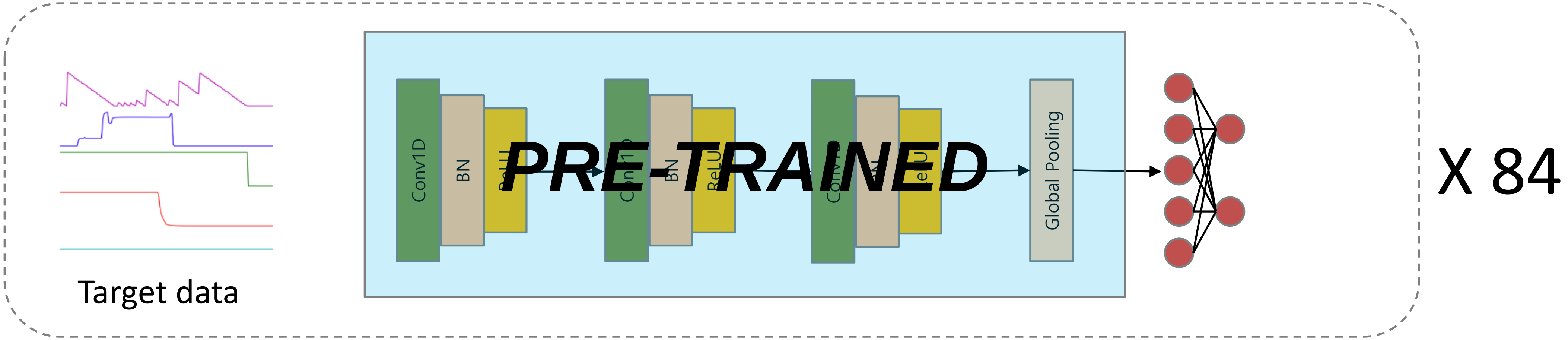


Transfer learning for time series classification

Method

❖ **Network Adapdation (Transfer)**

- 개별 Target 데이터 셋은 84(85 – 1)개의 Pre-trained 모델에 대해 Fine-tuning
- Fine-tuning 시 Pre-train 모델에서 Fully Connected Layer 제거 후 Target 데이터 클래스 맞게 교체
- Fine-tuning 시 교체 된 Fully Connected Layer 만 학습 시 수렴 안함 → 전체 네트워크 재 훈련 수행

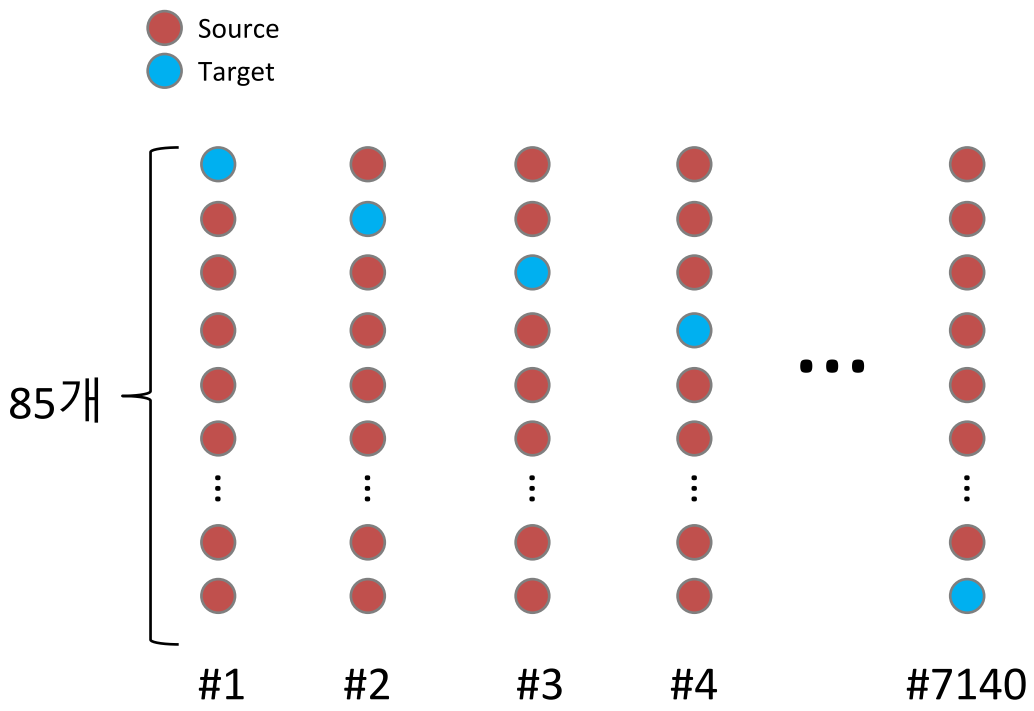


Transfer learning for time series classification

Method

❖ **Source Data Set 선정의 어려움**

- UCR Archive 내 85개의 데이터 셋 존재 → 각 Target 데이터 셋에 대해 84개의 Source 데이터 셋 존재
- 모든 Case 실험을 위해 7140 (85 X 84) 번의 실험이 필요 → 매우 비 효율적
- 실제 상황에서 위처럼 최적의 Source data set 추정 할 수 없음

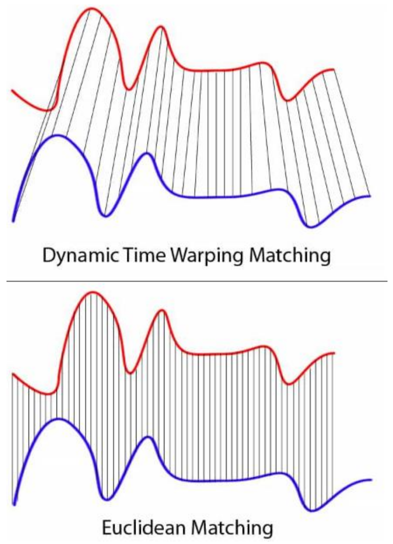


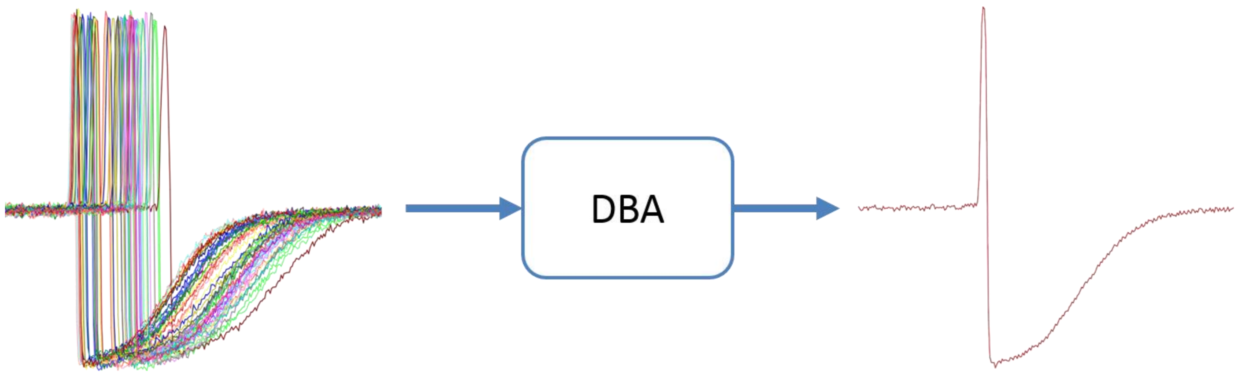
Transfer learning for time series classification

Method

❖ **Source Data Set 선정 Idea**

- 데이터셋 간의 유사성(IDS; Inter Datasets Similarity) 을 계산하여 최적의 소스 데이터셋을 선택
- 각 데이터셋의 Class 별 시계열 → Class 당 하나의 시계열로 변형 (by DTW Barycenter Averaging)
- UCR 데이터 셋 (85개) 간 유사성 행렬 생성 및 데이터 셋 간 거리 계산





Transfer learning for time series classification

Method

❖ **Dataset 간 거리 행렬 계산 알고리즘**

- 데이터 축소 단계 (lines 1-7):
  - 각 데이터셋의 클래스를 순회하며
    각 클래스의 시계열 세트를 DBA를 통해 평균화
- 거리 계산 단계 (lines 8-22):
  - 모든 가능한 데이터셋 쌍의 조합을 순회 (lines 8-10)
  - 각 데이터셋의 각 클래스를 순회 (lines 13 and 14)
- 두 데이터셋 간의 거리는 해당 클래스 간의 최소 DTW 거리로 설정 (lines 15-19)

Algorithm 1 Inter-datasets similarity

Input: N time series datasets in an array D
Output: $N \times N$ datasets similarity matrix
Initialization : matrix M of size $N \times N$
data reduction step
1: **for** $i = 1$ to N **do**
2: $\quad C = D[i].classes$
3: $\quad$ **for** $c = 1$ to $length(C)$ **do**
4: $\qquad avg\_init = medoid(C[c])$
5: $\qquad C[c] = DBA(C[c], avg\_init)$
6: $\quad$ **end for**
7: **end for**
distance calculation step
8: **for** $i = 1$ to N **do**
9: $\quad C_i = D[i].classes$
10: $\quad$ **for** $j = 1$ to N **do**
11: $\qquad C_j = D[j].classes$
12: $\qquad dist = \infty$
13: $\qquad$ **for** $c_i = 1$ to $length(C_i)$ **do**
14: $\qquad\quad$ **for** $c_j = 1$ to $length(C_j)$ **do**
15: $\qquad\qquad cdist = DTW(C_i[c_i], C_j[c_j])$
16: $\qquad\qquad dist = minimum(dist, cdist)$
17: $\qquad\quad$ **end for**
18: $\qquad$ **end for**
19: $\qquad M[i, j] = dist$
20: $\quad$ **end for**
21: **end for**
22: **return** M

Transfer learning for time series classification

Method

❖ **Dataset 간 거리 행렬 계산 알고리즘**

- 데이터 축소 단계 (lines 1-7):
  - 각 데이터셋의 클래스를 순회하며
    각 클래스의 시계열 세트를 DBA를 통해 평균화
- 거리 계산 단계 (lines 8-22):
  - 모든 가능한 데이터셋 쌍의 조합을 순회 (lines 8-10)
  - 각 데이터셋의 각 클래스를 순회 (lines 13 and 14)
- 두 데이터셋 간의 거리는 해당 클래스 간의 최소 DTW 거리로 설정 (lines 15-19)

Algorithm 1 Inter-datasets similarity

Input: N time series datasets in an array D

Output: $N \times N$ datasets similarity matrix

```
    Initialization : matrix M of size N × N
    data reduction step
 1: for i = 1 to N do
 2:   C = D[i].classes
 3:   for c = 1 to length(C) do
 4:     avg_init = medoid(C[c])
 5:     C[c] = DBA(C[c], avg_init)
 6:   end for
 7: end for
    distance calculation step
 8: for i = 1 to N do
 9:   C_i = D[i].classes
10:   for j = 1 to N do
11:     C_j = D[j].classes
12:     dist = ∞
13:     for c_i = 1 to length(C_i) do
14:       for c_j = 1 to length(C_j) do
15:         cdist = DTW(C_i[c_i], C_j[c_j])
16:         dist = minimum(dist, cdist)
17:       end for
18:     end for
19:     M[i, j] = dist
20:   end for
21: end for
22: return M
```

Transfer learning for time series classification

Method

❖ **Dataset 간 거리 행렬 계산 알고리즘**

- 데이터 축소 단계 (lines 1-7):
  - 각 데이터셋의 클래스를 순회하며
    각 클래스의 시계열 세트를 DBA를 통해 평균화
- 거리 계산 단계 (lines 8-22):
  - 모든 가능한 데이터셋 쌍의 조합을 순회 (lines 8-10)
  - 각 데이터셋의 각 클래스를 순회 (lines 13 and 14)
- 두 데이터셋 간의 거리는 해당 클래스 간의 최소 DTW 거리로 설정 (lines 15-19)

Algorithm 1 Inter-datasets similarity

Input: N time series datasets in an array D

Output: $N \times N$ datasets similarity matrix

Initialization : matrix M of size $N \times N$

data reduction step

1: **for** $i = 1$ to N **do**
2:   $C = D[i].classes$
3:   **for** $c = 1$ to $length(C)$ **do**
4:     $avg\_init = medoid(C[c])$
5:     $C[c] = DBA(C[c], avg\_init)$
6:   **end for**
7: **end for**

distance calculation step

8: **for** $i = 1$ to N **do**
9:   $C_i = D[i].classes$
10:   **for** $j = 1$ to N **do**
11:     $C_j = D[j].classes$
12:     $dist = \infty$
13:     **for** $c_i = 1$ to $length(C_i)$ **do**
14:       **for** $c_j = 1$ to $length(C_j)$ **do**
15:         $cdist = DTW(C_i[c_i], C_j[c_j])$
16:         $dist = minimum(dist, cdist)$
17:       **end for**
18:     **end for**
19:     $M[i, j] = dist$
20:   **end for**
21: **end for**
22: **return** M

Transfer learning for time series classification

Method

❖ **Dataset 간 거리 행렬 계산 알고리즘**

- 데이터 축소 단계 (lines 1-7):
  - 각 데이터셋의 클래스를 순회하며

    각 클래스의 시계열 세트를 DBA를 통해 평균화
- 거리 계산 단계 (lines 8-22):
  - 모든 가능한 데이터셋 쌍의 조합을 순회 (lines 8-10)
  - 각 데이터셋의 각 클래스를 순회 (lines 13 and 14)
- 두 데이터셋 간의 거리는 해당 클래스 간의 최소 DTW 거리로 설정 (lines 15-19)

Algorithm 1 Inter-datasets similarity

Input: N time series datasets in an array D

Output: $N \times N$ datasets similarity matrix

Initialization : matrix M of size $N \times N$

data reduction step

1: **for** $i = 1$ to N **do**
2:   $C = D[i].classes$
3:   **for** $c = 1$ to $length(C)$ **do**
4:     $avg\_init = medoid(C[c])$
5:     $C[c] = DBA(C[c], avg\_init)$
6:   **end for**
7: **end for**

distance calculation step

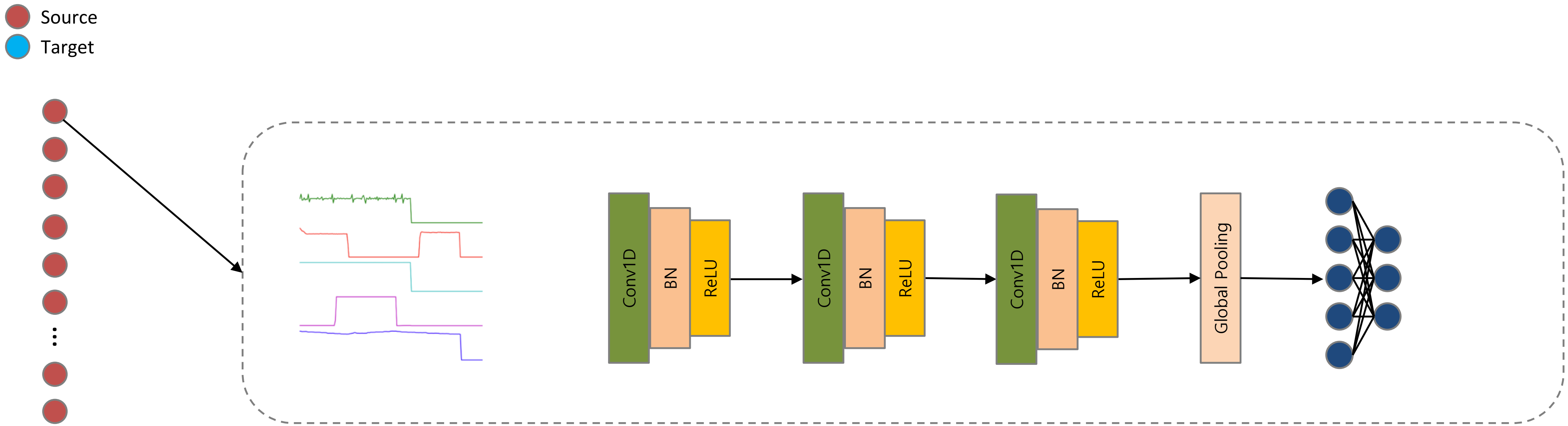
8: **for** $i = 1$ to N **do**
9:   $C_i = D[i].classes$
10:   **for** $j = 1$ to N **do**
11:     $C_j = D[j].classes$
12:     $dist = \infty$
13:     **for** $c_i = 1$ to $length(C_i)$ **do**
14:       **for** $c_j = 1$ to $length(C_j)$ **do**
15:         $cdist = DTW(C_i[c_i], C_j[c_j])$
16:         $dist = minimum(dist, cdist)$
17:       **end for**
18:     **end for**
19:     $M[i, j] = dist$
20:   **end for**
21: **end for**
22: **return** M

Transfer learning for time series classification

Experiments

❖ **Brute-force approach**

- 모든 데이터 셋(85개) 에 대해 개별 Pre-Trained 모델 학습

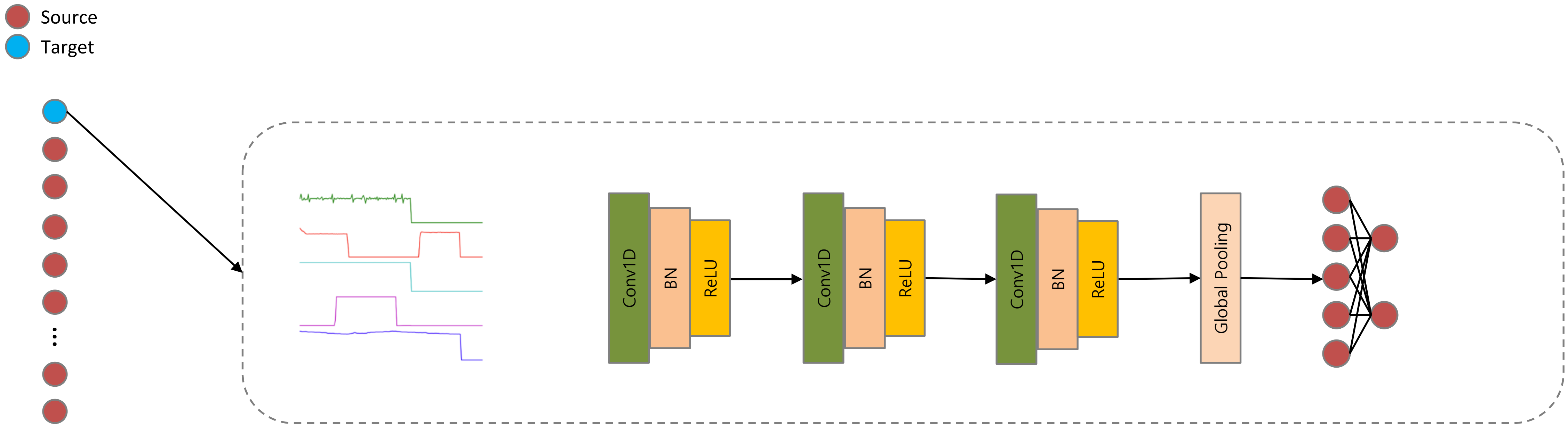


Transfer learning for time series classification

Experiments

❖ Brute-force approach

- 타겟 데이터셋 하나 당 84개의 Pre-Trained 된 모델을 Fine-Tuning
- 85개의 타겟 데이터셋 X 84개의 Pre-Trained 모델 존재 = 7140회 실험 필요

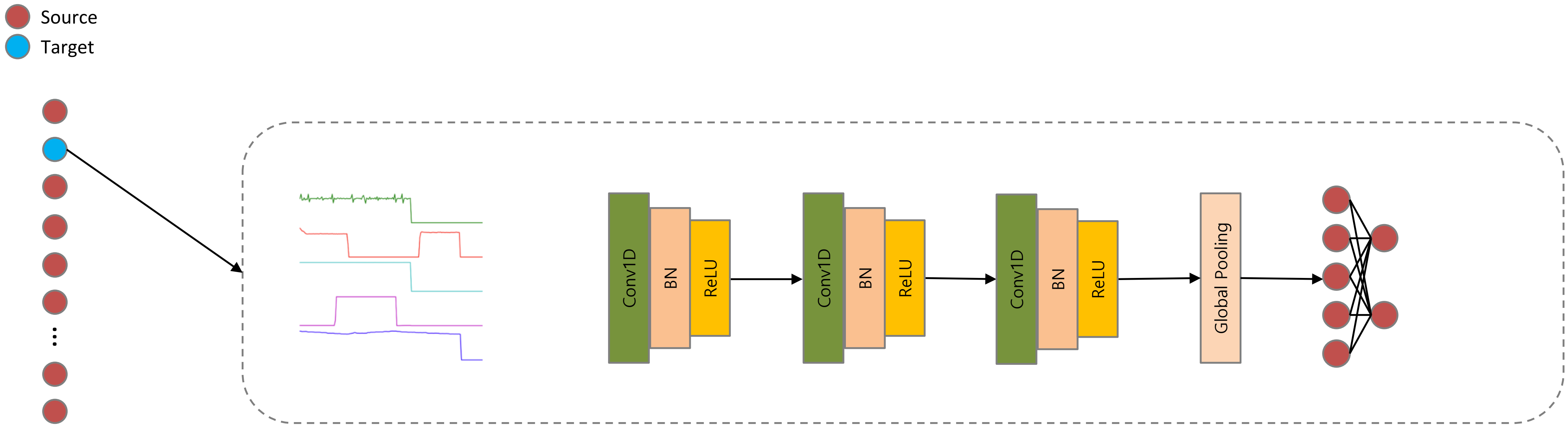


Transfer learning for time series classification

Experiments

❖ **Brute-force approach**

- 타겟 데이터셋 하나 당 84개의 Pre-Trained 된 모델을 Fine-Tuning
- 85개의 타겟 데이터셋 X 84개의 Pre-Trained 모델 존재 = 7140회 실험 필요

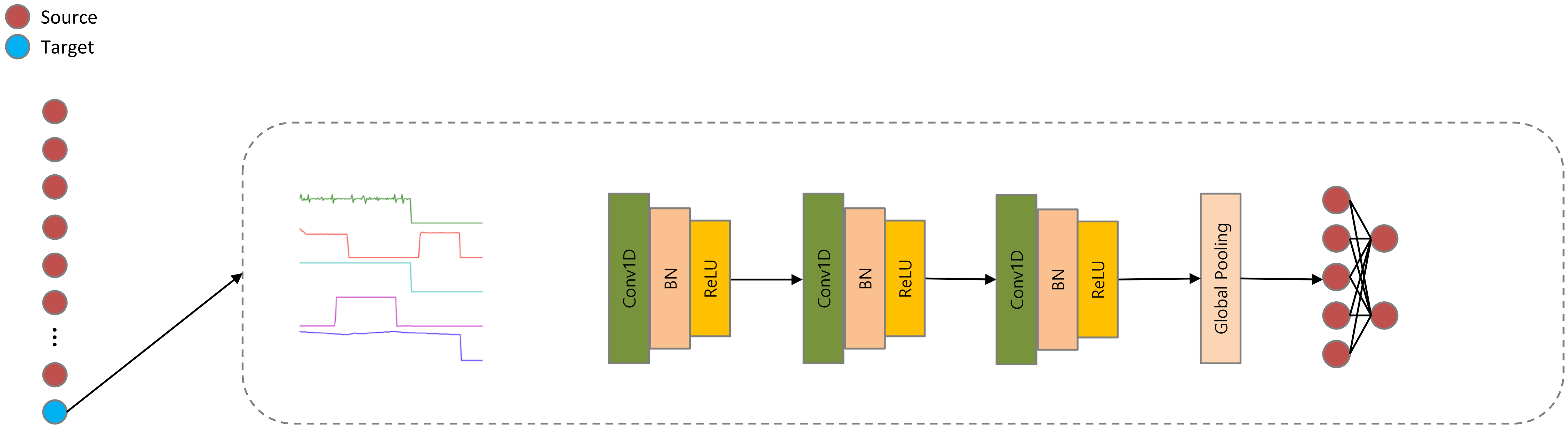


Transfer learning for time series classification

Experiments

❖ Brute-force approach

- 타겟 데이터셋 하나 당 84개의 Pre-Trained 된 모델을 Fine-Tuning
- 85개의 타겟 데이터셋 X 84개의 Pre-Trained 모델 존재 = 7140회 실험 필요

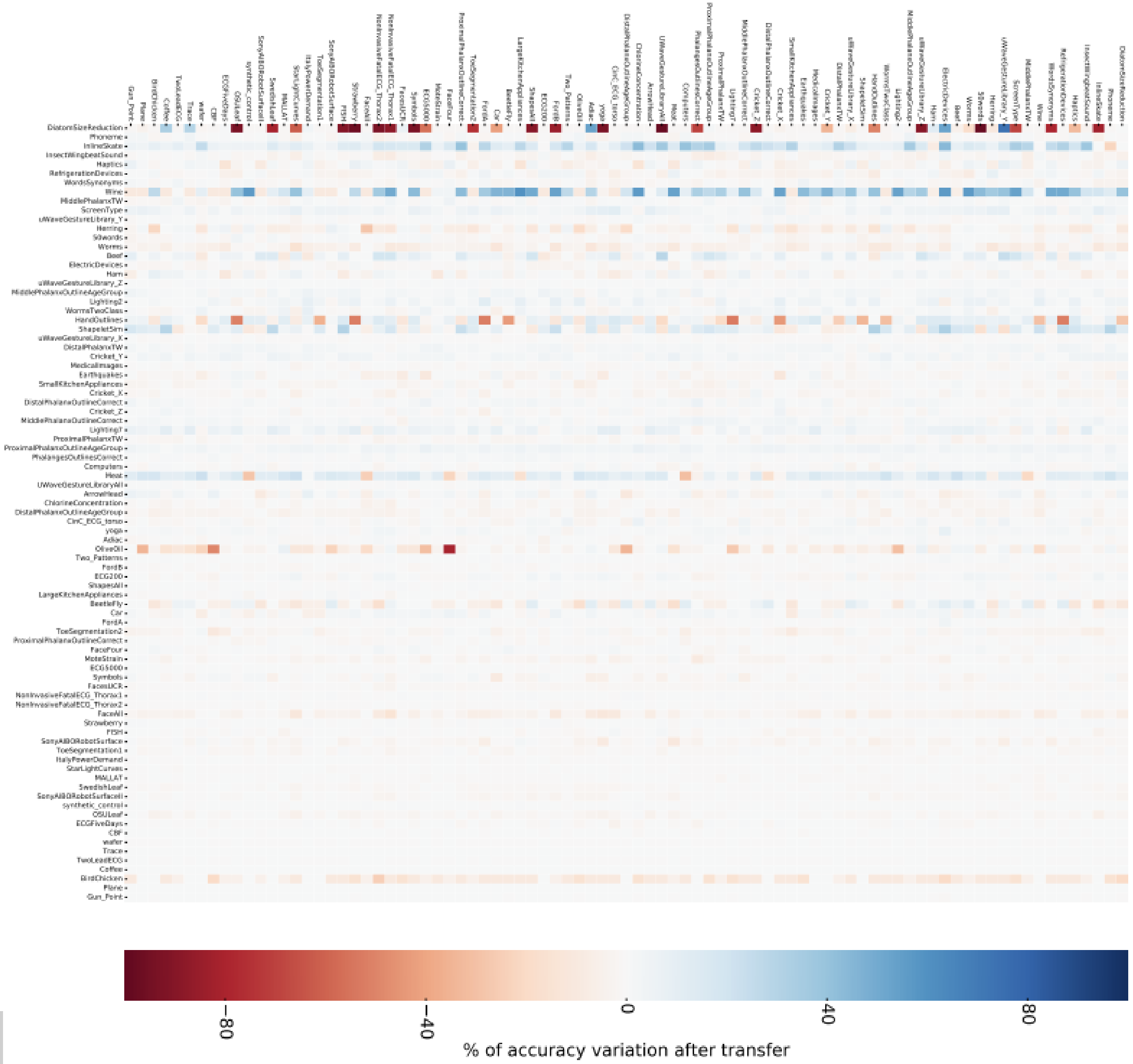


Transfer learning for time series classification

Experiments

- ❖ **Brute-force approach result**
  - ▪ 모델 Accuracy 변화율을 Heatmap 화
    - 행 : Source dataset
    - 열 : Target dataset
    - 빨간색 : Source이 Target 에 부정적 영향
    - 파란색 : Source이 Target 에 긍정적 영향
    - 회색 : Transfer learning 사용/미사용 차이 없음

- ❖ **Source / Target dataset에 대한 유사도를 정량화 후 Transfer learning 적용 여부 결정 필요**

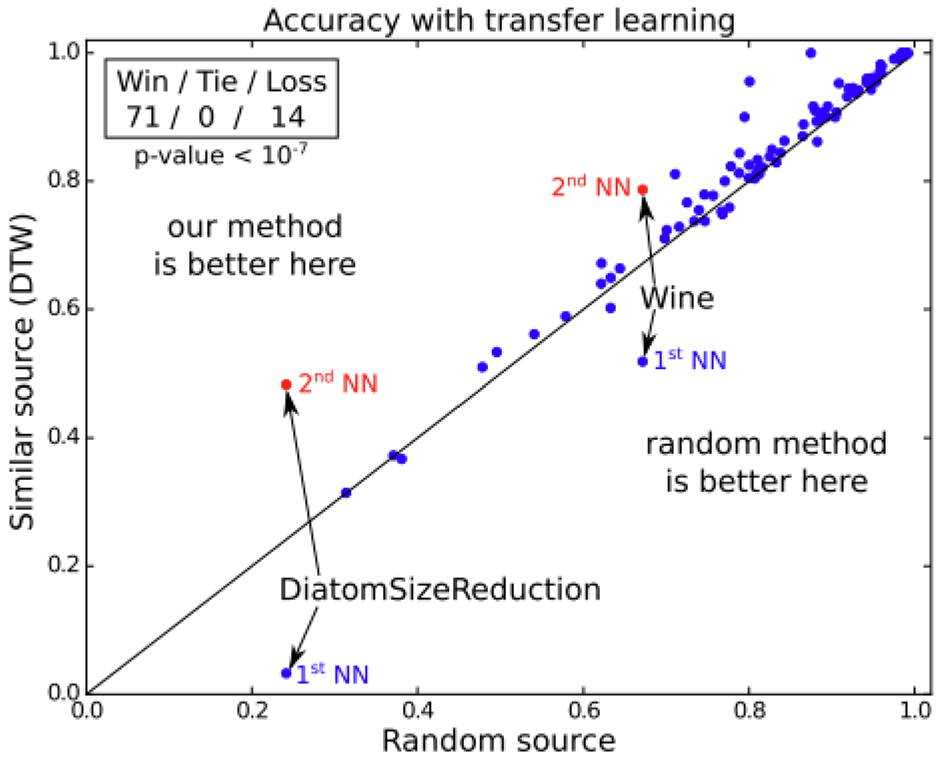


Transfer learning for time series classification

Experiments

❖ **IDS : Inter-Datasets Similarity 적용 전/후 Accuracy 비교**

- Source dataset을 Random 으로 선택 한 경우 대비 IDS 최소값을 갖는 Dataset 을 Source로 선택 한 경우 Accuracy 가 대부분 증가 함

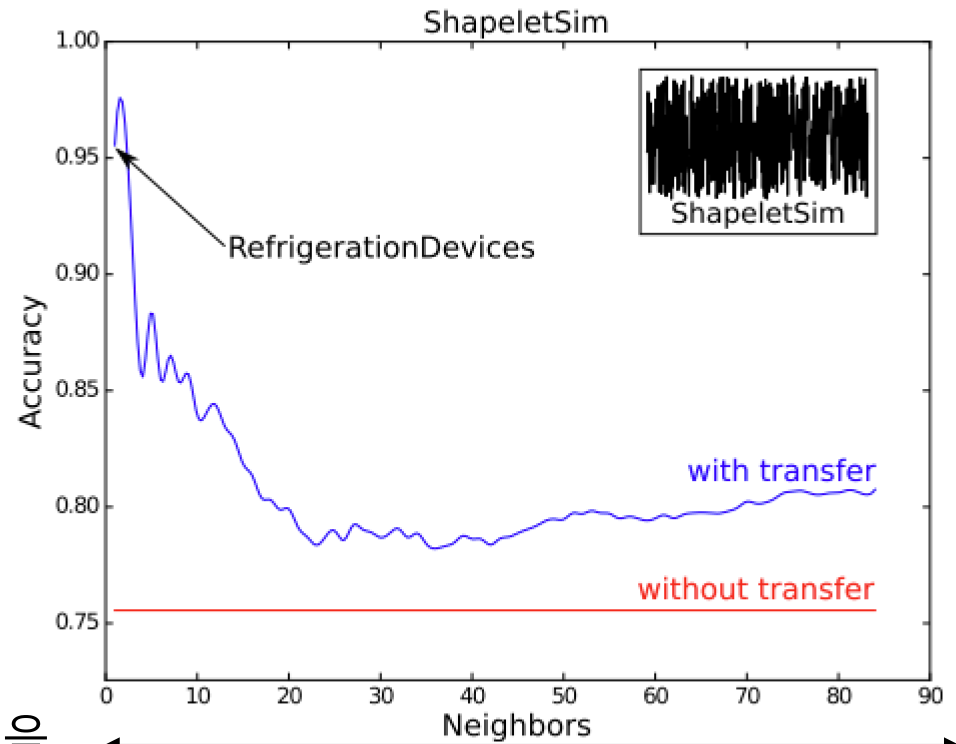


Transfer learning for time series classification

Experiments

❖ **개별 데이터셋 탐구 (Target Dataset : ShapletSim)**

- 가장 IDS 낮은 Source dataset 은 Refrigeration Devices
- 해당 Source로 Transfer learning 시 Accuracy 증가 (76% → 93%)



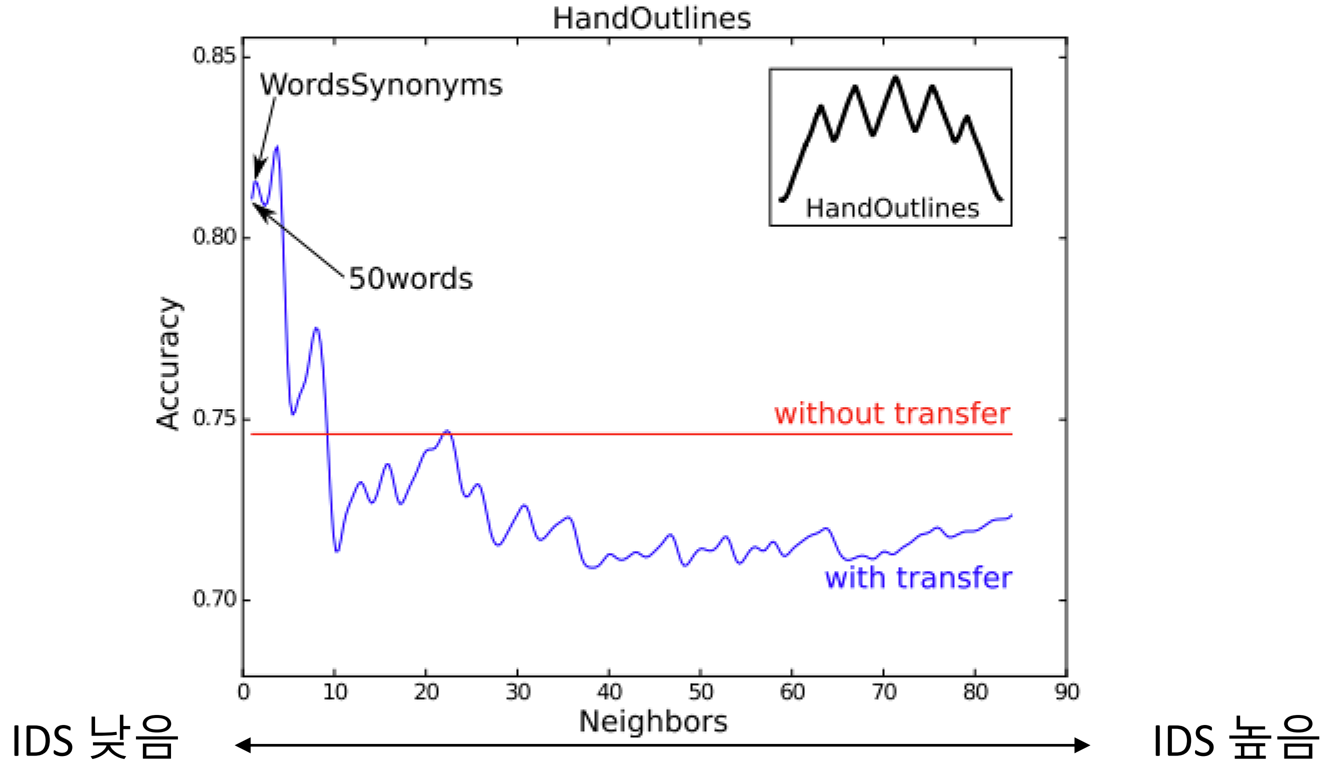
IDS 낮음 ← → IDS 높음

Transfer learning for time series classification

Experiments

❖ **개별 데이터셋 탐구 (Target Dataset : HandOutlines)**

- 가장 IDS 낮은 Source dataset 은 WordsSynonyms
- Random dataset으로 Transfer learning 수행 시 대부분 성능 감소하나 최적의 Source dataset을 사용시 성능 향상 확인

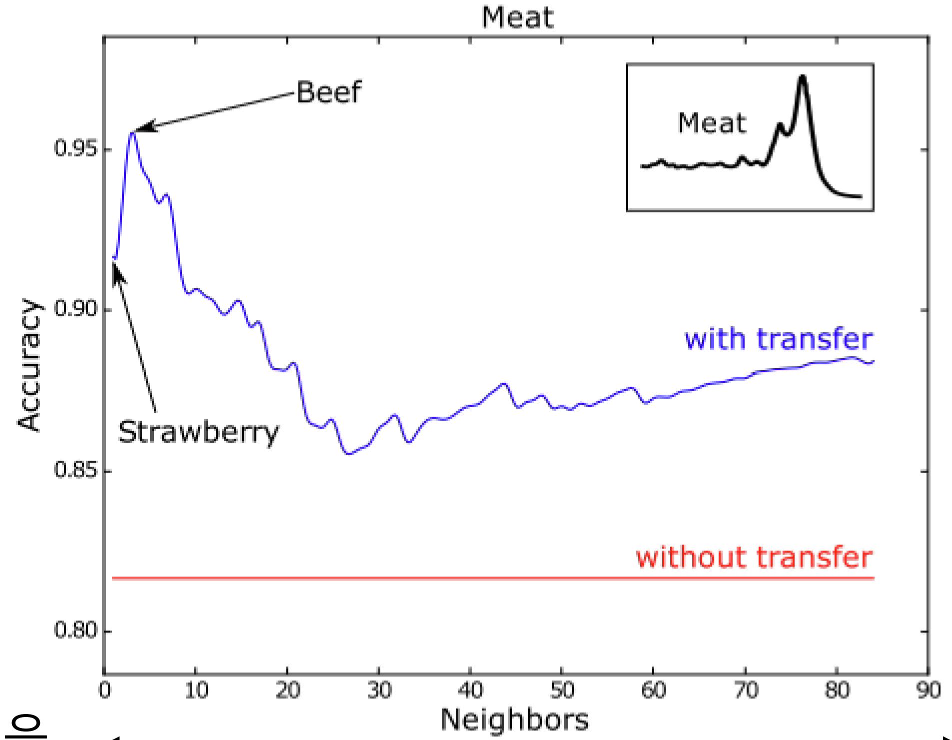


Transfer learning for time series classification

Experiments

❖ **개별 데이터셋 탐구 (Target Dataset : Meat)**

- Train dataset이 20개로 Transfer learning 이 대부분 유리 함
- 마찬가지로 IDS 가 작은 Source dataset으로 Transfer learning 수행 시 더 높은 Accuracy 달성



IDS 낮음 ⟷ IDS 높음

Transfer learning for time series classification

Summarization

❖ **Conclusion**

- 시계열 분류 문제에 대한 전이학습시 Source dataset 선정에 따른 성능 영향 최초 연구
- 잘못된 소스 데이터셋을 선택 시 모델이 성능이 저하 될 수 있음
- DTW를 사용하여 소스 데이터셋과 타겟 데이터셋 간의 유사도(IDS; Inter-Datasets Similarity) 계산 방법론 제안
- IDS기반 Target dataset에 최적인 Source dataset을 찾아냄으로써 효과적인 Transfer learning 가능성 제시

Source Model Selection for Deep Learning in the Time Series Domain

Paper

❖ Meiseles, A., & Rokach, L. (2020). Source model selection for deep learning in the time series domain. IEEE Access (31회 인용)

- **SMS : Source Model Selection 개념 제안**

Source Model Selection for Deep Learning in the Time Series Domain

AMIEL MEISELES AND LIOR ROKACH
Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Be'er Sheva 8410501, Israel
Corresponding author: Amiel Meiseles (amielm@post.bgu.ac.il)

ABSTRACT Transfer Learning aims to transfer knowledge from a source task to a target task. We focus on a situation when there is a large number of available source models, and we are interested in choosing a single source model that can maximize the predictive performance in the target domain. Existing methods compute some form of "similarity" between the source task data and the target task data. They then select the most similar source task and use the model trained on it for transfer learning. Previous methods do not account for the fact that it is the model parameters that are transferred rather than the data. Therefore, the "similarity" of the source data does not directly influence transfer learning performance. In addition, we would like the possibility of confidently selecting a source model even when the data it was trained on is not available, for example, due to privacy or copyright constraints. We propose to use the truncated source models as encoders for the target data. We then select a source model based on how well it clusters the target data in the latent encoding space, which we calculate using the Mean Silhouette Coefficient. We prove that if the encodings achieve a Mean Silhouette Coefficient of 1, optimal classification can be achieved using just the final layer of the target network. We evaluate our method using the University of California, Riverside (UCR) time series archive and show that the proposed method achieves comparable results to previous work, without using the source data.

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **기존 IDS(Inter-Datasets Similarity)의 문제점**

- Transfer learning 에서 전이 되는 것은 데이터가 아닌 Model의 Parameter임
- 따라서, Source dataset 과 Target dataset 의 유사성은 학습 성능에 직접적인 영향을 미치지 않음
- 현실적으로, Transfer learning 시 사용된 Source dataset을 접근 할 수 없는 경우가 많음

❖ **본 연구에서는**

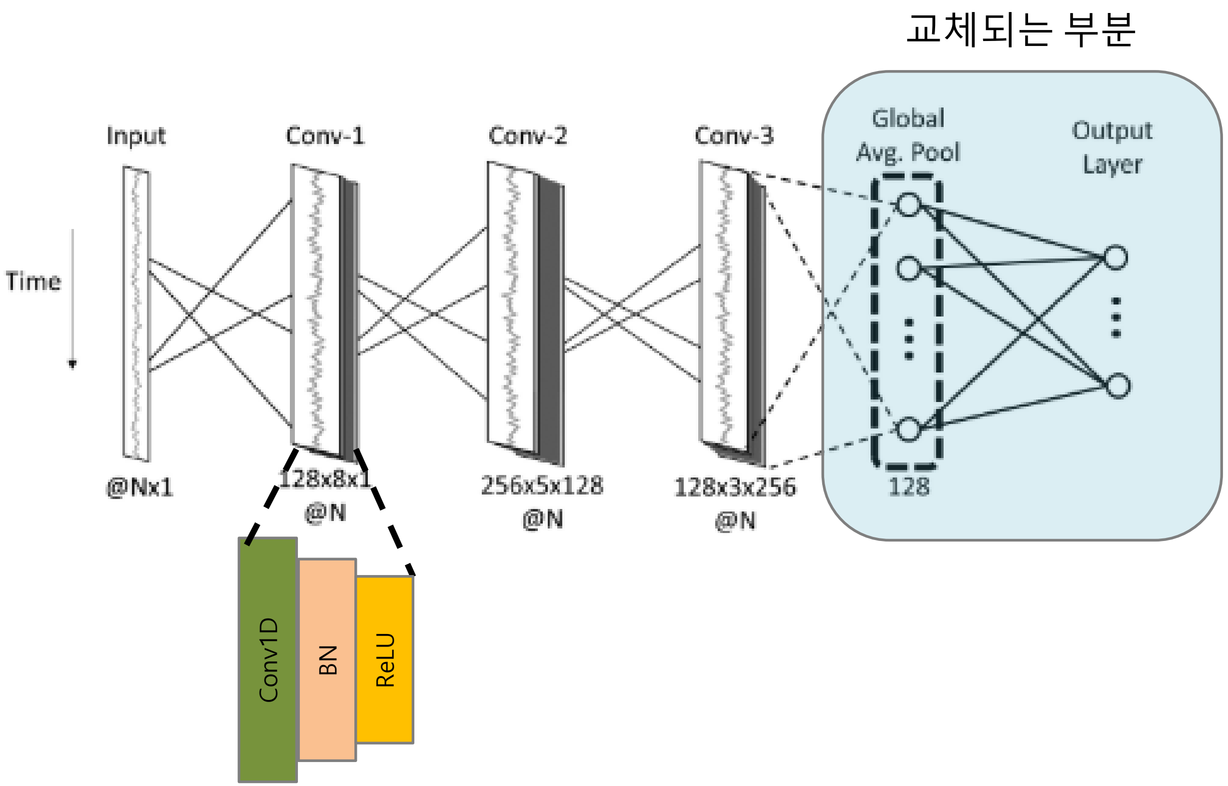
- Target 데이터와 Pre-trained 모델만 사용하여 최적의 소스 도메인 선택 방법론 제시
- Source-Target간 Transfer learning 성능을 MSC(Mean Silhouette Coefficient)통해 정량화

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Architecture**

- 이전 연구와 동일한 모델 구조로 진행 (Fawaz et al.)
- Pre-training 후 Target dataset의 Class 개수에 맞는 Fully Connect Layer로 교체 후 Fine-Tuning

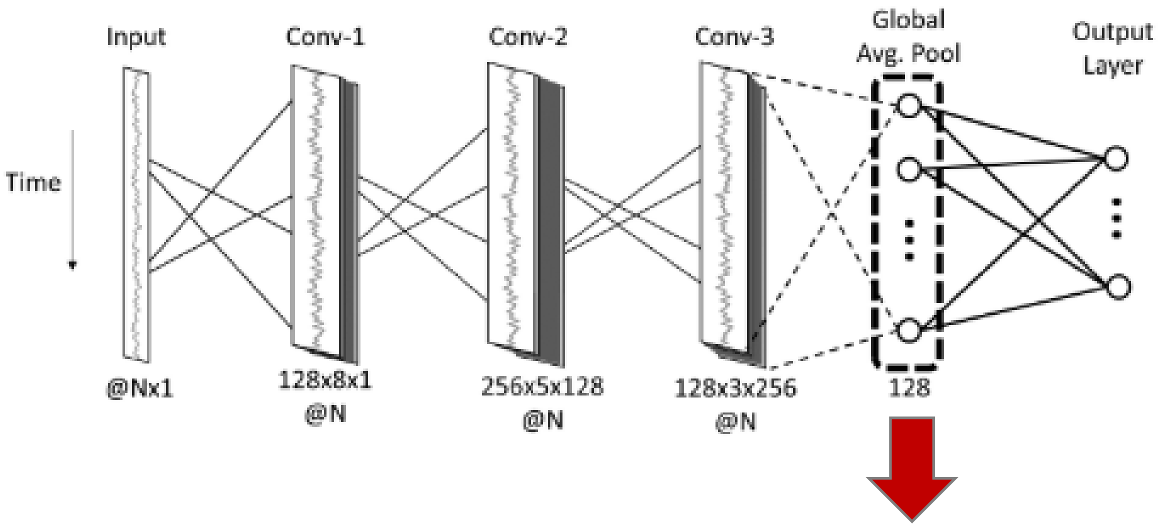


Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차**

- 각 Pre-Trained 모델에 Target 데이터를 Global Pooling layer 의 Output 형태로 인코딩



Encoded Input
(128-dimensional Vector)

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차**

- Silhouette Coefficient 계산[(1)]
- 거리 d는 Cosine distance 활용 (Manhattan, Euclidean 도 비교 평가 함)
- 현재 Source dataset의 Pre-Trained model로 인코딩 된 Target data Sample이 같은 Cluster(Class)에 잘 할당 될 수록 1에 가까움

$$\boxed{s(i)} = \frac{b-a}{max(a,b)} \quad (1)$$

where:

$$a = \frac{1}{|A|-1} \sum_{\substack{x \in A, \\ x \neq i}} d(i,x) \quad \text{and} \quad b = \min_{C \neq A} \frac{1}{|C|} \sum_{x \in C} d(i,x)$$

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차**

- Mean Silhouette Coefficient 계산[(2)]
- 현재 Source dataset 의 Pre-Trained model로 인코딩 된 전체 Target data Sample의 Silhouette Coefficient 평균

  → 1에 가까울 수록 현재 Source dataset을 활용한 품질이 좋음

$$\boxed{MSC(E, labels)} = \frac{1}{|E|} \sum_{i \in E} s(i) \qquad (2)$$

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차 예**

- 각 Target dataset의 Sample을 특정 Source Pre-Trained Model을 통해 Encoding

Sample

	Encoding	Label
e_0	1,2,1	c_0
e_1	1,1,2	c_0
e_2	2,2,1	c_1
e_3	3,4,1	c_1
e_4	4,4,3	c_2
e_5	4,1,3	c_2

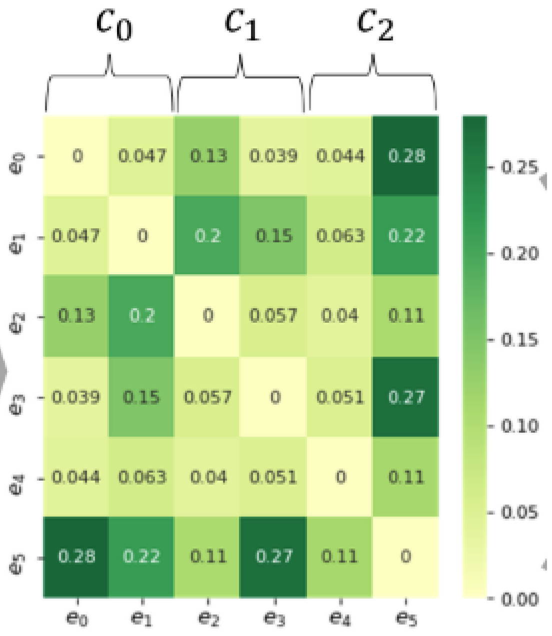
Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ Source Model Selection 절차 예

- Encoding 된 Vector 간 Cosine Distance 계산

	Encoding	Label
e_0	1,2,1	c_0
e_1	1,1,2	c_0
e_2	2,2,1	c_1
e_3	3,4,1	c_1
e_4	4,4,3	c_2
e_5	4,1,3	c_2

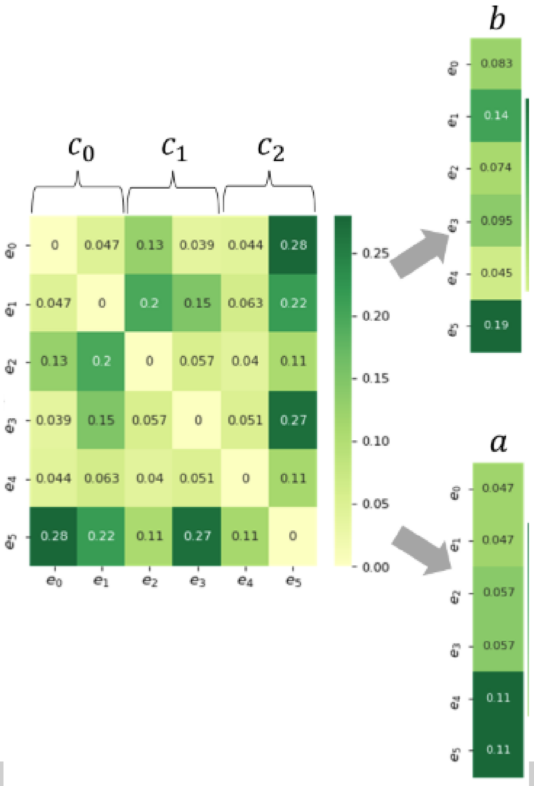


Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ Source Model Selection 절차 예

- Silhouette Coefficient 계산을 위해 a와 b 계산
  - a: 같은 클러스터 내의 다른 모든 데이터 포인트와의 평균 거리
  - b: 가장 가까운 클러스터의 모든 데이터 포인트와의 평균 거리



$$s(i) = \frac{b - a}{max(a, b)} \tag{1}$$

where:

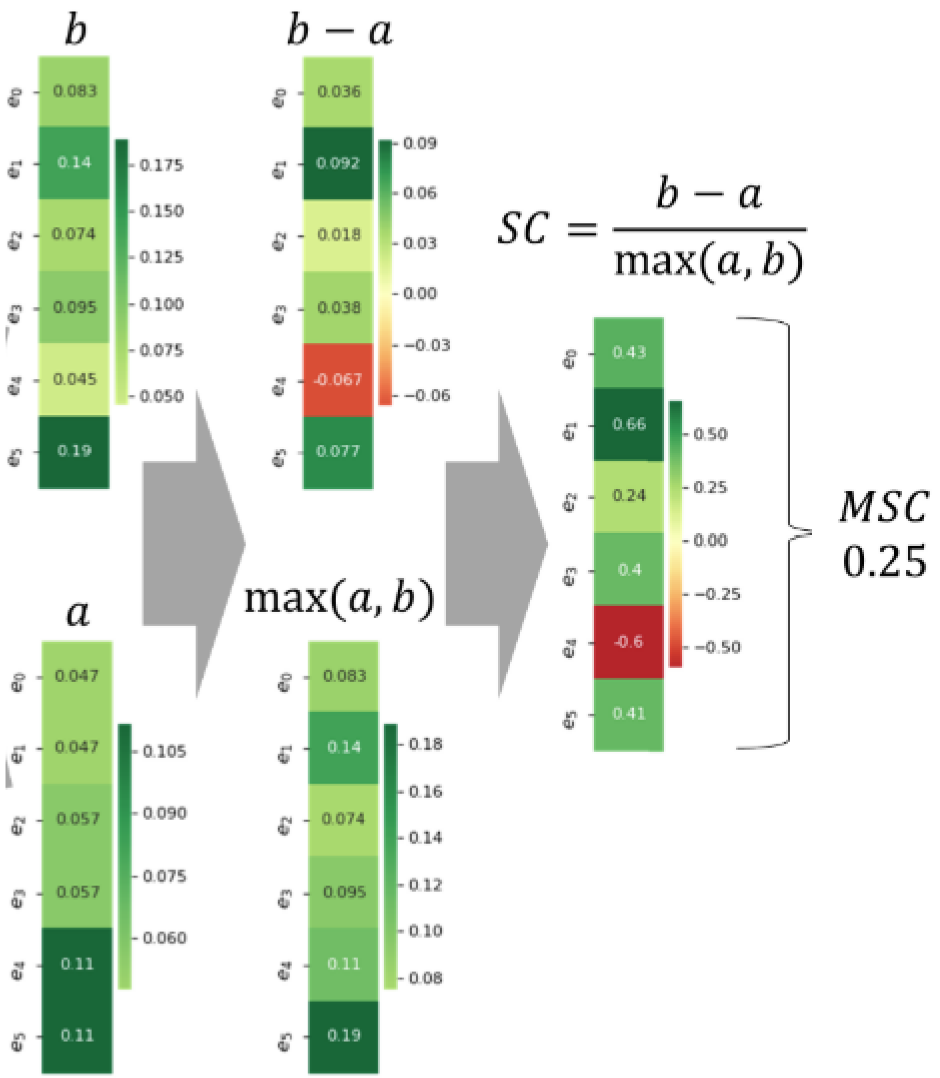
$$\boxed{a = \frac{1}{|A| - 1} \sum_{\substack{x \in A, \\ x \neq i}} d(i, x)} \quad \text{and} \quad \boxed{b = \min_{C \neq A} \frac{1}{|C|} \sum_{x \in C} d(i, x)}$$

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차 예**

- 각 Sample 별 Silhouette Coefficient 계산
- 전체 Sample Silhouette Coefficient의 평균 → MSC
- MSC가 1에 근접할 수록 Source model과 Target data로 Transfer learning 결과가 좋음

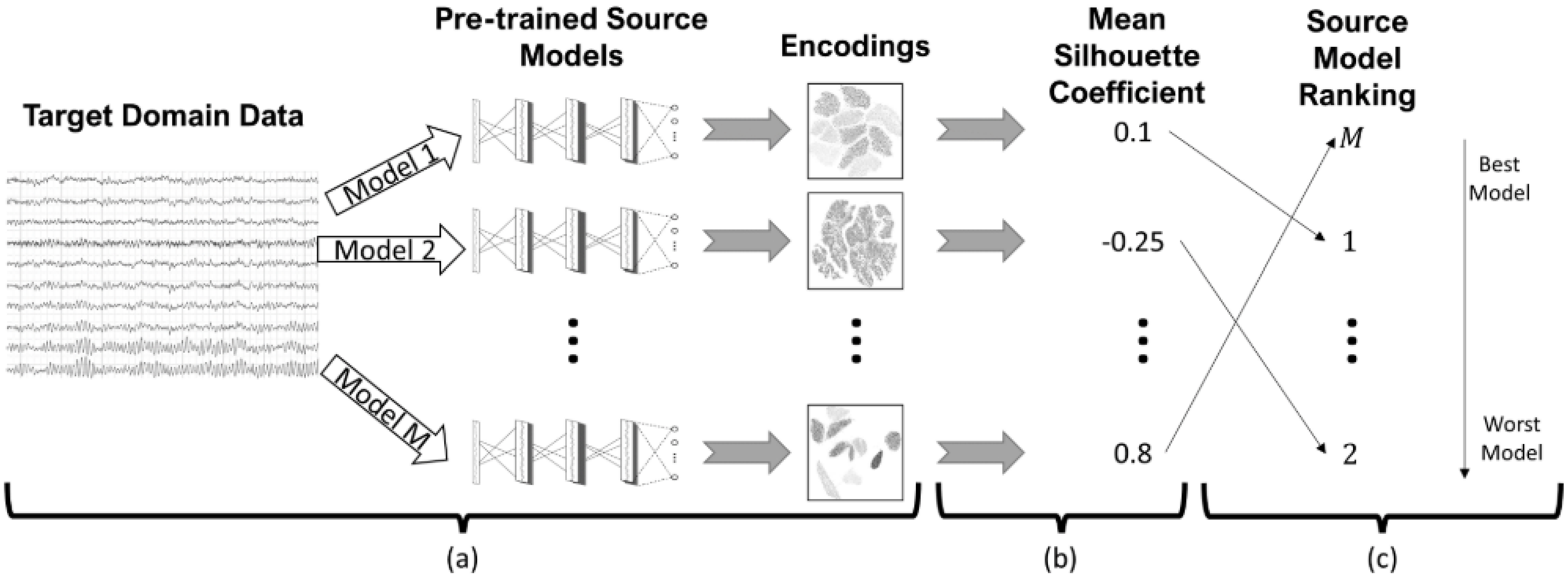


Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 절차 (Overview)**

- a) 각 Pre-trained model 로 부터 Encoding 추출
- b) MSC 계산
- c) MSC 우선순위화 (Higher is better)



Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 알고리즘**

- Input : Pre-Trained Models, Target dataset
- Output : MSC Ranking
- 1) Source Model 가져옴 (line 3)
- 2) Source Model에서 분류 Layer 제거 (line 4)
- 3) Target dataset의 Encoding 생성 (line 5)
- 4) Encoding의 MSC 계산 (line 6)
- 5) MSC 기준 Ranking (line 7)

Algorithm 1 Source Model Ranking

Input: *models*: List of all pre trained base models ℓ: layer from which to generate encodings
X_{target_train}: training sequences from D_{target}
Y_{target_train}: training class labels from D_{target}

Output: *rankings*: A $|models|$-length list with the source models ranked by their predicted transfer learning performance

1 $distances \leftarrow |models|$-length list initialized to Inf
2 **for** $s \leftarrow 1$ **to** $|models|$ **do**
3 $\quad M_{source} \leftarrow models[s]$
4 $\quad encoder \leftarrow M_{source}$ truncated at layer ℓ
5 $\quad encodings \leftarrow encoder(X_{target\_train})$
6 $\quad distances[s] \leftarrow -MSC(encodings, Y_{target\_train})$ ▷ calculate Mean Silhouette Coefficient using Cosine distance
7 $rankings \leftarrow argsort(distances)$ ▷ rank source models according to MSC distance in increasing order
8 **return** *rankings*

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 알고리즘**

- Input : Pre-Trained Models, Target dataset
- Output : MSC Ranking
- 1) Source Model 가져옴 (line 3)
- 2) Source Model에서 분류 Layer 제거 (line 4)
- 3) Target dataset의 Encoding 생성 (line 5)
- 4) Encoding의 MSC 계산 (line 6)
- 5) MSC 기준 Ranking (line 7)

Algorithm 1 Source Model Ranking

Input: *models*: List of all pre trained base models ℓ: layer from which to generate encodings
X_{target_train}: training sequences from D_{target}
Y_{target_train}: training class labels from D_{target}

Output: *rankings*: A $|models|$-length list with the source models ranked by their predicted transfer learning performance

1 $distances \leftarrow |models|$-length list initialized to Inf
2 **for** $s \leftarrow 1$ **to** $|models|$ **do**
3 $\quad M_{source} \leftarrow models[s]$
4 $\quad encoder \leftarrow M_{source}$ truncated at layer ℓ
5 $\quad encodings \leftarrow encoder(X_{target\_train})$
6 $\quad distances[s] \leftarrow -MSC(encodings, Y_{target\_train})$ ▷ calculate Mean Silhouette Coefficient using Cosine distance
7 $rankings \leftarrow argsort(distances)$ ▷ rank source models according to MSC distance in increasing order
8 **return** *rankings*

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 알고리즘**

- Input : Pre-Trained Models, Target dataset
- Output : MSC Ranking
- 1) Source Model 가져옴 (line 3)
- 2) Source Model에서 분류 Layer 제거 (line 4)
- 3) Target dataset의 Encoding 생성 (line 5)
- 4) Encoding의 MSC 계산 (line 6)
- 5) MSC 기준 Ranking (line 7)

Algorithm 1 Source Model Ranking

Input: *models*: List of all pre trained base models ℓ: layer from which to generate encodings
X_{target_train}: training sequences from D_{target}
Y_{target_train}: training class labels from D_{target}
Output: *rankings*: A $|models|$-length list with the source models ranked by their predicted transfer learning performance

```
1 distances ← |models|-length list initialized to Inf
2 for s ← 1 to |models| do
3     M_source ← models[s]
4     encoder ← M_source truncated at layer ℓ
5     encodings ← encoder(X_target_train)
6     distances[s] ← −MSC(encodings, Y_target_train)   ▷ calculate Mean Silhouette Coefficient using Cosine distance
7 rankings ← argsort(distances)   ▷ rank source models according to MSC distance in increasing order
8 return rankings
```

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 알고리즘**

- Input : Pre-Trained Models, Target dataset
- Output : MSC Ranking
- 1) Source Model 가져옴 (line 3)
- 2) Source Model에서 분류 Layer 제거 (line 4)
- 3) Target dataset의 Encoding 생성 (line 5)
- 4) Encoding의 MSC 계산 (line 6)
- 5) MSC 기준 Ranking (line 7)

Algorithm 1 Source Model Ranking

Input: *models*: List of all pre trained base models ℓ: layer from which to generate encodings
X_{target_train}: training sequences from D_{target}
Y_{target_train}: training class labels from D_{target}

Output: *rankings*: A |*models*|-length list with the source models ranked by their predicted transfer learning performance

```
1 distances ← |models|-length list initialized to Inf
2 for s ← 1 to |models| do
3     M_source ← models[s]
4     encoder ← M_source truncated at layer ℓ
5     encodings ← encoder(X_target_train)
6     distances[s] ← −MSC(encodings, Y_target_train)   ▷ calculate Mean Silhouette Coefficient using Cosine distance
7 rankings ← argsort(distances)   ▷ rank source models according to MSC distance in increasing order
8 return rankings
```

Source Model Selection for Deep Learning in the Time Series Domain

Method

❖ **Source Model Selection 알고리즘**

- Input : Pre-Trained Models, Target dataset
- Output : MSC Ranking
- 1) Source Model 가져옴 (line 3)
- 2) Source Model에서 분류 Layer 제거 (line 4)
- 3) Target dataset의 Encoding 생성 (line 5)
- 4) Encoding의 MSC 계산 (line 6)
- 5) MSC 기준 Ranking (line 7)

Algorithm 1 Source Model Ranking

Input: *models*: List of all pre trained base models ℓ: layer from which to generate encodings
X_{target_train}: training sequences from D_{target}
Y_{target_train}: training class labels from D_{target}

Output: *rankings*: A $|models|$-length list with the source models ranked by their predicted transfer learning performance

```
1 distances ← |models|-length list initialized to Inf
2 for s ← 1 to |models| do
3     M_source ← models[s]
4     encoder ← M_source truncated at layer ℓ
5     encodings ← encoder(X_target_train)
6     distances[s] ← −MSC(encodings, Y_target_train)   ▷ calculate Mean Silhouette Coefficient using Cosine distance
7 rankings ← argsort(distances)   ▷ rank source models according to MSC distance in increasing order
8 return rankings
```

Source Model Selection for Deep Learning in the Time Series Domain

❖ Dataset

- UCR Archive 의 다변량, 단변량 시계열 데이터 셋 활용
- 다변량 데이터 셋은 단변량으로 분할 (예 : 가속도계 측정값인 Cricket → CricketX, CricketY, CricketZ)

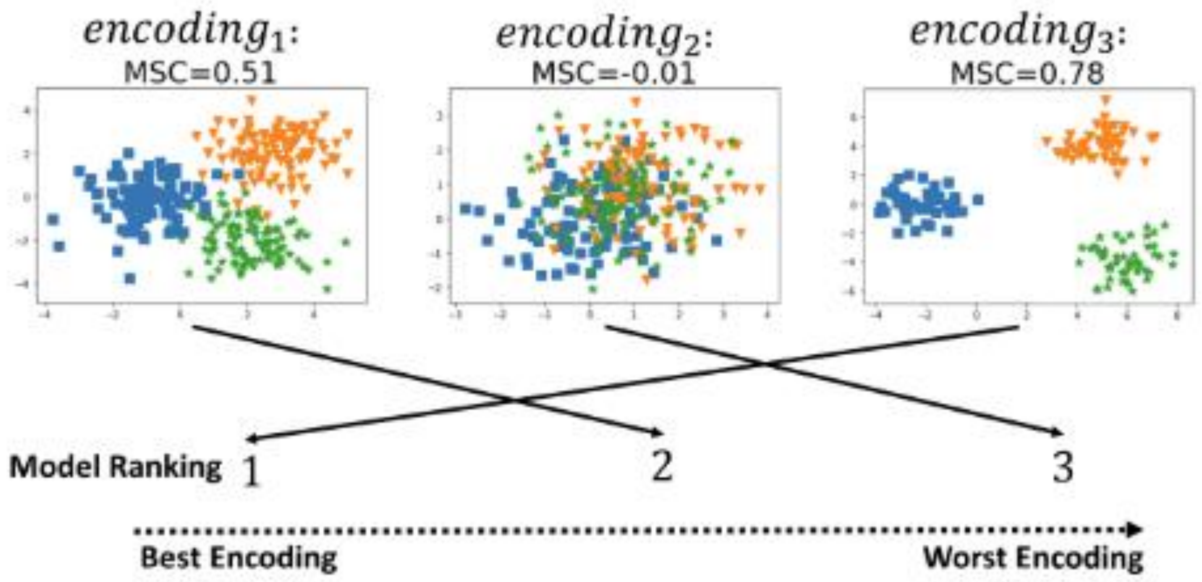
	Number of classes	Time series Length	Size of training set	Size of testing set	Total Size
min	2	24	16	20	40
max	60	2709	8926	8236	16637
median	3	300	200	400	780
mean	7.67	422.21	432.88	1164.8	1597.68
std. dev.	11.25	429.06	1016.19	1665.58	2417.73

Source Model Selection for Deep Learning in the Time Series Domain

Experiments

❖ **Source Model Selection 효과 분석**

- 특정 Source의 Pre-Trained model로 Encoding 된 결과를 2d화
- MSC가 높게 산출된 모델일 수록 인코딩 공간에서 잘 구분 됨을 확인 함



Source Model Selection for Deep Learning in the Time Series Domain

Experiments

❖ **Performance Metrics**

- Top-1 Accuracy : 최적의 Source model이 1st Rank로 우선순위 화 되었는지

  → 1순위 여부만 관심, 상대적인 순위 정보 누락

- Mean Reciprocal Rank (MRR)(5) : 모든 순위를 고려하여, 상대적인 정확도 평가

$$MRR = \frac{1}{|Q|} \sum_{i=0}^{|Q|} \frac{1}{rank_i} \tag{5}$$

Source Model Selection for Deep Learning in the Time Series Domain

Experiments

❖ Alternative Clustering Quality Functions

- 본 연구에서 제안된 Silhouette Coefficient 기반 Clustering 품질 계산 방법론과 대안과의 비교 목적

- Variance Ratio Criterion (VRC)[6]
  - SSb : 클러스터 간 변동 / SSw : 클러스터 내 변동
  - k : 클러스터 수
  - VRC가 높을 수록 높게 산출 됨

$$VRC_k = \frac{SS_b/(k-1)}{SS_w/(N-k)} \tag{6}$$

where:

$$SS_b = \sum_{j=1}^{k} n_j \cdot (\bar{x}_j - \bar{x})^2 \quad \text{and} \quad SS_w = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (x_{ij} - \bar{x}_j)^2$$

- Davies-Bouldin Index (DB)[7]
  - K : 클러스터 수
  - Dij는 클러스터 i와 j의 중심 사이의 거리
  - Sx는 클러스터 x의 중심과 클러스터 x내의 모든 시퀀스 간 평균 거리

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \frac{s_i + s_j}{d_{ij}} \tag{7}$$

Source Model Selection for Deep Learning in the Time Series Domain

Experiments

❖ **Source 데이터 선정 알고리즘 간 성능 비교**

- IDS와 본 연구 방법론인 SMS는 무작위로 Source 모델을 선택한 것 대비 통계적으로 유의미한 성능 향상을 보임
- IDS의 경우 Source 데이터를 사용하는 반면 SMS는 Source 모델만 사용함에도 Top-1 Accuracy 측면 더 높은 성능

Method	MRR	Top-1			Uses Source Data?
		Accuracy	Hits	p-value	
Random Source Model Selection	0.0597	1.2%	1.01	1	No
IDS [6]	**0.1162**	4.7%	4	0.019 *	Yes
SMS - our method	0.115	**7.1%**	**6**	**0.0006 ***	No

Source Model Selection for Deep Learning in the Time Series Domain

Experiments

❖ **클러스터링 품질 평가 알고리즘간 성능 비교**

- VRC, DB 및 Silhouette Coefficient에서 Cosine / Euclidean / Manhattan Distance 간 효과 평가
- 본 연구에서 제안 된 Cosine Distance 기반 Silhouette Coefficient 가 가장 높은 성능을 기록 함

Clustering Quality Function	MRR	Top-1			Uses Source Data?
		Accuracy	Hits	p-value	
Random Source Model Selection	0.0597	1.2%	1.01	1	No
Variance Ratio Criterion	0.0617	2.4%	2	0.2686	No
Davies-Bouldin Index	0.0956	3.5%	3	0.0814	No
Silhouette Coefficient using Euclidean Distance	0.1066	5.9%	5	0.0036 *	No
Silhouette Coefficient using Manhattan Distance	0.1102	5.9%	5	0.0036 *	No
Silhouette Coefficient using Cosine Distance (SMS)	**0.115**	**7.1%**	**6**	**0.0006 ***	No

Source Model Selection for Deep Learning in the Time Series Domain

Summarization

❖ Conclusion

- Source data 활용 없이 Transfer learning 에 효과적인 Source model 선정 방법론 제시
- Mean Silhouette Coefficient활용 Source 모델의 인코딩 품질을 측정
- 기존 연구 IDS대비 Source data활용 없이 Top-1 Accuracy 기준 우위, MMR 기준 동등한 성능 달성
- Source data가 없는 실제 환경에 적용 가능한 장점

Conclusion

❖ **Introduction**

- 각종 센서로부터 다양한 시계열 데이터가 폭발적으로 수집되고 있음
- 이러한 시계열 데이터 활용을 위해 딥러닝 모델을 도입하는데 대량의 레이블된 데이터가 필요함
- 실제 현장에서는 대량의 레이블링 된 데이터가 부족하여, 다양한 레이블 부족 문제에 대한 해결책 리뷰 하였음
- 이중 서로 다른 Domain의 데이터를 활용하여 부족한 레이블링 데이터를 극복하는 Transfer learning 과정에서 Source domain을 효과적으로 선택할 수 있는 방법론을 살펴봄

❖ **Source Selection Methodology in Transfer Learning of Time Series Data**

- IDS : Inter-Datasets Similarity
  - Source dataset 활용 Target dataset 간 유사도를 계산하여 최적 Source dataset 탐색
- SMS : Source Model Selection
  - 현실에서는 Source dataset에 대한 접근이 제한적이므로, Source Model 만 활용하여 MSC를 계산하는 방식으로 최적의 Source dataset을 탐색하는 방법론

고맙습니다

References

[1] Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., & Li, X. (2023). Label-efficient time series representation learning: A review. arXiv preprint arXiv:2302.06433.

[2] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2018, December). Transfer learning for time series classification. In 2018 IEEE international conference on big data (Big Data) (pp. 1367-1376). IEEE.

[3] Meiseles, A., & Rokach, L. (2020). Source model selection for deep learning in the time series domain. IEEE Access, 8, 6190-6200.