

# 시계열 데이터의 자동기계학습을 위한 메타 모델 기반의 특징 선택 방법

류서현<sup>1</sup> · 이다경<sup>1</sup> · 안길승<sup>2</sup> · 허 선<sup>1\*</sup>

<sup>1</sup>한양대학교 산업경영공학과 / <sup>2</sup>현대자동차 빅데이터실

## A Meta Model-based Feature Selection Method for AutoML of Time Series Data

Seohyeon Ryoo<sup>1</sup> · Dagyung Lee<sup>1</sup> · Gilseung Ahn<sup>2</sup> · Sun Hur<sup>1\*</sup>

<sup>1</sup>Department of Industrial and Management Engineering, Hanyang University

<sup>2</sup>Big Data Group, Hyundai Motors Company

Feature engineering is a key step to construct machine learning model as it determines the upper limit of model's performance. However, designing feature engineering is generally iterative, complex and time-consuming step. Also, the large scale of time series data is rapidly generated from the industry, but there is a shortage of data scientists to handle them. So, it has become necessary to automate this process. In this paper, we aim to develop a meta model-based feature selection method that can learn about which features work best given the dataset. The proposed meta-model is a kind of warm-start that can search from the candidate features that is expected to be good without starting a new search for each data. Proposed method is compared by real time-series datasets obtained from UEA & UCR Time Series Classification Repository. Then, we show the proposed method outperforms random search in terms of F1-measure.

**Keywords:** Automated Machine Learning(AutoML), Feature Selection, Time Series Classification, Decision Tree, Meta-Learning

### 1. 서 론

기계 학습(machine learning) 모델은 데이터 탐색, 데이터 전처리, 특징 공학(feature engineering), 알고리즘 선택, 하이퍼파라미터 튜닝(hyperparameter tuning) 등의 단계를 거쳐 개발한다. 각 단계에 해당하는 작업을 수행하기 위해서는 상당한 컴퓨팅 자원과 시간이 필요할 뿐만 아니라, 데이터 과학자 등의 전문 인력도 필요하다. 그러나 최근 데이터 양과 관련 연구 수요가 폭발적으로 증가하면서 금융, 마케팅, 제조 등 다양한 분야의 데이터 관련 인력은 크게 부족하다(He *et al.*, 2021). 이에 따라 데이터 관련 전문 지식이 충분하지 않은 사용자도 손쉽게 기

계 학습 모델을 구축할 수 있도록 기계 학습 파이프라인의 각 프로세스를 자동화 한 자동 기계 학습(automated machine learning; AutoML)이 많은 주목을 받고 있다.

자동 기계 학습은 기계 학습 모델의 성능을 최대화하기 위해 전처리 기법, 특징 공학, 알고리즘 및 관련 하이퍼파라미터 등의 최적화 문제를 해결하는 기술이다(Zöller and Huber, 2021). 그러나 이들 최적화 문제는 최적해를 분석적으로 구하는 것이 어려울 뿐만 아니라, 해 하나를 평가하는 데에 시간이 오래 걸린다는 공통점이 있다. 이에 따라 유전 알고리즘(Ahn and Hur, 2020; Safarik *et al.*, 2018), 베이지안 최적화(Wu *et al.*, 2019), 입자 군집 최적화(Toal *et al.*, 2011) 등 다양한 휴리스틱

이 성과는 2019년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2019R1A2C1088255).

\* 연락처: 허 선 교수, 15588 경기도 안산시 상록구 한양대학교 55 한양대학교 산업경영공학과, Tel : 031-400-5265, Fax : 031-400-5265,

E-mail : hursun@hanyang.ac.kr

2022년 10월 6일 접수; 2022년 11월 16일 수정본 접수; 2022년 11월 23일 게재 확정.

알고리즘이 최적화 문제를 해결하는데 활용된다.

최적화 대상 가운데 특징 공학은 기계 학습 모델을 개발하는 전체 단계에서 가장 중요하다(Christ *et al.*, 2016). 그러나 자동 기계 학습 관련 연구는 주로 알고리즘 선택과 하이퍼파라미터 튜닝 문제에 초점을 두고 있다. 이는 다른 특징을 갖는 데이터셋에 같은 종류의 특징 공학 기법을 일괄적으로 적용할 수 없기 때문이다(Wolpert and Macready, 1997). 이에 더해 기계 학습 알고리즘과 하이퍼파라미터는 다양한 데이터로 학습하고 평가하는 과정을 통해 탐색의 효율을 높일 수 있는 반면, 특징 공학은 실험과 경험을 활용하는 것이 어렵다. 즉, 새로운 데이터가 입력될 때마다 여러 기법을 적용하고 비교하는 과정을 통해서 적절한 특징 공학 기법을 찾아야 한다.

시계열 분류(time series classification)는 분류기를 사용하여 시계열 인스턴스(instance)를 미리 정의된 클래스에 할당하는 지도 학습 문제이다(Keogh and Kasetty, 2003). 최근 시계열 데이터는 제조, 금융, 의료 등 다양한 분야에서 수집되고 있을 뿐만 아니라 관련 연구도 지속적으로 증가하고 있다(Parmentier *et al.*, 2021; Abanda *et al.*, 2019).

시계열 분류를 할 때에는 시계열 데이터의 시간 순서와 길이가 다른 인스턴스를 고려해야 하므로 시계열이 아닌 데이터의 분류 문제와는 접근방법이 다르다(Bagnall *et al.*, 2017). 시계열 분류를 수행하는 대표적인 방법으로 특징 기반의 시계열 분류를 들 수 있는데, 이 방법은 시계열 인스턴스의 평균, 왜도, 첨도, 상관성, 엔트로피 등의 다양한 통계량과 정보량을 추출하고, 각 인스턴스를 특징 벡터로 변환하여 분류한다(Fulcher and Jones, 2014). 이 연구에서는 시계열 인스턴스를 분류하는 데에 활용하는 함수 형태의 특징을 시계열 특징이라고 부르기로 한다. 또한, 시계열 특징으로부터 추출하여 시계열 데이터의 고유한 특성을 나타내는 특징을 메타 특징이라고 한다.

이 연구에서는 시계열 분류 자동화를 위한 메타 모델 기반의 특징 선택 방법을 제안한다. 여기서 시계열 분류 자동화란 자동 기계 학습을 시계열 분류에 적용한 것을 말한다. 또한 메타 모델은 시계열 특징과 메타 특징을 입력 값으로 받고 모델의 예상 성능을 출력하는 대체 모델(surrogate model)이라고 할 수 있다. 제안하는 메타 모델은 두 종류의 결정 나무(decision tree)가 상하(upper and lower)로 결합된 구조이다. 상부 나무인 메타 나무(meta tree)는 데이터를 분리하는 회귀나무로서, 각 잎 노드(leaf node)에는 메타 특징을 바탕으로 성능의 분포가 유사한 데이터가 모인다. 하부 나무인 특징 샘플링 나무(feature sampling tree)는 시계열 특징을 입력하면 예상되는 성능을 출력하는 회귀나무이다. 이 때 메타 나무의 잎 노드가 특징 샘플링 나무의 뿌리 노드(root node)이다.

본 연구의 주요 기여는 다음과 같다. 첫째, 문헌 조사를 통해 시계열 분류에 적합한 시계열 특징들을 함수 형태로 정의한다. 이렇게 정의한 시계열 특징은 본 연구에서 제안한 방법 뿐만 아니라 일반적인 시계열 분류에도 활용할 수 있다. 둘째, 시

계열 분류에 활용할 수 있는 메타 특징을 정의한다. 셋째, 데이터마다 적절한 특징을 선택할 수 있는 메타 모델을 개발한다. 이 메타 모델은 새로운 데이터와 유사한 기존 데이터의 학습 경험을 바탕으로 특징 선택에 고려할 만한 후보 특징 집합을 빠르게 찾는다. 즉, 데이터의 특성에 따른 적절한 시계열 특징을 파악하여 이를 시계열 분류 자동화에서 특징을 선택하는데 사용한다.

논문의 나머지는 다음과 같이 구성되어 있다. 제2장에서는 본 연구에서 고려하는 문제를 구체적으로 정의한다. 제3장에서는 제안 방법을 단계별로 자세히 설명한다. 제4장에서는 제안 방법과 랜덤 서치(random search)를 비교하는 실험을 수행하고 그 결과를 분석함으로써 제안 방법이 유효함을 보인다. 마지막으로 제5장에서는 본 연구에 대한 결론과 향후 연구 방향을 제시한다.

## 2. 문제 정의

본 연구에서 고려하는 문제는 새로 입력된 데이터셋  $D_{new}$ 에 대하여 사용자가 설정한 모델  $f$ 과 성능 지표  $S$ 를 사용하여 가장 우수한 성능을 갖는 특징 집합  $\phi^*$ 을 찾아내는 것이다. 이는 식 (1)과 같이 표현할 수 있다.

$$\phi^* = \underset{\phi}{\operatorname{argmax}} S(f_{\phi} | D_{new}) \quad (1)$$

여기서  $f_{\phi}$ 는  $\phi$ 를 특징 집합으로 사용한 분류 모델이다. 또한  $S(f_{\phi} | D_{new})$ 는  $D_{new}$ 에 대하여 사용자가 설정한 성능 지표  $S$ (예: 정확도, F1 점수 등)로  $f_{\phi}$ 를 평가했을 때의 성능을 나타내며, 이 연구에서는 식 (2)와 같이  $k$ 겹 교차 검증( $k$ -fold cross validation)을 사용하여 계산한다.

$$S(f_{\phi} | D) = \frac{1}{k} \sum_{i=1}^k S(f_{\phi} | D_{train}^{(i)}, D_{test}^{(i)}) \quad (2)$$

여기서 전체 데이터  $D$ 는  $k$ 개의 부분 집합( $D = \bigcup_{i=1}^k D^{(i)}$ )으로 나누어지며,  $D_{train}^{(i)} = D - D^{(i)}$ 과  $D_{test}^{(i)} = D^{(i)}$ 는 각각  $i$ ( $i = 1, 2, \dots, k$ ) 번째 반복에서의 학습 데이터와 테스트 데이터를 나타낸다. 즉,  $S(f_{\phi} | D_{train}^{(i)}, D_{test}^{(i)})$ 는  $f_{\phi}$ 를  $D_{train}^{(i)}$ 로 학습하고  $D_{test}^{(i)}$ 로 평가했을 때의 성능이다.

이 연구에서는 식 (1)에서 최적의 시계열 특징 집합  $\phi^*$ 를 찾기 위해  $\phi$ 에 대한 탐색 공간  $\Omega(\phi \subset \Omega)$ 을 정의한다. 이는 시계열 특징으로 사용할 수 있는 특징 공간을 제한하여 효율적으로 해를 탐색하기 위함이다. 하지만 탐색 공간을 한정하더라도 전역 탐색을 통해 최적해를 구하는 데 오랜 시간이 소요될 수 있으므로 메타 모델을 활용하여 가능한 짧은 시간 내에

최적해를 구할 수 있게 한다.

제안하는 메타 모델은 기존 데이터셋  $D_{old}$ 에서 탐색한 해, 즉 특징 집합을 바탕으로 새로운 데이터셋  $D_{new}$ 에서 최적의 특징 집합을 탐색하는 시간을 줄일 수 있다. 이는 데이터마다 새롭게 탐색을 시작하지 않고 좋을 것이라고 예상되는 해부터 탐색하는 일종의 워밍 스타트(warm-start) 방법이라고 할 수 있다.

### 3. 제안 방법

이 절에서는 본 연구에서 제안하는 방법을 자세히 설명한다. 3.1절에서는 시계열 특징과 메타 특징을 정의하고, 이를 기반으로 메타 데이터를 생성하는 방법을 설명한다. 3.2절에서는 메타 모델 및 그 학습 방법을 소개한다. 3.3절에서는 학습한 메타 모델을 사용하여 새로운 데이터에 적합한 특징 집합을 생성하는 방법을 설명한다.

#### 3.1 메타 데이터 생성

선행 연구(Christ *et al.*, 2018; Barandas *et al.*, 2020)를 참고하여 함수 형태의 시계열 특징 89개 ( $\Omega = \{\phi_1, \phi_2, \dots, \phi_{89}\}$ )를 선정하고 이를 부록 A의 <Table A.1>에 정리하였다. 또한 Rivolli *et al.* (2022)의 연구를 참고하여 본 연구의 목적에 부합하는 43개의 메타 특징 ( $\mathcal{P} = \{p_1, p_2, \dots, p_{43}\}$ )을 부록 A의 <Table A.2>와 같이 선정하였다. 이 때 메타 특징은 <Table A.1>에 정리한 시계열 특징으로부터 계산된다. 따라서 메타 특징 역시 시계열 특징과 마찬가지로 데이터의 특성을 나타낸다고 할 수 있다.

<Figure 1>은 메타 데이터를 생성하는 과정이다. 메타 데이터는 다양한 데이터셋과 시계열 특징 간의 관계를 학습하여 만들어진 데이터로서 메타 모델을 학습하는 데 사용한다. 기존 데이터  $D \in D_{old}$ 로부터 메타 데이터를 만드는 방법은 다음과 같다.

첫째, <Table A.1>에서 정의한 함수 형태의 시계열 특징  $\phi$ 에  $D$ 를 매핑하고, 이를 바탕으로 <Table A.2>에 정의한 메타 특징  $\mathcal{P}$ 를 계산한다.

둘째,  $\Omega$ 에서 임의로 각 특징을 사용할 지 결정하는 방식으로 특징 집합  $\phi$ 을 샘플링한다. 이 때  $\phi$ 는  $\tilde{\phi}$  식 (3)과 같이 지지 변수로 표현한다.

$$\tilde{\phi}_k = \begin{cases} 1, & \text{if } \phi_k \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

마지막으로, 식 (2)를 이용하여 사용자가 설정한 분류 모델  $f$ 에 대하여  $\phi$ 를 평가하고  $S(f_\phi|D)$  값을 구한다. 이 과정을 통해 얻은  $\phi, \mathcal{P}, S(f_\phi|D)$ 의 값은 메타 데이터의 샘플이 된다. 즉,  $(\phi, \mathcal{P})$ 는 특징 벡터이고  $S(f_\phi|D)$ 은 라벨(label)이다. <Figure 2>는 메타 데이터의 샘플을 예시하고 있다.

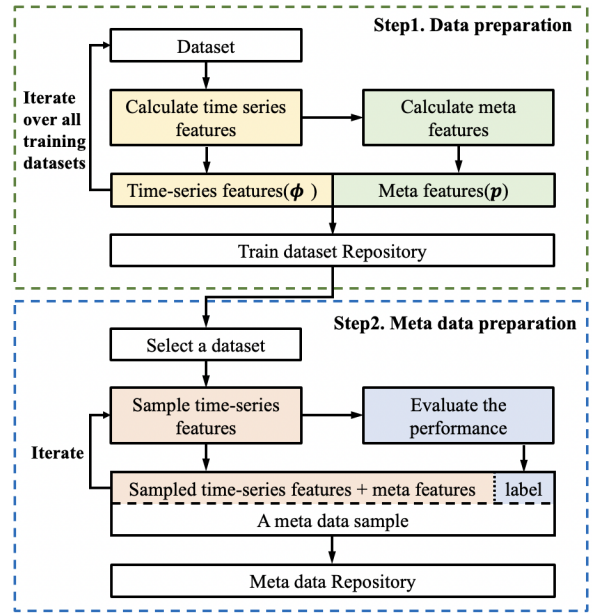


Figure 1. Meta Data Creation Process

Dataset	Time-series features					Meta-features				Performance
	$\phi_1$	$\phi_2$	$\phi_3$	...	$\phi_{89}$	$p_1$	$p_2$	...	$p_{43}$	
ACSF1	1	0	0	...	0	0.445	0.112	...	2.302	0.9410
Beef	1	1	0	...	1	1.483	0.056	...	1.609	0.88
ACSF1	0	1	1	...	0	0.445	0.112	...	2.302	0.7857
⋮										⋮
Car	0	1	1	...	1	0.741	0.044	...	1.386	0.74
ACSF1	0	0	0	...	1	0.445	0.112	...	2.302	0.85
Beef	1	0	1	...	0	1.483	0.056	...	1.609	0.9552
⋮										⋮

Figure 2. An Example of Meta Data Sample

메타 데이터는 다양한 분야와 여러 형태의 시계열 데이터셋을 다양한 시계열 특징 집합으로 분류하여 성능을 측정하고 이를 라벨로 사용한 데이터이다. 이 메타 데이터의 메타 특징, 시계열 특징과 성능 간의 관계를 바탕으로 3.2절의 메타 모델을 학습한다.

#### 3.2 메타 모델

이 연구에서 제안하는 메타 모델 구조는 <Figure 3>와 같다. 그림에서 볼 수 있듯이 제안하는 메타 모델은 메타 나무와 특징 샘플링 나무가 상하로 결합된 구조이다. 즉, 메타 나무의 잎 노드는 특징 샘플링 나무의 뿌리 노드가 된다.

메타 모델에 결정 나무를 사용한 이유는 결정 나무가 데이터 공간을 수직 혹은 수평으로 나누는 특성이 있어 결정 나무의 각 잎 노드에는 같은 조건을 갖는 샘플들이 포함되기 때문이다. 결정 나무의 특성을 활용하면 메타 나무는 성능이 비슷한 데이터를 구분할 수 있으며, 특징 샘플링 나무는 특정 특징의 사용 여부를 기준으로 데이터를 구분할 수 있다.

메타 모델은 과거 탐색에서 얻은 경험과 정보인 메타 데이터로 메타 나무와 특징 샘플링 나무를 학습하여 만든다. 이는 새로운 데이터셋이 주어졌을 때 적합한 후보 특징 집합을 탐색하는 데에 사용된다. 먼저, 메타 나무는 3.1절에서 만든 메타 데이터의 메타 특징  $p$ 를 입력으로 하고 성능  $S(f_{\tilde{\phi}}|D)$ 를 출력으로 하여 학습한다. 즉, 메타 나무는 메타 특징을 바탕으로 데이터를 분류하며, 같은 잎 노드에 속한 데이터는 메타 특징을 기준으로 시계열적 특성과 성능이 서로 유사하다고 할 수 있다. 한편 특징 샘플링 나무는 메타 트리의 각 잎 노드에 속한 메타 데이터의  $\tilde{\phi}$ 를 입력으로 하고 성능  $S(f_{\phi}|D)$ 를 출력으로 하여 학습한다. 즉, 특징 샘플링 나무는 메타 특징을 기준으로 유사한 데이터에 대해  $\tilde{\phi}$ 의 요소가 1인 특징만 사용하여 모델을 학습했을 때의 성능을 출력한다.

### 3.3 샘플링 방법

메타 모델에서 특징 샘플링 나무는 특징 중요도를 계산하는데 사용될 뿐만 아니라, 평균적으로 성능이 우수한 잎 노드의 규칙 집합을 바탕으로 특징 샘플링을 수행하는 데에도 사용된다. 먼저, 각 샘플링 나무  $T$ 의 개별 잎 노드  $t \in T$ 에 대하여 해당 잎 노드에 속하는 메타 데이터의 라벨( $S$ , 즉 성능) 평균  $\mu_t$ 와 표준편차  $\sigma_t$ 를 계산한다. 그 다음, 각 특징 샘플링 나무의 모든 잎 노드로부터 식 (4)와 같이 확률 값을 계산한다. 이는 각 잎 노드의 속성 값으로 샘플링을 위한 잎 노드 선택 기준으로 사용된다.

$$Prob_t = \frac{u_t + \sigma_t}{\sum_{s \in T} (u_s + \sigma_s)} \quad (4)$$

새로 입력된 시계열 데이터셋  $D_{new}$ 에 대한 특징 샘플링은

세 단계에 거쳐 이뤄진다. 첫째,  $D_{new}$ 에서 메타 특징을 추출하여 메타 나무에 입력한 뒤,  $D_{new}$ 가 도달한 잎 노드가 뿌리 노드인 특징 샘플링 나무를 선택한다. 둘째, 선택한 특징 샘플링 나무의 각 잎 노드를 사전에 식 (4)에서 계산한 확률 값을 바탕으로 샘플링한다. 셋째, 선택한 잎 노드의 규칙 집합을 바탕으로 특징을 선택한다. 예를 들어, 규칙 집합이  $\{(\tilde{\phi}_1=1), (\tilde{\phi}_2=1), (\tilde{\phi}_3=0)\}$  이라면 전체 후보 특징 집합 가운데  $\phi_1$ 과  $\phi_2$ 를 반드시 선택하고  $\phi_3$ 을 선택하지 않는 특징 집합(예:  $\{\phi_1, \phi_2, \phi_5\}$ , 또는  $\{\phi_1, \phi_2, \phi_4, \phi_6, \phi_8\}$ ) 가운데 하나를 랜덤하게 선택한다.

식 (4)의 확률 값을 바탕으로 특징 샘플링 나무의 노드를 선택하는 것은 탐색의 다양화를 높이기 위함이다. 즉, 매 반복에서 새로운 후보 특징 집합을 생성할 때, 우연의 성질을 갖는 확률적 탐색을 통해 다양한 특징 공간을 탐색할 수 있다. 식 (4)에서 평균과 표준편차를 기준으로 잎 노드를 선택한 것은 베이직한 최적화에서 사용하는 탐사(exploitation)와 탐험(exploration) 전략을 참고한 것이다. 두 전략은 해에 따른 목적함수의 값을 알 수 없을 때 사용한다. 탐사 전략은 근처 해가 우수한 해를 탐색하는 전략이며, 탐험 전략이란 편차가 커서 잘 알지 못하는 해를 탐색하는 전략이다(Shahriari *et al.*, 2015). 잎 노드의 평균이 크다는 것은 평균적으로 성능이 우수한 규칙으로 구성됐다는 의미이므로 이러한 노드를 택하는 확률을 높이는 것은 탐사를 수행하기 위한 전략이 될 수 있다. 한편, 잎 노드의 표준편차가 크다는 것은 특징 샘플링에 의한 효과가 들쭉날쭉하다는 의미이며, 이 노드가 선택될 확률을 높이는 것은 예상치 못한 우수한 규칙을 찾는 탐험의 효과를 노릴 수 있다. 즉, 특징 집합을 탐색하는 데 있어 수 백 번에서 수 만 번까지의 반복을 거치므로 단 한번이라도 매우 좋은 해를 탐색한다면 이는 성능이 매우 우수한 특징 집합을 찾을 수 있다

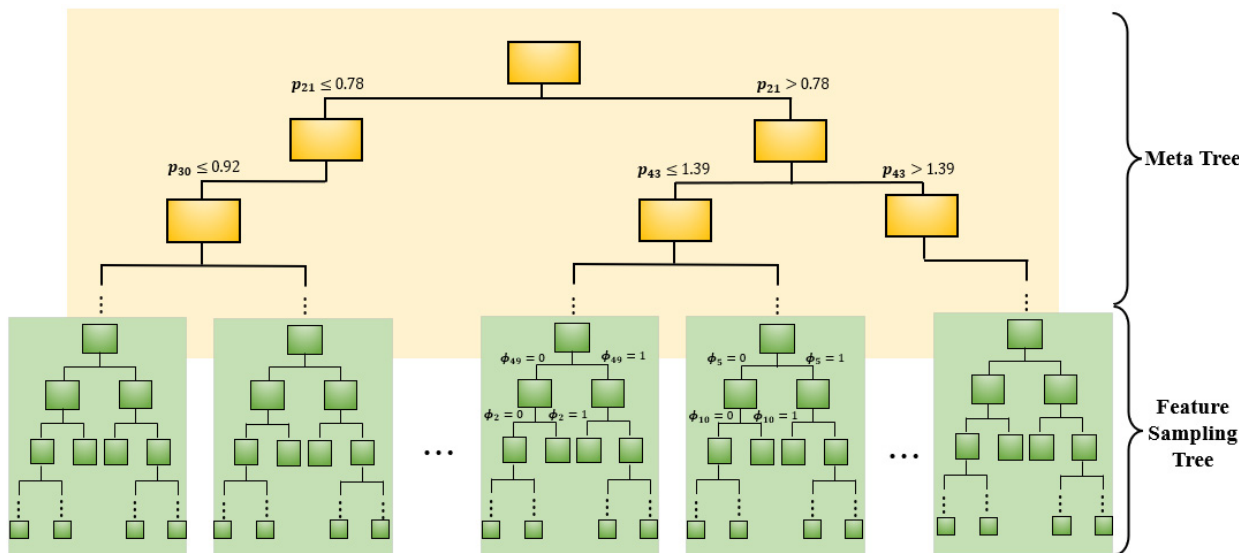


Figure 3. Example of Meta Model Structure

는 것을 의미한다. 이에 더해 평균 혹은 표준편차 중 하나의 값이 매우 크면 작은 값이 무시될 수 있지만, 식 (4)에서 볼 수 있듯이 평균과 표준편차의 합을 기준으로 확률을 계산하므로 평균 혹은 표준편차가 값에 큰 영향을 주지 않는다.

#### 4. 실험 및 결과

이 절에서는 본 연구에서 제안한 메타모델의 성능을 검증하기 위해 메타 모델과 랜덤 서치 각각에 의해 선택된 특징들을 테스트 데이터셋에 적용하여 성능을 비교하는 방식으로 실험을 진행한다. 여기서 랜덤 서치는 사전에 정의한 탐색 공간에 있는 특징을 랜덤하게 선택하여 탐색하는 방법이다.

##### 4.1 실험 방법

실험에 사용한 68개 시계열 데이터셋은 UEA & UCR Time Series Classification Repository(<https://www.timeseriesclassification.com/>)에서 수집했다. 이 가운데 64개는 메타 모델을 학습하는 데 사용하며, 나머지 4개는 평가에 활용한다. 메타 모델을 학습하는 데 사용한 데이터셋과 평가에 사용한 데이터셋은 부록 B의 <Table B.1>과 같다.

시계열 분류 모델로는 랜덤 포레스트(random forest; RF)와 서포트 벡터 머신(support vector machine; SVM)을 사용하며, 성능 척도로는 마이크로 F1 점수(micro f1 score)를 사용한다.

또한 k겹 교차 검증에서 k는 5로 설정한다.

##### 4.2 실험 결과

<Figure 3>과 <Figure 4>는 제안 방법과 랜덤 서치에 의해 얻은 특징들을 사용하여 랜덤 포레스트와 서포트 벡터 머신으로 평가용 데이터셋의 시계열 분류를 수행하여 얻은 결과를 나타낸다. 여기서 x축은 특징 샘플링 반복 횟수를 나타내며, y축은 특징 샘플링 결과를 모델에 적용하여 얻은 누적 마이크로 F1 점수를 나타낸다. 즉, y 축은 매 반복을 거칠 때마다 제안 방법과 랜덤 서치가 탐색한 해 가운데 가장 우수한 성능을 갖는 특징 집합의 성능을 표시하고 있다.

<Figure 3>의 (a)~(d)는 부록 B의 <Table B.1>에 있는 평가용 데이터셋(No.65, 66, 67, 68)에 대해 <Figure 2>에 제시한 메타 모델로부터 특징을 샘플링하고, 이를 랜덤 포레스트에 적용하는 과정을 10,000번 반복하여 얻은 결과이다. (a), (b)의 첫 고점 발생 시점을 살펴보면, 제안 방법이 더 빠르게 수렴했을 뿐만 아니라 성능면에서도 더 우수하다는 것을 확인할 수 있다. 즉, 제안 방법을 바탕으로 샘플링한 특징을 모델에 적용하여 얻은 성능은 반복 횟수에 상관없이 제안 방법이 전반적으로 높았다. (c)의 경우 제안 방법이 랜덤 서치보다 특징 집합을 더 늦게 찾았지만, 수렴했을 때의 성능은 랜덤 서치보다 높았다. 반면 (d)의 경우, 이 연구에서 제안한 방법과 랜덤 서치를 비교하는 것이 큰 의미가 없다. 이는 정의한 시계열 특징과 메타 특징이 해당 데이터셋에 대하여 매우 유용한 특징으로 구성되어 있어

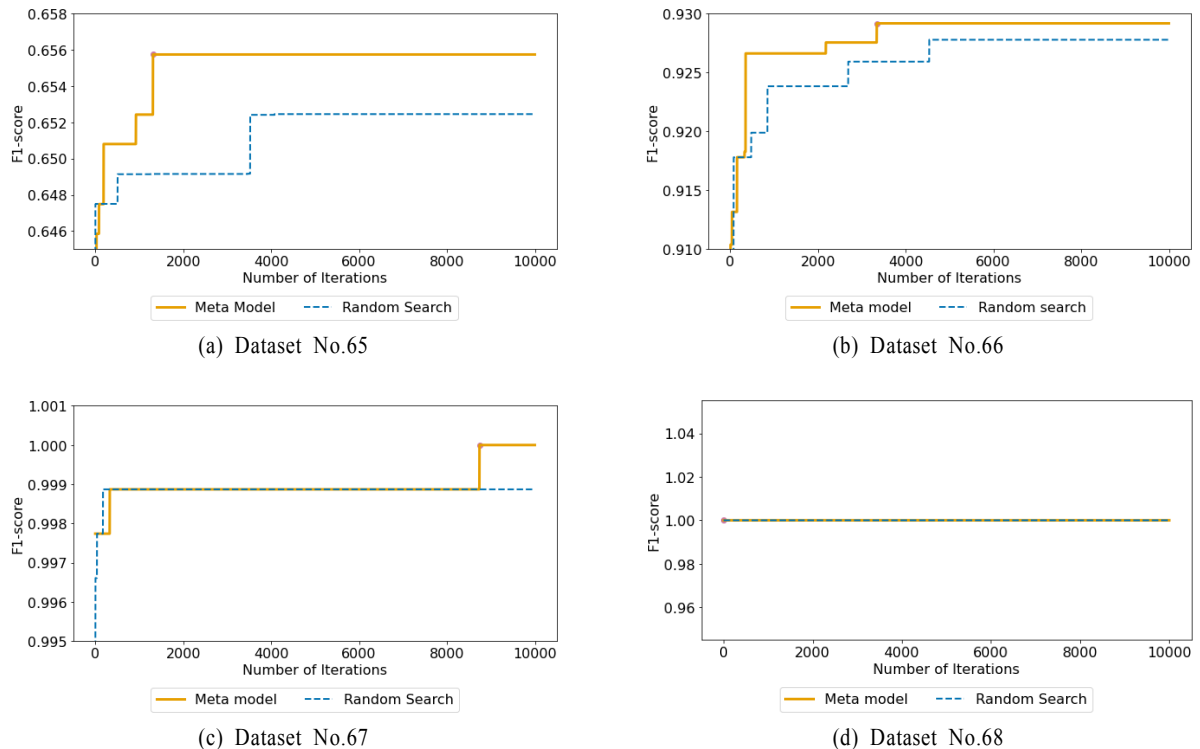


Figure 3. Comparison of F1-Scores According to Number of Iterations with RF



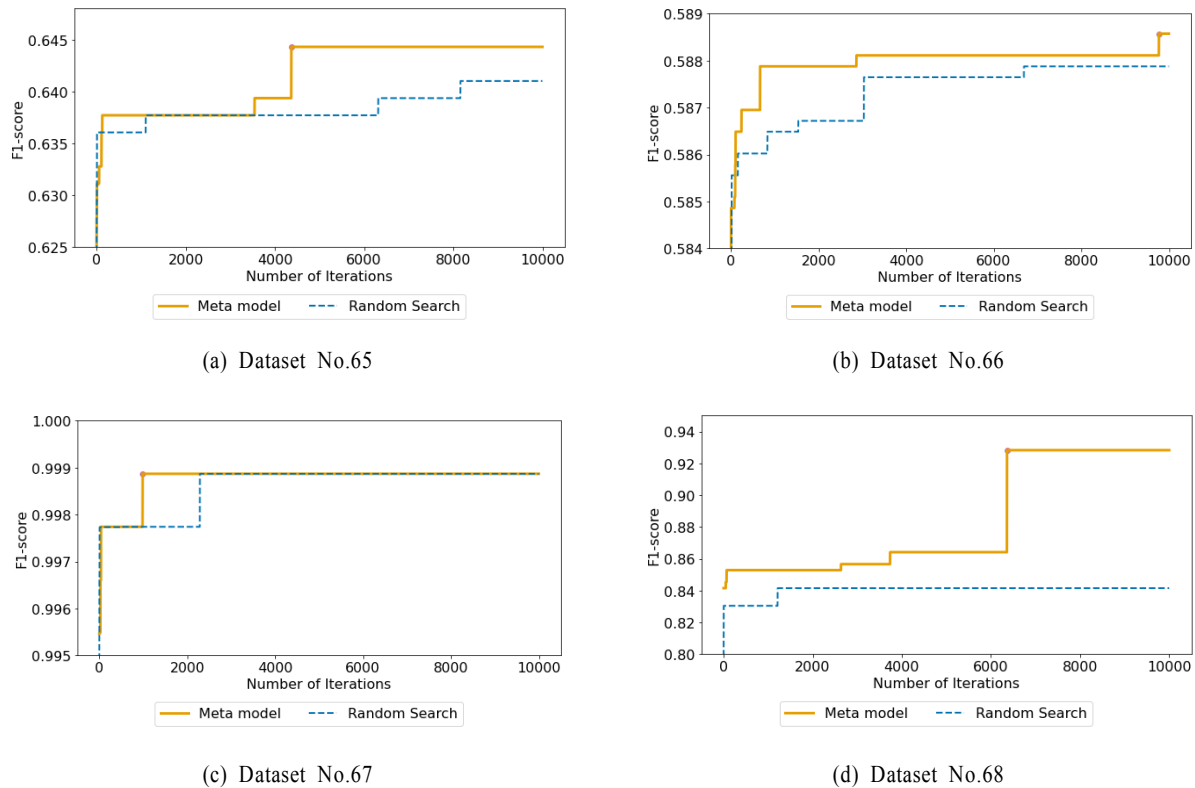


Figure 4. Comparison of F1-Scores According to Number of Iterations with SVM

서 성능이 1에 가깝게 나온 것으로 추측된다.

<Figure 4>의 (a)~(d)는 각 데이터셋(No.65, 66, 67, 68)에 대해 <Figure 2>에 제시한 메타 모델로부터 특징을 샘플링하고, 이를 서포트 벡터 머신에 적용하는 과정을 10,000번 반복하여 얻은 결과이다. (a)에서 특징 샘플링 반복 횟수 약 4,000번까지는 제안 방법과 랜덤 서치와의 성능 우위가 명확히 드러나지 않지만, 4,000이후부터는 비교적 제안 방법의 성능이 더 높은 것을 확인하였다. (b), (d)의 경우 제안 방법이 랜덤 서치에 비해 특징 집합을 찾는 데 더 오랜 시간이 걸렸지만, 특징 샘플링 반복 횟수에 상관없이 제안 방법의 성능이 더 우수한 것을 확인하였다. (c)에서는 제안 방법과 랜덤 서치가 수렴했을 때의 성능은 같지만, 제안 방법이 더 빠르게 수렴했음을 확인하였다.

이 절에서는 메타 모델을 활용하여 데이터셋마다 다른 특징 선택 방법을 분류 모델에 적용하여 성능을 비교하였다. 그 결과, 전반적으로 제안한 메타 모델 기반의 특징 선택 방법이 랜덤 서치보다 더 빠르게 최적해에 수렴했을 뿐만 아니라 성능 면에서도 향상을 보였다. 이를 통해 제안한 메타 모델 기반의 특징 선택 방법이 유의미하다고 판단할 수 있다. 특히 본 연구에서는 주어진 데이터에서 성능이 우수한 특징 집합을 탐색하는 데 있어 적게는 수 백 번에서 최대 만 번까지의 반복을 거친다. 랜덤 서치를 무한히 수행하게 되면 결국 모든 특징조합을 다 비교한 셈이 되므로 성능이 우수한 특징 집합을 찾을 수 있다고 볼 수 있다. 따라서 본 연구에서 제안한 방법에 의한 결과를 오랜 시간 동안 반복한 랜덤 서치의 결과와 비교하여 더 우

수함을 보임으로써 본 연구 방법의 타당성을 검증하였다.

## 5. 결론

이 연구에서는 시계열 데이터 분류에서 특징 선택을 위한 새로운 메타 모델과 특징 샘플링 방법을 제안하였다. 이 때 특징 집합을 효율적으로 탐색하기 위해 시계열 분류에 효과적으로 알려진 시계열 특징과 메타 특징을 정의하여 특징 공간을 제한하였다. 제안하는 메타 모델에서 특징 샘플링 나무의 잎 노드는 베이지안 최적화 기법의 탐사와 탐험 전략을 기반으로 특징을 샘플링하는데 사용하였으며, 실험 결과 랜덤 서치 방법보다 우수한 성능을 보였다.

제안하는 메타 모델은 메타 데이터의 사전 학습 경험을 바탕으로  $S$ (성능) 측면에서 각 데이터셋에 적합한 특징 집합을 훈련하여 만들어진 모델이다. 이는 새로운 데이터가 주어지면 해당 데이터와 유사한 데이터의 사전 학습 경험을 바탕으로 데이터마다 적절한 특징 공학을 설계하지 않고도 적합한 후보 특징 집합을 빠르게 찾는다. 따라서 제안하는 메타 모델이 자동 기계학습을 실현하는 데 기여한다고 볼 수 있다.

이 연구에서는 모델 기반의 메타 특징과 파라미터 조건은 고려하지 않았다. 따라서 추후 연구로 모델에 적합한 메타 특징과 파라미터 튜닝을 고려하여 제안한 방법을 더욱 확장한다면 자동 기계학습에 의한 분류 성능 향상에 도움이 될 것이다.

그리고 주어진 데이터와 유사한 특성을 가진 데이터를 찾아 적절한 특징 집합을 찾는 메타 모델이 어떠한 특성을 가지는 데이터에서 우수한 성능을 갖는지는 추후 연구에서 다룰 예정이다.

## 참고문헌

- Abanda, A., Mori, U., and Lozano, J. A. (2019), A Review on Distance Based Time Series Classification, *Data Mining and Knowledge Discovery*, **33**(2), 378-412.
- Ahn, G. and Hur, S. (2020), Efficient Genetic Algorithm for Feature Selection for Early Time Series Classification, *Computers & Industrial Engineering*, **142**, 106345.
- Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017), The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances, *Data Mining and Knowledge Discovery*, **31**(3), 606-660.
- Bandt, C. and Pompe, B. (2002), Permutation Entropy: A Natural Complexity Measure for Time Series, *Physical Review Letters*, **88**(17), 174102.
- Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., and Gamboa, H. (2020), TSFEL: Time Series Feature Extraction Library, *SoftwareX*, **11**, 100456.
- Batista, G. E., Keogh, E. J., Tataw, O. M., and de Souza, V. (2014), CID: An Efficient Complexity-Invariant Distance for Time Series, *Data Mining and Knowledge Discovery*, **28**(3), 634-669.
- Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A. W. (2018), Time Series Feature Extraction on Basis of Scalable Hypothesis Tests (tsfresh—a python package), *Neurocomputing*, **307**, 72-77.
- Fulcher, B. D. and Jones, N. S. (2014), Highly Comparative Feature-based Time-series Classification, *IEEE Transactions on Knowledge and Data Engineering*, **26**(12), 3026-3037.
- He, X., Zhao, K., and Chu, X. (2021), AutoML: A Survey of the State-of-the-art, *Knowledge-Based Systems*, **212**, 106622.
- Hill, T. P. (1995), A Statistical Derivation of the Significant-digit Law, *Statistical Science*, 354-363.
- Kampouraki, A., Manis, G., and Nikou, C. (2008), Heartbeat Time Series Classification with Support Vector Machines, *IEEE Transactions on Information Technology in Biomedicine*, **13**(4), 512-518.
- Keogh, E. and Kasetty, S. (2003), On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, *Data Mining and Knowledge Discovery*, **7**(4), 349-371.
- Parmentier, L., Nicol, O., Jourdan, L., and Kessaci, M. E. (2021), AutoTSC: Optimization Algorithm to Automatically Solve the Time Series Classification Problem, In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 412-419.
- Rivoli, A., Garcia, L. P., Soares, C., Vanschoren, J., and de Carvalho, A. C. (2022), Meta-features for Meta-learning, *Knowledge-Based Systems*, **240**, 108101.
- Safarik, J., Jalowiczor, J., Gresak, E., and Rozhon, J. (2018), Genetic Algorithm for Automatic Tuning of Neural Network Hyperparameters, In *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything (SPIE)*, 168-174.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015), Taking the Human out of the Loop: A Review of Bayesian Optimization, *Proceedings of the IEEE*, **104**(1), 148-175.
- Toal, D. J., Bressloff, N. W., Keane, A. J., and Holden, C. M. E. (2011), The Development of a Hybridized Particle Swarm for Kriging Hyperparameter Tuning, *Engineering Optimization*, **43**(6), 675-699.
- Thomas, L. (1997), Retrospective power analysis, *Conservation Biology*, **11**(1), 276-280.
- Wolpert, D. H. and Macready, W. G. (1997), No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, **1**(1), 67-82.
- Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., and Deng, S. H. (2019), Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization, *Journal of Electronic Science and Technology*, **17**(1), 26-40.
- Zöller, M. A. and Huber, M. F. (2021), Benchmark and Survey of Automated Machine Learning Frameworks, *Journal of Artificial Intelligence Research*, **70**, 409-472.

## &lt;부록 A&gt;

Table A.1 Time-series Features Considered in the Paper

Feature	Description	Formulation	Reference
$\phi_1$	Mean	$\frac{\sum_{t=1}^T x_t}{T}$	Christ <i>et al.</i> (2018)
$\phi_2$	Standard deviation	$\sqrt{\frac{\sum_{t=1}^T (x_t - \phi_1)^2}{T}}$	Kampouraki <i>et al.</i> (2008)
$\phi_3$	Skewness	$\frac{1}{(T-1)} \frac{\sum_{t=1}^T (x_t - \phi_1)^3}{\phi_2^3}$	
$\phi_4$	Kurtosis	$\frac{1}{(T-1)} \frac{\sum_{t=1}^T (x_t - \phi_1)^4}{\phi_2^4}$	
$\phi_5$	Maximum	$\max_t x_t$	Christ <i>et al.</i> (2018)
$\phi_6$	Minimum	$\min_t x_t$	
$\phi_7 - \phi_{15}$	Percentile	$\left(\frac{p}{10}\right)$ th percentile of $x$ • $p = 1, 2, 3, \dots, 9$	
$\phi_{16}$	Root mean square	$\sqrt{\frac{\sum_{t=1}^T x_t^2}{T}}$	Kampouraki <i>et al.</i> (2008)
$\phi_{17}$	Absolute energy	$\sum_{t=1}^T x_t^2$	Christ <i>et al.</i> (2018)
$\phi_{18}$	Absolute maximum	$ \max_{t=1, \dots, T} x_t $	
$\phi_{19}$	Complexity-invariant distance	$\sqrt{\sum_{t=1}^{T-1} (x_t - x_{t-1})^2}$	Batista <i>et al.</i> (2014)
$\phi_{20}$	Normalized complexity-invariant distance	$\sqrt{\sum_{t=1}^{T-1} (z_t - z_{t-1})^2}$ • $z_t = \frac{x_t - \phi_1}{\phi_7}$	
$\phi_{21} - \phi_{25}$	Autocorrelation	$\frac{\sum_{t=1}^{T-l} (x_t - \phi_1)(x_{t+l} - \phi_1)}{\sum_{t=1}^T (x_t - \phi_1)^2}$ • $l = 1, 2, 3, 4, 5$	Bagnall <i>et al.</i> (2015) Kampouraki <i>et al.</i> (2008)
$\phi_{26} - \phi_{30}$	Partial autocorrelation	$\frac{COV(x_t, x_{t-l}   x_{t-1:t-l+1})}{\sqrt{VAR(x_t   x_{t-1:t-l+1}) VAR(x_{t-l}   x_{t-1:t-l+1})}}$ • $l = 1, 2, 3, 4, 5$ • $COV$ : covariance • $VAR$ : variance	Bagnall <i>et al.</i> (2015)
$\phi_{31} - \phi_{34}$	Cumulative mean of autocorrelation	$\frac{1}{n} \sum_{i=1}^n \phi_{20+i}$ • $n = 2, 3, 4, 5$	Christ <i>et al.</i> (2018)
$\phi_{35} - \phi_{38}$	Cumulative standard deviation of autocorrelation	$\sqrt{\frac{\sum_{i=1}^n (\phi_{20+i} - \phi_{29+n})^2}{n}}$ • $n = 2, 3, 4, 5$	
$\phi_{39}$	Count above mean	$\frac{\sum_{t=1}^T I(x_t > \phi_1)}{T}$ • $I$ : Indicator function • $I(cond) = \begin{cases} 1, & \text{if } cond \equiv True \\ 0, & \text{otherwise} \end{cases}$	Christ <i>et al.</i> (2018)
$\phi_{40}$	Count below mean	$\frac{\sum_{t=1}^T I(x_t < \phi_1)}{T}$	



Feature	Description	Formulation	Reference
$\phi_{41}$	First location of maximum	$\frac{1}{T} \times \operatorname{argmax}_t I(x_t \equiv \phi_8)$	
$\phi_{42}$	First location of minimum	$\frac{1}{T} \times \operatorname{argmax}_t I(x_t \equiv \phi_9)$	
$\phi_{43}$	Mean change	$\frac{1}{T-1} \sum_{t=1}^{T-1} x_{t+1} - x_t$	
$\phi_{44}$	Mean absolute change	$\frac{1}{T-1} \sum_{t=1}^{T-1}  x_{t+1} - x_t $	
$\phi_{45}$	Mean of the 10 largest absolute value	$\frac{1}{10} \sum_{k=1}^{10}  \tilde{x}_k $ • $ \tilde{x}_k $ : the $k$ th largest absolute value	
$\phi_{46}$	Mean second derivative central	$\frac{1}{4(T-2)} \sum_{t=1}^{T-1} x_{t+2} - 2 \cdot x_{t+1} + x_t$	
$\phi_{47} - \phi_{51}$	Third-order cumulants	$\frac{1}{T-2l} \sum_{t=1}^{T-2l} x_{t+2l} \cdot x_{t+l} \cdot x_t$ • $l = 1, 2, 3, 4, 5$	Thomas <i>et al.</i> (1997)
$\phi_{52}$	Newcomb-Benford distribution for	Correlation between $\log_{10}\left(1 + \frac{1}{d}\right)$ and $\Pr(x \leq \phi_{6+d})$ where $d$ is 1, 2, 3, ..., 9	Hill <i>et al.</i> (1995)
$\phi_{53} - \phi_{55}$	Binned entropy	$-\sum_{k=0}^{\min(b,T)} p_k \log(p_k)$ • $b = 3, 5, 10$ • $p_k$ : ratio of samples in the $k$ th bin	
$\phi_{56}$	Longest strike above mean	Length of the longest consecutive subsequence in $x$ that is bigger than $\phi_1$	
$\phi_{57}$	Longest strike below mean	Length of the longest consecutive subsequence in $x$ that is smaller than $\phi_1$	
$\phi_{58}$	Number of crossing mean	$\frac{\sum_{t=1}^{T-1} I((x_t > \phi_1)(x_{t+1} < \phi_1)) + I((x_t < \phi_1)(x_{t+1} > \phi_1))}{T}$	
$\phi_{59} - \phi_{61}$	Number of continuous wavelet transform peaks	Number of peaks whose length is bigger than $l$ obtained by continuous wavelet transform • $l = 2, 5, 10$	Christ <i>et al.</i> (2018)
$\phi_{62} - \phi_{64}$	Number of peaks	Number of peaks of at least support $n$ in the time series $x$ • $n = 2, 5, 10$	
$\phi_{65}$	Percentage of reoccurring data points	The number of points occurring more than once / the number of points (i.e., T)	
$\phi_{66}$	Percentage of reoccurring values	The number of unique values occurring more than once / the number of unique values	
$\phi_{67} - \phi_{68}$	Permutation entropy	Permutation entropy given window size • Window size = 3, 5	Bandt <i>et al.</i> (2002)
$\phi_{69} - \phi_{71}$	Ratio beyond r times sigma	$\frac{\sum_{t=1}^T I( x_t - \phi_1  > r \times \phi_2)}{T}$ • $r = 1, 2, 3$	Christ <i>et al.</i> (2018)
$\phi_{72} - \phi_{74}$	Symmetry looking	$I( \phi_1 - \phi_{11}  < r \times  \phi_5 - \phi_6 )$ • $r = 0.01, 0.05, 0.1$	
$\phi_{75} - \phi_{79}$	Time reversal asymmetry statistic	$\frac{1}{T-2l} \sum_{t=1}^{T-2l} (x_{t+2 \cdot l})^2 \cdot x_{t+l} - x_{t+l} \cdot x_t^2$ • $l = 1, 2, 3, 4, 5$	Fulcher and Jones (2014)

Feature	Description	Formulation	Reference
$\phi_{80}$	Variation coefficient	$\frac{\phi_2}{\phi_1}$	Christ <i>et al.</i> (2018)
$\phi_{81} - \phi_{83}$	empirical cumulative distribution function	Max, min and mean of empirical cumulative distribution function with 10 bins	Barandas <i>et al.</i> (2020)
$\phi_{84} - \phi_{86}$	Fast fourier transformation coefficient	Max, min and mean value of each spectrogram frequency when sampling frequency is 1	Kampouraki <i>et al.</i> (2008) Barandas <i>et al.</i> (2020)
$\phi_{87}$	Max frequency	Maximum frequency obtained by fast Fourier transform	Kampouraki <i>et al.</i> (2008)
$\phi_{88}$	Peak to peak distance	$\phi_5 - \phi_6$	Barandas <i>et al.</i> (2020)
$\phi_{89}$	Slope	Slope of the linear equation fitted to the time series $x$	Barandas <i>et al.</i> (2020)

Table A.2 Meta Features Considered in the Paper

Feature	Description	Formulation
$p_1$	Ratio of the number of features per the number of records	$\frac{d}{n}$ • $d$ : number of features • $n$ : number of records
$p_2$	Ratio of the number of classes per the number of features	$\frac{q}{d}$ • $q$ : number of classes
$p_3 - p_5$	Frequencies of the class values	$S[prob_1, \dots, prob_q]$ • $prob_k = \frac{1}{n} \sum_{j=1}^n l(y_j = c_k)$ • $c_k$ : class $k$ • $S$ : max, mean, min
$p_6$	Ratio of the number of records per the number of features	$\frac{n}{d}$
$p_7$	Ratio of the number of records per the number of classes	$\frac{n}{q}$
$p_8$	Number of features	$d$
$p_9$	Number of classes	$q$
$p_{10}$	Number of records	$n$
$p_{11}$	Number of numeric features	$\sum_{i=1}^d l(x_j \in R^n)$
$p_{12} - p_{14}$	Absolute features correlation	$S[corr_1, \dots, corr_d]$ • $corr_j$ : correlation between feature $j$ and label
$p_{15} - p_{17}$	Interquartile range of features	$S[IQR_1, \dots, IQR_d]$ • $IQR_j$ : interquartile range of feature $j$
$p_{18} - p_{20}$	Kurtosis of features	$S[kurt_1, \dots, kurt_d]$ • $kurt_j$ : kurtosis of feature $j$
$p_{21} - p_{23}$	Median absolute deviation of features	$S[mad_1, \dots, mad_d]$ • $mad_j$ : median absolute deviation of feature $j$
$p_{24} - p_{26}$	Maximum value of features	$S[max_1, \dots, max_d]$ • $max_j$ : maximum value of feature $j$
$p_{27} - p_{29}$	Mean value of features	$S[mean_1, \dots, mean_d]$ • $mean_j$ : mean value of feature $j$

Feature	Description	Formulation
$p_{30} - p_{32}$	Minimum value of features	$S[\min_1, \dots, \min_d]$ • $\min_j$ : minimum value of feature $j$
$p_{33} - p_{35}$	Number of features pairs with high correlation	$\frac{2}{d(d-1)} \sum_{i=1}^{d-1} \sum_{j=i+1}^d l( corr_{i,j}  \geq \tau)$ • $corr_{i,j}$ : correlation between feature $i$ and $j$ • $\tau$ : threshold (0.5, 0.7, 0.9)
$p_{36}$	Number of features with normal distribution.	$\sum_{j=1}^d l(pval_j \geq 0.05)$ • $pval_j$ : p-value of normal test for feature $j$
$p_{37} - p_{39}$	Standard deviation of the features	$S[sd_1, \dots, sd_d]$ • $sd_j$ : standard deviation of feature $j$
$p_{40} - p_{42}$	Skewness of features	$S[skev_1, \dots, skev_d]$ • $skev_j$ : skewness of feature $j$
$p_{43}$	Class entropy	$\sum_k \pi_k \log \pi_k$ • $\pi_k$ : the prior probability for class $k$

<부록 B>

**Table B.1** Information of the Collected Datasets

No	Purpose	Dataset	Size	Length	Number of Classes	Type
1	Meta-model training	ACSF1	200	1460	10	DEVICE
2		Beef	60	470	5	SPECTRO
3		BinaryHeartbeat	409	18530	2	AUDIO
4		Car	120	577	4	SENSOR
5		CatsDogs	275	14773	2	AUDIO
6		CBF	930	128	3	SIMULATED
7		Chinatown	365	24	2	TRAFFIC
8		Coffee	56	286	2	SPECTRO
9		Computers	500	720	2	DEVICE
10		DodgerLoopDay	158	288	7	SENSOR
11		DodgerLoopGame	158	288	2	SENSOR
12		DodgerLoopWeekend	158	288	2	SENSOR
13		Earthquakes	461	512	2	SENSOR
14		ElectricDevices	16637	96	7	DEVICE
15		EOGHorizontalSignal	724	1250	12	EOG
16		EOGVerticalSignal	724	1250	12	EOG
17		EthanolLevel	1004	1751	4	SPECTRO
18		FordA	4921	500	2	SENSOR
19		FordB	4446	500	2	SENSOR
20		FreezerRegularTrain	3000	301	2	SENSOR
21		FreezerSmallTrain	2878	301	2	SENSOR
22		Fungi	204	201	18	OTHER
23		Ham	214	431	2	SPECTRO
24		HouseTwenty	135	3000	2	DEVICE
25		InsectEPGRegularTrain	311	601	3	EPG

No	Purpose	Dataset	Size	Length	Number of Classes	Type	
26		ItalyPowerDemand	1096	24	2	SENSOR	
27		LargeKitchenAppliances	750	720	3	DEVICE	
28		Lightning2	121	637	2	SENSOR	
29		Lightning7	143	319	7	SENSOR	
30		Mallat	2400	1024	8	SIMULATED	
31		Meat	120	448	3	SPECTRO	
32		MelbournePedestrian	3633	24	10	TRAFFIC	
33		MoteStrain	1272	84	2	SENSOR	
34		NonInvasiveFetalECGThorax1	3765	750	42	ECG	
35		NonInvasiveFetalECGThorax2	3765	750	42	ECG	
36		OliveOil	60	570	4	SPECTRO	
37		PigAirwayPressure	312	2000	52	HEMODYNAMICS	
38		PigArtPressure	312	2000	52	HEMODYNAMICS	
39		PigCVP	312	2000	52	HEMODYNAMICS	
40		PLAID	1074	325 (average length)	11	DEVICE	
41		Plane	210	144	7	SENSOR	
42		PowerCons	360	144	2	DEVICE	
43		RefrigerationDevices	750	720	3	DEVICE	
44		RightWhaleCalls	12896	4000	2	AUDIO	
45		Rock	70	2844	4	SPECTRO	
46		ScreenType	750	720	3	DEVICE	
47		SemgHandGenderCh2	900	1500	2	SPECTRO	
48		SemgHandMovementCh2	900	1500	6	SPECTRO	
49		SemgHandSubjectCh2	900	1500	5	SPECTRO	
50		ShakeGestureWiimoteZ	100	167 (average length)	10	SENSOR	
51		ShapeletSim	200	500	2	SIMULATED	
52		SharePriceIncrease	1930	60	2	FINANCIAL	
53		SmallKitchenAppliances	750	720	3	DEVICE	
54		SmoothSubspace	300	15	3	SIMULATED	
55		SonyAIBORobotSurface1	621	70	2	SENSOR	
56		SonyAIBORobotSurface2	980	65	2	SENSOR	
57		Strawberry	983	235	2	SPECTRO	
58		SyntheticControl	600	60	6	SIMULATED	
59		Trace	200	275	4	SENSOR	
60		TwoLeadECG	1162	82	2	ECG	
61		TwoPatterns	5000	128	4	SIMULATED	
62		UMD	180	150	3	SIMULATED	
63		Wafer	7164	152	2	SENSOR	
64		Wine	111	234	2	SPECTRO	
65		Test	AbnormalHeartbeat	606	3053	5	AUDIO
66			ChlorineConcentration	4307	166	3	SIMULATED
67			ECGFiveDays	884	136	2	ECG
68			InsectEPGSmallTrain	266	601	3	EPG

## 저자소개

**류서현** : 한양대학교 산업경영공학과에서 학사학위를 취득하고 동대학원에서 석사과정에 재학 중이다. 주요 관심분야는 기계학습 자동화, 특징 선택, 데이터 마이닝이다.

**이다경** : 한양대학교 산업경영공학과에서 학사학위를 취득하고 동대학원에서 석사과정에 재학 중이다. 주요 관심분야는 데이터 마이닝, 시계열 제조 데이터 특징선택, 기계학습의 하이퍼 파라미터 최적화이다.

**안길승** : 한양대학교 산업경영공학과에서 학사, 박사 학위를 취득하고 현재 현대자동차 빅데이터실에 재직 중이다. 주요 관심분야는 기계학습 자동화, 개인화 모형, 예방 정비이다.

**허 선** : 서울대학교 산업공학과에서 학사, 석사를 취득하고 텍사스 A&M 대학교에서 산업공학 박사학위를 취득하였다. 주요 관심분야는 클라우드 제조 시스템, 제조 데이터 분석, 데이터 전처리 기법, 확률 그래프 모형이다.