



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

회분식 공정의 품질 변수 추정에
대한 기계 학습 기반 접근법

Machine Learning Based Approaches to
Estimation of Quality Variables in Batch
Processes

2023년 2월

서울대학교 대학원

화학생명공학부

박준수

회분식 공정의 품질 변수 추정 에 대한 기계 학습 기반 접근법

지도교수 이 종 민

이 논문을 공학석사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

화학생물공학부

박 준 수

박준수의 공학석사 학위논문을 인준함

2022년 12월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

초 록

안전하고 경제적인 배치 공정의 운전 방향을 제시하기 위해서 제품 품질 예측 모델의 개발이 필요하다. 하지만 배치 공정은 비정상상태에서 운전되어 강한 비선형성을 보이기 때문에 물리 법칙 기반 모델의 구축이 어렵다. 또한 각 배치 별로 서로 다른 운전 시간을 가지고 운전 조건에 따라 여러 개의 phase로 구성되는 배치 공정의 특성은 품질 예측 모델의 개발을 더욱 어렵게 만든다. 따라서 본 연구에서는 서로 다른 운전 시간을 갖는 배치 공정에 대해서 공정 운전의 phase를 고려한 데이터 기반 품질 예측 모델을 개발하였다. 공정 데이터의 시계열성을 고려한 phase 분할을 위해서 warped K-means clustering (WKM) 알고리즘을 적용하여 phase를 분할하였다. 분리된 각각의 phase에 대하여 별도의 Recurrent neural network (RNN) 셀을 학습시키는 형태의 품질 예측 모델을 개발하였다. 개발된 품질 예측 모델은 기존의 대표적인 방법론인 Dynamic time warping (DTW)과 Multiway partial least squares (MPLS)를 사용한 경우보다 좋은 성능을 보였다. 또한 phase를 구분하지 않고 하나의 RNN 셀을 사용한 경우보다 phase 별로 서로 다른 RNN 셀을 사용하는 경우에 더 좋은 성능을 보이는 것을 확인하였다. 추가적으로 해당 품질 예측 모델을 온라인으로 활용 가능한 형태로 발전시킴으로써 공정 운전 중 발생하는 이상 진단 및 최적 운전 조건의 탐색에 활용할 수 있을 것이다.

주요어 : 배치 공정, 품질 예측 모델, Warped K-means clustering, Recurrent neural network

학번 : 2021-25124

목 차

제 1 장 서 론	5
제 2 장 방 법 론.....	12
제 1 절 배치 공정 데이터의 전처리.....	12
제 2 절 Dynamic Time Warping	18
제 3 절 Multiway Partial Least Squares	20
제 4 절 Phase Partition.....	21
제 1 항 Sub-PCA 알고리즘	21
제 2 항 Warped K-Means Clustering 알고리즘	23
제 5 절 Recurrent Neural Network	26
제 1 항 Vanilla RNN, LSTM, GRU.....	26
제 2 항 Multi RNN 모델	31
제 3 장 페니실린 생산 공정	33
제 4 장 결과 및 분석.....	38
제 1 절 DTW-MPLS 모델	39
제 2 절 Single RNN 모델.....	45
제 3 절 Multi RNN 모델	52
제 1 항 Phase Partition.....	52
제 2 항 Multi RNN 모델	59
제 5 장 결 론	70

그림 목차

그림 1. 3차원 형태의 배치 공정 데이터에 대한 VARIABLE-WISE UNFOLDING과 BATCH-WISE UNFOLDING의 시각화..... 13

그림 2. (A) VANILLA RNN, (B) LSTM, (C) GRU의 구조..... 27

그림 3. MULTI RNN의 구조..... 32

그림 4. 페니실린 생산 BIOREACTOR 모식도..... 35

그림 5. 공정 운전 시간 분포..... 36

그림 6. DTW 기법을 통한 정렬 (A)전, (B)후의 반응기 내부 부피 변화..... 40

그림 7. MPLS 모델의 주성분 개수에 따른 (A) R2 SCORE, (B) MSE 값의 변화..... 43

그림 8. 테스트 데이터셋에 대한 DTW-MPLS 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT..... 44

그림 9. 테스트 데이터에 대한 SINGLE RNN 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT..... 47

그림 10. 테스트 데이터에 대한 SINGLE LSTM 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT..... 48

그림 11. 테스트 데이터에 대한 SINGLE GRU 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT..... 49

그림 12. SINGLE RNN 모델의 예측 결과에 따른 (A)RMSE, (B) MAPE, (C) R2 SCORE 값의 비교..... 51

그림 13. SUB-PCA 기법을 사용하여 클러스터의 개수를 (A)3개, (B)4개, (C)5개로 설정했을 때 군집화 결과..... 54

그림 14. 분할하는 PHASE의 개수에 따른 $LOG(SQE)$ 값의 변화..... 56

그림 15. 분할하는 PHASE의 개수에 따른 PPCI 값의 변화..... 57

그림 16. WKM 알고리즘을 적용하여 PHASE를 분할한 결과.....	58
그림 17. 테스트 데이터에 대한 MULTI RNN 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT.....	60
그림 18. 테스트 데이터에 대한 MULTI LSTM 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT.....	61
그림 19. 테스트 데이터에 대한 MULTI GRU 모델의 (A)품질 예측 결과, (B)예측 결과에 따른 PARITY PLOT.....	62
그림 20. 모델 별 (A)RMSE, (B)MAPE, (C) R2 SCORE의 비교	66
그림 21. 전체 배치 데이터에 대해서 MULTI LSTM 모델로부터 구한 각 PHASE 별 HIDDEN STATE의 값.....	68
그림 22. 전체 배치 데이터에 대해서 MULTI LSTM 모델로 구한 PHASE 별 HIDDEN STATE의 L2 NORM의 평균값	69

표 목차

표 1. 페니실린 생산 bioreactor의 공정 변수.	37
표 2. Single RNN 모델의 예측 오차.	50
표 3. 각 모델의 품질 예측 과정에서 발생한 오차 비교.	65

제 1 장 서 론

배치 공정은 고분자, 반도체, 의약품 등 다양한 산업 분야에서 고부가 가치를 가지는 제품을 생산하기 위해서 널리 사용되고 있다 [1]. 배치 공정의 제품 품질은 운전이 종료된 이후에 측정되는데, 공정의 복잡도, 장비의 이상, 원료 품질의 변화, 운전 조건의 변화 등의 이유로 인해서 레시피에서 주어진 제품 품질과 큰 차이를 보일 수 있다 [2]. 배치 공정의 제품 품질을 추정하는 모델을 개발한다면 제품 품질을 저하시키는 잘못된 운전 조건 혹은 이상 원인을 파악하여 안전하고 경제적인 공정 운전의 방향성을 제시하는 데에 적용될 수 있다. 따라서 배치 공정의 제품 품질 추정 모델은 제품의 품질 향상과 더불어 안전한 공정 운전과 경제성 향상에 도움을 줄 수 있다 [3,4]. 하지만 배치 공정은 비정상상태에서 운전되어 시간에 따른 운전 상태의 변화가 크고 공정의 동특성이 강한 비선형성을 보이기 때문에 배치 공정의 분석은 많은 어려움을 겪는다. 또한 배치 운전 별로 운전 시간이 다르고 운전 조건에 따라서 여러 개의 phase로 구성된다는 특성으로 인해 배치 공정의 모니터링 및 품질 추정은 도전적인 과제이다.

품질 추정 모델은 물리 법칙을 기반으로 하는 화이트 박스 모델 (white box model)과 데이터를 기반으로 하는 블랙 박스 모델 (black box model)로 구분할 수 있다. 배치 공정은 비정상상태에서 운전되어 복잡한 동특성을 보이기 때문에 공정의 물리 법칙을 기반으로 하는 화이트 박스 모델로부터 제품 품질을 정밀하게 추정하는 방법론을 구축하는 것은 어려운 일이다. 반면에 데이터 기반의 블랙 박스 모델은 공정 메커니즘에 대해서 완벽하게 이해하지 못하더라도 측정된 공정 데이터를 활용하여 품질을 추정하는 방법론을 구축할 수 있기 때문에 데이터 기반의 모델링과 모니터링 기법은 최근 많은 관심을 받고 있다

[5~7]. 배치 공정의 공정 데이터는 연속 공정의 공정 데이터와는 다르게 배치, 변수, 시간으로 구성된 3차원 형태를 가지는데, 이를 고려하여 공정 정보를 분석하기 위해서 multivariate statistical process monitoring (MSPM) 혹은 multivariate statistical process control (MSPC)이라고 불리는 통계적 기법을 사용하여 배치 공정의 특성을 고려한 소프트 센서가 연구되었다 [8~10]. 특히 초기 연구로써 제안된 multiway principal component analysis (MPCA)와 multiway partial least squares (MPLS)는 이후에 다양한 후속 연구가 진행되면서 배치 공정 분석의 de-facto standard로써 자리 잡았다 [11~16].

배치 공정을 분석하는 많은 연구에서 각각의 배치 운전이 동일한 운전 시간을 가진다고 가정하지만, 실제 배치 공정에서 각 배치 별 공정 운전 시간은 동일하지 않다. 공정 운전 중에 나타나는 외란, 초기 운전 조건의 차이, 그리고 공정 운전 중에 발생하는 운전 조건의 변화 등의 다양한 요인으로 인해서 각각의 배치는 서로 다른 운전 시간을 가진다. 각 배치 공정마다 운전 시간이 다르게 나타나는 문제를 다루기 위해서 trajectory alignment 혹은 synchronization 기법을 적용하여 배치 공정의 시간 축을 정렬하는 방법이 연구되었다 [17~24]. [17,18]에서는 시간의 흐름에 따른 변화의 경향성이 일정한 공정 변수를 시간 축 대신 사용하여 공정 변수를 정렬하는 indicator variable technique (IVT)을 적용했다. 하지만 정렬을 위한 기준으로 사용할 indicator variable을 선정하기 위해 공정에 대한 전문적인 지식이 필요하고, indicator variable로 사용할 수 있는 변수가 존재하지 않는 경우에는 사용할 수 없다는 단점이 존재한다. 또한 크로마토그래피에서 나타나는 피크를 정렬하기 위해 고안된 방법인 correlation optimization warping (COW) [19]과 음성 인식 분야에서 데이터 정렬을 위해

고안된 dynamic time warping (DTW) [22]과 같은 time warping 기법을 배치 공정에 적용하여 기준이 되는 운전 경로에 대해서 서로 다른 길이의 공정 데이터의 시간 축을 정렬하는 방법이 연구되었다 [19~24]. 하지만 이와 같은 time warping 기법을 사용하면 공정 운전의 궤적이 늘어나거나 줄어들면서 공정에 대한 정보 혹은 공정 변수 사이의 관계가 왜곡될 수 있고, 실시간으로 주어지는 공정 정보에 대해서는 사용하기 어렵다는 한계를 가지고 있다 [25,26].

배치 공정은 공정 운전 조건의 변화에 따라서 여러 개의 phase로 분리할 수 있다. 이때 여러 개의 phase로 구성된 공정은 phase에 따라서 공정 데이터의 상관 관계가 크게 변화하기 때문에 각각의 배치 운전이 하나의 phase로 구성되어 있다고 가정하는 기존의 배치 공정 분석 방법을 적용할 경우 낮은 성능을 보인다 [25]. 특히 [27]과 [28]의 연구에서는 공정의 phase를 분할하여 모니터링하는 경우에 그렇지 않은 경우보다 더 좋은 성능을 보이는 것을 확인했다. 배치 공정의 공정 데이터를 phase 별로 분할하기 위해서 전문가의 지식을 활용하거나 공정 데이터로부터 indicator variable을 선정하여 phase를 나누는 방법을 고려할 수 있다 [29]. 하지만 이와 같이 공정에 대한 전문적인 지식을 기반으로 하는 phase 분할 방법은 배치 공정의 phase를 분할하기 위해 필요한 정보가 충분히 주어지지 않은 경우 혹은 공정의 특성을 완벽하게 알고 있지 않은 경우에는 적용할 수 없다는 한계를 가진다. 최근에는 공정 운전 중에 측정할 수 있는 데이터를 기반으로 한 phase 분할 방법이 활발하게 연구되고 있다 [30~32]. 데이터를 기반으로 phase를 분할하는 방법은 시간 별로 측정되는 공정 데이터에서 공정 변수의 상관관계가 phase 마다 다르게 나타난다는 특성을 이용하여 각 phase를 분할한다. [30]에서 제시된 sub-PCA는

각 시간에 대해서 주어지는 2차원의 공정 데이터에 principal component analysis (PCA)를 적용하고 K-means clustering 기법을 사용하여 높은 상관관계를 가지는 시점 별로 phase를 분할했다. [31]은 PCA를 적용하여 top-down 단계에서 각 phase를 분할하고 bottom-up 단계에서 phase의 개수를 줄여 최종적으로 분할된 phase를 구하는 방법을 제안했다. 또한 [32]에서는 Gaussian mixture model (GMM)을 3차원의 배치 공정 데이터에 대한 hybrid unfolding 기법과 결합하여 phase를 분할하는 방법을 제안했다. 하지만 이와 같은 기존 연구들은 배치 공정의 전체 운전 시간과 각 phase 별 지속 시간이 모두 동일하다고 가정하기 때문에 전체 운전 시간과 phase의 지속 시간이 일정하지 않은 배치 공정에 대해서는 적용할 수 없다는 한계를 가진다.

공정 운전 시간이 동일하지 않고 여러 개의 phase로 구분된 배치 공정의 공정 데이터 분석을 위해서 다양한 연구가 수행되었다 [33~37]. [33]에서는 공정 운전 시간이 동일하다고 가정한 기존의 sub-PCA 모델을 수정하여 반복적으로 sub-PCA 기법을 적용함으로써 phase를 분할하는 모델을 제안했고, [34]에서는 inner-phase analysis를 기반으로 한 phase 분할 모델을 제안했다. 하지만 이와 같은 방법론들은 일반적인 K-means clustering 기법을 적용한 phase 분류 방법으로써, 배치 공정의 공정 데이터가 가지고 있는 시계열성을 고려하지 못하기 때문에 각 phase에 서로 연결되지 않은 시간대의 데이터가 포함될 수 있다. [35,36]에서는 각각 support vector data description (SVDD) 기법과 stepwise sequential phase partition (SSPP) 기법을 활용하여 배치 공정의 운전 시간에 상관없이 공정 데이터로부터 phase를 구분하는 방법론을 제시했다. 또한 [37]에서는 시계열 데이터의 순차성을 고려한 warped K-means (WKM)

clustering 기법을 활용하여 배치 공정 데이터의 3차원 형태를 유지하면서 공정 운전 시간에 관계없이 phase를 분할하는 방법을 제안했다.

연속 공정은 장기간에 걸친 정상상태를 기반으로 하여 운전되지만, 배치 공정은 운전되는 기간 동안 끊임없이 물리적, 화학적인 변화를 겪기 때문에 배치 공정을 모델링하기 위해서는 정상상태가 아닌 동적 변화를 고려해야 한다. 특히 배치 공정은 운전 조건에 따라서 phase의 변화가 자주 나타나기 때문에 배치 공정의 동특성은 강한 비선형성을 보이게 된다. 또한 배치 공정의 공정 데이터는 다변수로 구성되어 있고, 각각의 변수가 강한 상관관계를 가지고 있기 때문에 공정 변수 분석에 어려움을 더한다. 따라서 선형성을 가정하는 MPCA, MPLS 기반의 초기 연구들은 배치 공정의 동특성으로 인한 공정 데이터의 비선형성을 다루지 못한다는 한계점을 가진다. [38~41]에서는 배치 공정의 비선형성을 다루기 위해서 kernel을 활용한 kernel principal component analysis (KPCA), kernel partial least squares (KPLS) 기법을 활용한 배치 공정의 모니터링 및 품질 예측 모델을 제안했다. 이외에도 산업계에 존재하는 다양한 배치 공정의 비선형성을 다루기 위해서 artificial neural networks (ANN) [42,43], least absolute shrinkage and selection operator regression (LASSO regression) [44], support vector machine (SVM) [45,46], relevance vector machine (RVM) [47,48], 그리고 Gaussian process regression (GPR) [49~51] 등의 방법론을 사용한 배치 공정의 모니터링에 관한 연구가 진행되었다. 하지만 이와 같은 방법론은 공정 데이터가 가지는 시계열성을 고려하지 못한다는 한계를 가진다.

각 배치의 운전 시간이 균일하지 않은 배치 공정에 대해서 공정

데이터가 가지는 비선형성과 시계열성을 고려한 모니터링을 위해서 recurrent neural network (RNN)의 활용을 고려할 수 있다. RNN은 시간의 흐름에 따라 변화하는 데이터를 분석하기 위해 고안된 인공신경망의 일종으로, 필기 인식 (handwriting recognition) [52], 음성 인식 (speech recognition) [53], 자연어 처리 (natural language process) [54] 등의 분야에서 시계열 데이터의 분석을 위해서 주로 사용된다. RNN은 인공 신경망의 일종이기 때문에 비선형성을 보이는 시스템을 다룰 수 있고, 입력되는 시퀀스의 길이가 일정하지 않아도 사용할 수 있기 때문에 운전 시간이 일정하지 않은 배치 공정의 모니터링을 위한 모델에 사용할 수 있다. [55]에서는 배치 공정의 운전 과정을 모사하는 모델을 위해 RNN을 활용하는 방법을 제안했고, [56]에서는 이전 배치 데이터에서 발생한 모델의 오차를 활용하여 현재 배치 데이터에 대한 RNN 모델의 오차를 보정하는 방법을 제안했다. 또한 [57]에서는 배치 공정의 system identification을 위해서 RNN의 일종인 LSTM을 활용하는 방법을 제안했다.

배치 공정의 품질 추정에 사용되는 입력 데이터는 배치 운전 기간 동안 누적된 공정 정보이지만, 출력 데이터로 사용되는 제품의 품질 데이터는 한번의 배치 공정이 끝났을 때에 얻을 수 있는 값이기 때문에 입력 데이터와 출력 데이터 사이에서 차원의 불균형이 나타난다. 따라서 배치 공정의 제품 품질을 추정하는 모델을 개발하기 위해서 공정 데이터의 차원을 축소하여 제품 품질과 관련된 공정 정보를 담고 있는 hidden feature를 찾아내는 방법이 필요하다. 이때 RNN을 활용하여 공정 데이터의 시계열성을 고려하면서 입력 데이터의 차원을 축소하여 hidden feature를 추출하는 방법을 고려할 수 있고, [58~61]에서 RNN의 일종인 LSTM을 활용하여 배치 공정의 hidden feature를

추출하는 방법을 제안했다. 특히 [58]에서는 LSTM을 stacked autoencoder (SAE)와 결합하여 배치 공정의 multi-phase를 고려한 품질 예측 모델을 개발했고, [59]에서는 LSTM 기반의 just-in-time-learning (JITL)을 활용한 품질 예측 모델을 개발했다.

본 연구에서는 생물 반응기의 배치 공정에 대해서 운전 시간이 동일하지 않은 공정 데이터를 활용하여 여러 개의 phase를 고려한 기계 학습 기반의 품질 변수 추정을 탐구해보았다. 데이터를 phase 별로 분할하기 위해서 데이터의 순차성을 고려한 WKM 알고리즘을 적용하여 배치 공정의 데이터를 여러 개의 phase로 분할하였다. 이후에 Vanilla RNN, LSTM, 그리고 GRU를 활용하여 여러 개의 phase에 대해서 적용할 수 있는 형태의 RNN 모델을 구성하여 제품 품질 변수를 추정하는 모델을 개발했다. 특히 배치 공정을 구성하는 여러 개의 phase를 다루기 위해서 각 phase 별로 구분된 RNN 셀을 활용하여 공정의 phase를 고려하지 않는 경우와 비교하고 결과를 분석해보았다.

본 논문의 구성은 다음과 같다. 2장에서는 배치 공정에서 주어지는 3차원 데이터를 전처리 하는 방법, 배치 공정의 공정 데이터를 정렬하고 분석하는 기존의 방법론, 그리고 phase를 분할하는 방법과 multi-phase를 고려한 RNN 기반 품질 예측 모델의 구조에 대해서 설명한다. 3장에서는 대상 시스템으로 사용한 페니실린 생산 공정에 대해 소개한다. 마지막으로 4장에서 대상 공정에 대한 phase 분할 결과를 보이고, RNN의 종류와 구성에 따른 품질 변수 추정 방법론의 성능을 비교 및 분석한다.

제 2 장 방 법 론

제 1 절 배치 공정 데이터의 전처리

배치 공정의 공정 데이터는 배치(I), 변수(J), 시간(K)으로 구성된 삼차원 행렬 형태 $\mathbf{X}(I \times J \times K)$ 로 구성되어 있기 때문에 기존의 연속 공정의 공정 데이터를 분석하는 방법을 동일하게 적용할 수 없다. 따라서 삼차원의 행렬을 이차원 형태로 unfolding하는 과정이 필요하다. 배치 공정의 데이터를 unfolding 하는 방법으로는 batch-wise unfolding과 variable-wise unfolding의 두가지 방법이 존재한다 [25]. 각각의 unfolding 방법은 그림 1과 같이 표현할 수 있다.

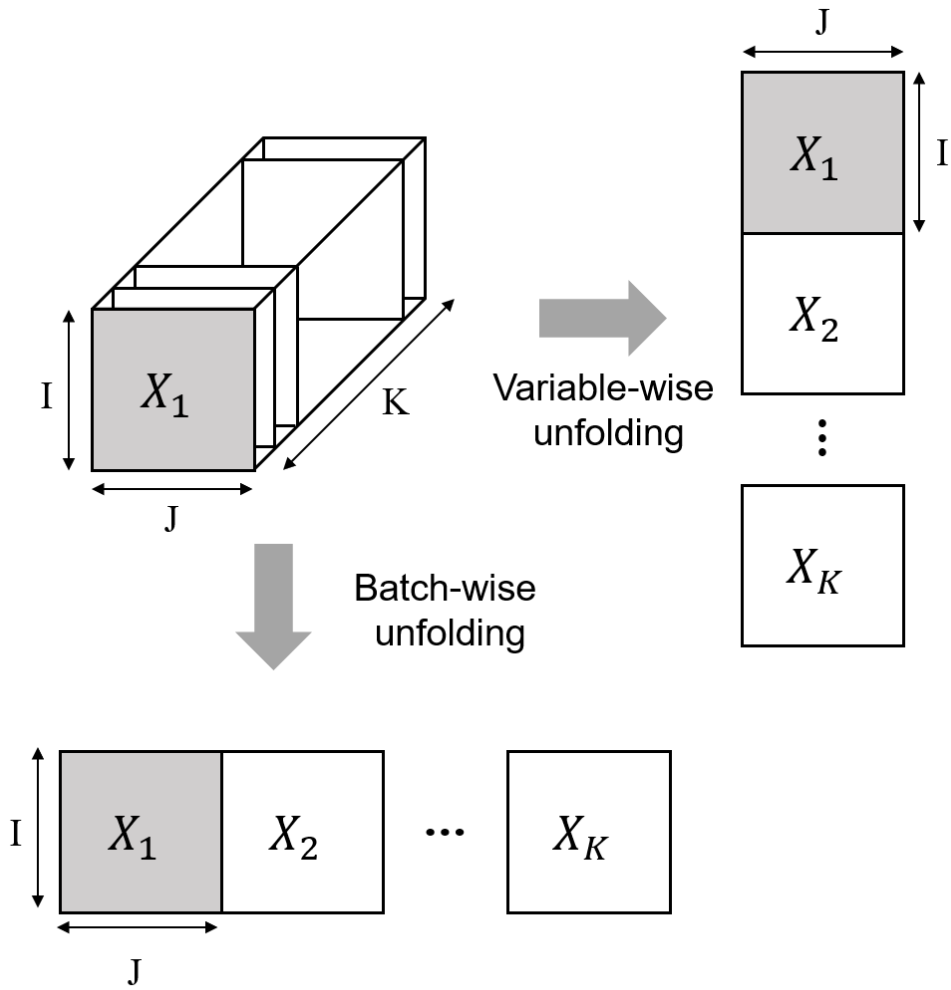


그림 1. 3차원 형태의 배치 공정 데이터에 대한 variable-wise unfolding과 batch-wise unfolding의 시각화.

Batch-wise unfolding은 배치 방향의 차원을 유지하고 변수와 시간 방향의 공정 정보를 통합하여 $\mathbf{X}(I \times JK)$ 형태로 unfolding 하는 방법이다. Batch-wise unfolding은 배치와 배치 사이에서 나타나는 공정 변수와 시간의 관계를 분석할 때 사용할 수 있기 때문에 MPCA, MPLS와 같은 기존의 데이터 분석 방법론에 주로 적용되었다. 하지만 한 배치 안에서 나타나는 공정 변수의 경로에 대한 분석이 불가능하고, 반응 중에 나타나는 여러 phase를 고려할 수 없다는 단점이 있다. 또한 batch-wise unfolding은 각 배치의 운전 시간이 일정하지 않은 경우에는 적용할 수 없고, 상대적으로 많은 배치에 대한 공정 데이터가 필요하다.

반면에 variable-wise unfolding은 변수 방향의 차원을 유지하고 배치와 시간 방향의 공정 정보를 통합하여 $\mathbf{X}(J \times IK)$ 형태로 unfolding 하는 방법이다. Variable-wise unfolding은 각 시점마다 주어지는 공정 정보를 하나의 객체로 설정하여 삼차원 행렬을 unfolding 한다. 이때 variable-wise unfolding은 배치와 배치 사이에서 나타나는 공정 운전의 차이를 고려할 수 없다는 문제를 가지고 있다. 또한 variable-wise unfolding은 지연된 측정값에 대해 고려하지 않기 때문에 auto-correlation 혹은 lagged cross-correlation 문제를 다룰 수 없다는 단점이 있다. 하지만 batch-wise unfolding과는 다르게 variable-wise unfolding은 여러 배치의 운전 시간이 다른 경우에도 적용할 수 있고, 상대적으로 배치 데이터가 적어도 사용할 수 있다는 장점을 가지고 있다. 특히 variable-wise unfolding은 각 공정 변수의 경로에 대한 정보가 손실되지 않기 때문에 phase를 구분하는 방법론에 주로 사용된다.

다변수로 구성된 공정 데이터를 분석하기 위해서는 각 공정 데이터의 단위와 크기에 의한 영향을 제거하기 위해서 데이터를 정규화

해주어야 한다. 먼저 공정 데이터를 centering 해주는데, 배치 공정은 비정상상태에서 운전하기 때문에 연속 공정과 같이 정상상태에서의 값을 사용하여 centering 해줄 수 없다. 따라서 전체 공정 기간 동안의 평균값을 사용하는 mean value centering과 각 공정 운전 시점의 평균값을 사용하는 mean trajectory centering을 사용할 수 있다. Mean value centering은 공정 운전의 경로를 왜곡하지 않고, 운전 시간이 균일하지 않은 배치 공정에도 사용할 수 있다는 특징이 있다. 식 (1)과 같이 전체 공정 데이터에 대해서 j 번째 변수의 평균값을 사용하여 centering 할 수 있다.

$$\bar{x}_{i,j,k} = x_{i,j,k} - \frac{1}{IK} \sum_{i=1}^I \sum_{k=1}^K x_{i,j,k} \quad (1)$$

이때 i, j, k 는 각각 배치, 공정 변수, 시간의 인덱스를 의미하고, $x_{i,j,k}$, $\bar{x}_{i,j,k}$ 는 각각 centering 전, 후의 공정 변수를 의미한다. 한편 mean trajectory centering은 평균적인 운전 경로에서 떨어진 정도를 고려하는 방법이기 때문에 공정 변수가 가지는 비선형성을 다룰 수 있다. 하지만 mean trajectory centering은 운전 경로를 하나의 객체로 고려하기 때문에 공정 운전 시간이 균일하지 않거나 multiphase를 고려하는 경우에는 사용할 수 없다는 단점이 있다. Mean trajectory centering에서 j 번째 변수의 평균값을 사용한 centering은 식 (2)와 같이 표현할 수 있다.

$$\bar{x}_{i,j,k} = x_{i,j,k} - \frac{1}{I} \sum_{i=1}^I x_{i,j,k} \quad (2)$$

평균값을 사용하여 centering을 마친 공정 데이터에 대해서 표준편차를 나누어 줌으로써 각 공정 데이터의 분산이 동일하도록 만들어준다. 이때 unfolding 한 데이터의 2차원 행렬에서 열 방향에

대해서 표준 편차를 계산한다. 2차원 행렬에서 계산한 열 방향의 표준 편차를 σ 라고 했을 때, 공정 데이터의 정규화를 식으로 표현하면 식 (3)과 같다.

$$\hat{x}_{i,j,k} = \frac{\bar{x}_{i,j,k}}{\sigma} \quad (3)$$

하지만 전체 데이터를 unfolding 한 2차원 행렬의 표준 편차로 정규화를 하게 되면 multiphase 특성을 가지는 배치 공정의 데이터에서 각 phase의 데이터가 가지는 분산을 다루지 못한다는 문제점이 있다. 이와 같은 문제를 해결하기 위해서는 phase를 분리한 뒤에 각 phase의 데이터에 대해서 데이터를 unfolding 하고 phase 별 2차원 행렬에 대해서 표준 편차를 구하고 나누어 주는 방식으로 정규화를 해주어야 한다.

본 연구에서는 phase 분리와 품질 추정 모델의 학습 과정에 대해서 서로 다른 방식의 unfolding과 정규화 방법을 적용했다. 배치 별로 다른 운전 시간을 가지는 공정 데이터에 대해서 phase를 분리하기 위해서 각 배치 공정의 운전 시간과 상관없이 적용할 수 있는 variable-wise unfolding을 적용하여 3차원 데이터를 unfolding 했고, 마찬가지로 균일하지 않은 운전 시간에 대해서 사용할 수 있는 mean value centering 기법을 사용하여 데이터를 정규화 했다. 정규화 된 데이터에 대해서 각 배치 공정의 phase를 분리한 뒤에, 품질 추정 모델을 학습시키기 위해서, 각각의 phase에 대해서 따로 variable-wise unfolding을 적용하여 phase의 개수만큼의 2차원 행렬을 구성하고 각 phase의 데이터에 대해서 mean value centering을 적용하여 데이터의 전처리를 진행했다. 또한 DTW와 MPLS를 활용한 품질 예측 모델을 구성하였는데, DTW 기법을 사용하여 시간 축을 정렬하기 이전에는 variable-wise unfolding을 사용하여 정규화를 사용했고, 정렬된

데이터에 대해서 MPLS 모델을 구성할 때는 batch-wise unfolding 및 정규화를 사용했다.

제 2 절 Dynamic Time Warping

시계열 데이터 분석 방법의 일종인 dynamic time warping (DTW)은 두개의 시퀀스 정보에서 유사도를 측정하여 패턴을 연결하는 방법이다. 음성 인식, 화자 구분 등의 분야에서 주로 활용되었으며, 화학 공학 분야에서는 배치 별로 운전 시간이 서로 다른 배치 공정의 시간 축을 정렬하기 위해서 사용되었다 [22~24, 62]. DTW 기법은 전체 공정의 운전 경로를 알고 있을 때, 기준이 되는 운전 경로와 다른 배치의 공정 운전 경로에서 시퀀스 사이의 유사도를 계산하여 시간 축을 정렬한다.

DTW 기법은 symmetric DTW와 asymmetric DTW로 구분할 수 있다. Symmetric DTW는 두 시퀀스 데이터의 모든 데이터 포인트를 포함하여 배치 공정의 운전 경로를 확장하는 방식으로 시간 축을 정렬하며, 기존의 두 시퀀스 데이터보다 긴 시퀀스를 생성한다. 반면에 asymmetric DTW는 원래 데이터에서 일부 데이터 포인트를 생략함으로써 기준이 되는 운전 경로의 길이와 동일하게 다른 시퀀스의 길이를 정렬한다. [22]의 연구에서는 운전 시간이 서로 다른 배치 공정의 공정 데이터를 정렬하기 위해서 기준이 되는 운전 경로를 선정하고, 각 배치의 공정 데이터를 symmetric DTW로 정렬하여 공정 데이터를 최대한 보존한 뒤에 asymmetric DTW를 적용하여 시간 축을 정렬하는 방법을 제안하였다.

DTW를 사용하여 두 경로 $T = (x_1^T, \dots, x_t^T)$ 와 $R = (x_1^R, \dots, x_r^R)$ 사이의 거리를 최소화하도록 정렬하는 방법은 다음과 같다. 먼저 두 경로의 인덱스를 K 개의 포인트로 연결하는 맵핑 F 를 식 (4)와 같이 정의한다.

$$F = [c(1), c(2), \dots, c(K)] \quad (4)$$

$$c(k) = [i(k), j(k)], 1 \leq k \leq K$$

처음에 주어지는 경로 T 와 R 의 길이는 다를 수 있지만, 시간 축이 정렬된 $T(i(k)), R(j(k))$ 는 동일한 길이 K 를 가지게 된다. Symmetric DTW는 K 가 t, r 보다 큰 값을 가지고, asymmetric DTW에서는 K 를 reference trajectory의 길이 r 과 동일하도록 설정한다. $i(k), j(k)$ 인덱스를 가지는 두 지점의 거리를 $d(i, j)$ 라고 할 때, 유클리드 거리를 사용하여 두 경로 T 와 R 사이의 거리를 정의하면 식 (5)와 같이 표현할 수 있다.

$$d(i, j) = \|R(i) - T(j)\|_2 \quad (5)$$

$$D(T, R) = \frac{\sum_{k=1}^K d(i(k), j(k))w(k)}{\sum_{k=1}^K w(k)}$$

이때 $w(k)$ 는 두 지점 사이의 거리의 중요도를 나타내는 가중치이며, 본 연구에서는 1의 값을 사용했다. 맨해튼 거리, 마할라노비스 거리 등의 다른 거리 지표를 활용하여 변형된 형태의 DTW를 적용할 수도 있다.

$$F^* = \arg \min_F D(T, R) \quad (6)$$

최종적으로 식 (6)과 같이 두 경로의 거리를 최소화하는 맵핑 F^* 를 구하여 시간 축을 정렬한다. 이때 최적 맵핑 F^* 는 제약 조건이 있는 동적 최적화 문제를 풀어서 계산할 수 있다.

제 3 절 Multiway Partial Least Squares

부분 최소 제곱법 (partial least squares, PLS)은 차원 축소 기법의 일종으로, 입력 데이터 X 와 출력 데이터 Y 에 대해서 X 와 Y 사이의 상관관계가 가장 커지는 잠재 공간으로 투영하는 선형 회귀 모델이다 [63]. PLS는 관측된 데이터보다 변수의 개수가 더 많고 변수 사이의 다중공선성 (multicollinearity)이 강하게 나타날 때 사용되는 방법이다. PLS를 적용했을 때 X, Y 는 각각 score matrix T, U 와 loading matrix P, Q 로 분해되며, 식 (7), (8)과 같이 나타낼 수 있다.

$$X = TP^T + E = \sum_{i=1}^{n_{pc}} t_i p_i^T + E \quad (7)$$

$$Y = UQ^T + F = \sum_{i=1}^{n_{pc}} u_i q_i^T + F \quad (8)$$

이때 $X \in R^{N \times n_p}$, $Y \in R^{N \times n_q}$ 이고 N 은 샘플의 개수, n_p 와 n_q 는 각각 X, Y 를 이루는 변수의 개수, n_{pc} 는 주성분의 개수, 그리고 E, F 는 각각 X, Y 에 대한 오차항을 의미한다.

Multiway PLS (MPLS)는 [10]에서 제안한 방법론으로, 기존의 PLS를 확장하여 3차원 구조를 가지는 배치 공정의 공정 데이터를 다루기 위해 고안되었다. MPLS는 각 배치의 운전 시간이 모두 동일하다고 가정한다. MPLS를 적용하기 위해서는 먼저 batch-wise unfolding 및 정규화를 통해 공정 데이터를 전 처리하여 2차원 형태의 데이터를 구성해야 한다. 이후에 2차원 형태의 공정 데이터 $X(I \times JK)$ 와 품질 변수 데이터 Y 에 대해서 PLS를 적용하여 품질 예측 모델을 구성할 수 있다.

제 4 절 Phase Partition

제 1 항 Sub-PCA 알고리즘

sub-PCA는 [30]의 연구에서 제시된 알고리즘으로, 3차원 형태의 배치 공정 데이터에 대해서 공정 변수 사이의 상관관계를 분석하여 phase를 분리하는 방법이다. 각 배치의 운전 시간이 동일할 때, 각 시점 k 마다 주어지는 2차원의 배치 데이터 $X^k(I \times J)$ 에 대해서 주성분 분석 (principal component analysis, PCA)을 사용하여 공정 변수 간 상관관계를 표현하는 loading matrix를 구한다. 이후에 각 주성분의 고유값 (eigenvalue)으로 가중치를 계산한 변형된 형태의 loading matrix를 구하고, 해당 행렬에 대해서 K-means clustering 알고리즘을 사용하여 상관관계가 비슷한 시점끼리 군집화 하여 phase를 분할한다. 하지만 이 방법은 각 배치의 운전 시간과 각 phase의 지속 시간이 동일한 경우에만 사용할 수 있기 때문에 각 배치 별로 운전 시간이 동일하지 않은 일반적인 배치 공정에 대해서 적용할 수 없다는 문제점을 가지고 있다.

이후 [33]에서 기존의 sub-PCA 기법을 변형하여 운전 시간이 균일하지 않은 배치 공정에 대해서 적용할 수 있는 알고리즘을 제안했다. 가장 짧은 배치 공정의 운전 시간이 K_s 일 때, K_s 개의 2차원 행렬 $\tilde{X}^k(I \times J)$ 를 구할 수 있다. 이때 기존 sub-PCA 알고리즘과 마찬가지로 각 시점마다 PCA를 적용하면 식 (9)와 같이 각 시점에 대한 loading matrix, \tilde{P}^k 를 계산할 수 있다.

$$\tilde{X}^k = \tilde{T}^k(\tilde{P}^k)^T \quad (k = 1, 2, \dots, K_s) \quad (9)$$

또한 식 (10)과 같이 주성분의 고유값을 이용하여 가중치를 계산한 loading matrix, \check{P}^k 를 구할 수 있다.

$$\check{P}^k = [p_1^k \cdot g_1^k, p_2^k \cdot g_2^k, \dots, p_j^k \cdot g_j^k], g_j^k = \frac{\lambda_j^k}{\sum_{j=1}^J \lambda_j^k} \quad (10)$$

이때 λ_j^k 는 k 시점의 time slice가 보이는 상관관계 행렬의 j 번째 고유값이고, p_j^k 는 가중치를 고려하지 않은 loading matrix \check{P}^k 의 열 벡터를 의미한다. K_s 개의 \check{P}^k 에 대해서 K-means clustering 알고리즘을 적용하여 사용자가 설정한 phase 개수만큼의 클러스터로 군집화 한다. 첫번째 phase의 데이터를 분류하기 위해서 첫번째 클러스터에 속한 \check{P}^k 의 평균값을 식 (11)과 같이 계산한다.

$$P^* = \frac{1}{N} \sum_{k=1}^N \check{P}^k \quad (11)$$

이때 P^* 는 cumulative explained variance가 0.9를 넘기는 principal component subspace와 residual subspace로 나눌 수 있다. Principal component subspace를 \bar{P}^* 라고 할 때, 공정 데이터 x 에 대해서 squared prediction error (SPE)는 식 (12), (13)과 같이 계산할 수 있다.

$$e = x - x\bar{P}^*(\bar{P}^*)^T \quad (12)$$

$$SPE = e^T e \quad (13)$$

데이터 x 가 현재 탐색 중인 phase에 속한다면 SPE가 낮게 나타나지만, 데이터가 다음 phase에 속한다면 PCA를 통해 변수 사이의 관계를 설명할 수 없기 때문에 높은 SPE가 나타난다. 따라서 사용자가 지정한 기준치인 SPE^* 에 대해서 더 큰 오차가 나타나는 시점까지 phase를 분할할 수 있다. 분할된 phase의 데이터를 제거한 이후에 모든 phase를 분리할 때까지 동일한 과정을 반복하여 수행함으로써 공정 데이터의 phase를 분할한다.

제 2 항 Warped K-Means Clustering 알고리즘

Warped K-means clustering (WKM) 알고리즘은 데이터의 순서를 고려하지 않는 기존의 K-means clustering 알고리즘에 시계열성과 관련된 hard constraint 조건을 추가하여 시계열 데이터의 군집화를 할 수 있도록 고안된 알고리즘이다 [64]. WKM 알고리즘은 n 개의 d 차원 시계열 데이터 $\mathbf{x}_i \in R^d (i = 1, \dots, n)$ 로 구성된 데이터셋 $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 에 대해서 기존의 K-means clustering 알고리즘의 목적함수와 동일한 sum of quadratic error (SQE)를 최소화하도록 하는 C 개의 순차적인 클러스터로 군집화 한다. 이때 목적함수 J 는 식 (14), (15)와 같다.

$$J = \sum_{c=1}^C J_c = \sum_{c=1}^C \sum_{i=b_c}^{b_{c+1}-1} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2, \quad (14)$$

$$\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{i=b_c}^{b_{c+1}-1} \mathbf{x}_i \quad (15)$$

$\boldsymbol{\mu}_c$ 는 c 번째 클러스터의 평균값을 의미하고, n_c 와 b_c 는 각각 c 번째 클러스터의 원소의 개수와 클러스터 시작 시점의 인덱스를 의미한다. WKM 알고리즘은 목적함수를 최소화하는 최적의 데이터 군집화를 위해서 각각의 데이터에 대해서 클러스터를 옮기는 작업을 수행한다. 이때 데이터 \mathbf{x} 가 j 번째 클러스터에서 i 번째 클러스터로 이동할 때 목적함수의 변화량 $\Delta J(\mathbf{x}, j, i)$ 와 새로운 목적함수 \hat{J} 은 식 (16), (17)과 같이 정의할 수 있다.

$$\Delta J(\mathbf{x}, j, i) = \frac{n_i}{n_i + 1} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 - \frac{n_j}{n_j - 1} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2 \quad (16)$$

$$\hat{J} = J + \Delta J(\mathbf{x}, j, i) \quad (17)$$

이때 i 번째 클러스터와 j 번째 클러스터의 새로운 평균값 $\hat{\boldsymbol{\mu}}_i$ 와 $\hat{\boldsymbol{\mu}}_j$ 은 식

(18), (19)와 같이 계산할 수 있다.

$$\hat{\mu}_j = \mu_j - \frac{x - \mu_j}{n_j - 1} \quad (18)$$

$$\hat{\mu}_i = \mu_i + \frac{x - \mu_i}{n_i + 1} \quad (19)$$

이때 $\Delta J(x, j, i)$ 가 음수라면 데이터 x 는 j 번째 클러스터에서 i 번째 클러스터로 이동하고, 양수라면 이동하지 않는다.

WKM 알고리즘에서는 시계열 데이터의 순차성을 고려하여 군집화를 진행하기 위해서 각 데이터를 새로운 클러스터로 옮길 때 hard constraint를 적용한다. i 번째 클러스터의 초기 절반 데이터는 $i - 1$ 번째 클러스터로의 움직임만 허용되고, i 번째 클러스터의 후기 절반 데이터는 $i + 1$ 번째 클러스터로의 움직임만 허용된다. 이와 같은 hard constraint를 적용하여 각 클러스터의 경계선에 있는 데이터만 움직임으로써 전체 시계열 데이터가 가지는 순차성을 잃지 않고 군집화를 진행할 수 있다. 더 이상 데이터의 움직임이 없을 때까지 반복적으로 알고리즘을 수행한다. 이때 클러스터의 개수 C 는 사용자의 임의로 설정해주어야 하며, 클러스터의 초기화를 위해서는 trace segmentation (TS) 기법을 적용한다 [65]. 보다 자세한 WKM 알고리즘은 [64]에서 확인할 수 있다.

일반적인 배치 공정은 전체 운전 시간과 각 phase의 지속 시간이 배치마다 다르게 나타나기 때문에, 운전 시간이 동일한 배치 공정에 대해서 적용되는 phase 분할 방법을 사용하는 것이 불가능하다. 이때 WKM 알고리즘을 적용한다면 운전 시간과 phase의 기간에 상관없이 각각의 배치에 대해서 따로 phase를 분리할 수 있다 [37]. 하지만 WKM 알고리즘을 적용하기 위해서는 클러스터의 개수를 먼저 설정해주어야 하는데, phase를 분리하기 이전에 미리 phase의 개수를

알기 어렵다는 문제가 있다. WKM 알고리즘을 적용할 때 임의로 클러스터의 개수를 늘리면 SQE는 점차 줄어들지만 계산의 복잡도가 증가하고 phase를 구분하는 의미가 없어진다. 반대로 클러스터의 개수를 줄이면 계산의 복잡도는 줄어들지만 각 phase가 가지는 공정의 특성을 충분히 반영하지 못한다는 문제가 발생한다. 따라서 [37]에서 phase의 개수에 따른 trade-off 관계를 표현하기 위해서 식 (20), (21), (22)와 같은 phase performance combination index (PPCI) 값을 제안하였다.

$$PPCI_p = \gamma \tilde{J}_p + (1 - \gamma) \tilde{p} \quad (20)$$

$$J_p = \log\left(\sum_{i=1}^I J_p^i\right), p = 1, \dots, p^* \quad (21)$$

$$\tilde{J}_p = \frac{J_p - \text{mean}(J_p)}{\text{std}(J_p)}, \tilde{p} = \frac{p - \text{mean}(p)}{\text{std}(p)} \quad (22)$$

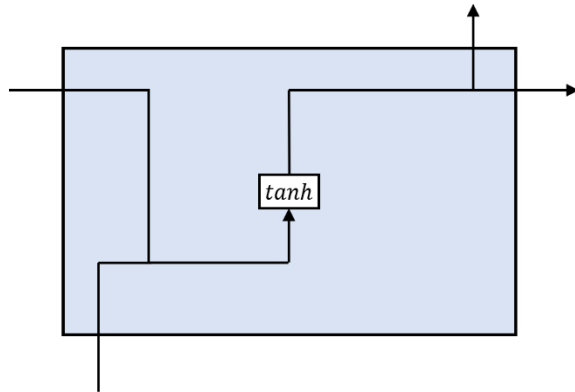
\tilde{J}_p 는 phase의 개수를 p 로 했을 때 각 배치가 가지는 SQE 값의 로그 총합을 정규화 한 값이고, \tilde{p} 는 정규화 된 phase의 개수 p 를 의미한다. 이때 $\gamma \in (0,1)$ 은 phase 분할 성능과 복잡도 사이의 trade-off를 고려한 계수이다. 따라서 PPCI 값을 최소화하여 분할 성능과 복잡도의 trade-off에 대해서 최적의 phase 개수를 선택할 수 있다.

제 5 절 Recurrent Neural Network

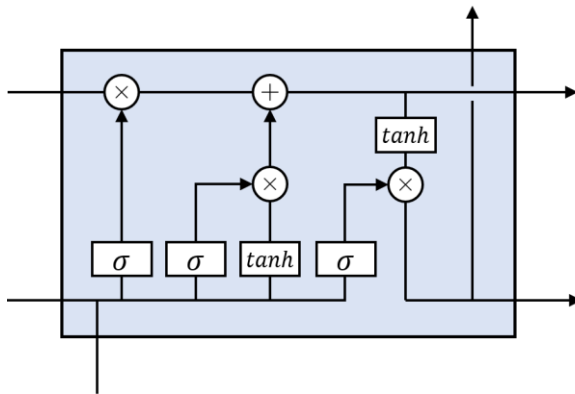
제 1 항 Vanilla RNN, LSTM, GRU

Recurrent neural network (RNN)는 순환 구조를 가지는 인공 신경망의 일종으로, 음성, 문자와 같은 시계열 데이터를 처리하는 데에 적합한 모델로 알려져 있으며, RNN의 구조는 그림 2와 같다. RNN은 시퀀스의 길이에 상관없이 입력과 출력을 처리할 수 있기 때문에 다양하고 유연한 구조를 형성할 수 있다는 장점을 가지고 있다 [66, 67]. 하지만 가장 기본적인 형태의 RNN인 Vanilla RNN의 경우 시퀀스의 길이가 늘어나면 역전파를 하는 과정 중에 기울기 (gradient)가 점차 줄어들어 학습 능력이 저하되는 기울기 소실 (gradient vanishing) 문제를 가진다 [68]. 이는 시퀀스 길이가 늘어남에 따라 시퀀스 초반의 데이터가 시퀀스 후반의 데이터에 주는 영향을 고려하지 못하는 장기 의존성 문제 (long-term dependency)를 일으킨다. 이와 같은 Vanilla RNN의 단점을 보완하기 위해서 LSTM이 제안되었다.

(a)



(b)



(c)

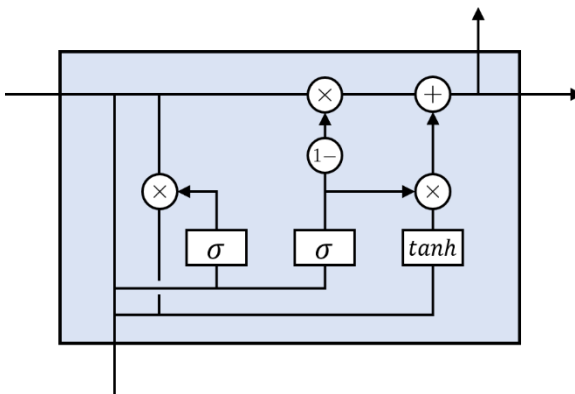


그림 2. (a) Vanilla RNN, (b) LSTM, (c) GRU의 구조.

Long short-term memory (LSTM)는 Vanilla RNN과 동일하게 순환 구조를 가지는 인공 신경망으로, 입력된 시퀀스 중에서 어떤 정보를 기억하고 어떤 정보를 잊을 것인지 선택해서 중요한 정보를 오래 기억하도록 하여 장기 의존성 문제를 해결한다 [69]. LSTM은 Vanilla RNN과는 다르게 hidden state 뿐만 아니라 cell state를 활용하는데, cell state c_t 는 과거부터 t 시점까지의 모든 정보를 저장하고 있다고 가정한다. 따라서 이전 시점의 정보인 h_{t-1} , c_{t-1} 과 t 시점에 제공된 시퀀스 정보인 x_t 을 기반으로 하여 LSTM 내부에서 일어나는 연산을 통해서 현재 시점의 cell state인 c_t 을 계산한다. 이때 LSTM의 내부에서는 망각 게이트 (forget gate), 입력 게이트 (input gate), 출력 게이트 (output gate)로 구성된 3개의 게이트를 활용하여 장기 의존성 문제를 다룬다. 망각 게이트는 과거 정보를 얼마나 기억하고 잊을지를 결정하는 게이트이며, 현재 시점의 입력 벡터 x_t 와 이전 시점의 hidden state 벡터 h_{t-1} 에 대해서 sigmoid 함수를 적용하여 식 (23)과 같이 표현할 수 있다.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (23)$$

입력 게이트는 현재 정보를 얼마나 반영할지를 결정하는 게이트이며, 망각 게이트와 마찬가지로 현재 시점의 입력 벡터 x_t 와 이전 시점의 hidden state 벡터 h_{t-1} 에 대해서 sigmoid 함수를 적용하여 식 (24)와 같이 표현할 수 있다.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (24)$$

이때 cell state에 새롭게 추가할 정보를 담은 벡터는 \tanh 함수를 사용하여 식 (25)와 같이 계산한다.

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \quad (25)$$

망각 게이트와 입력 게이트를 활용하여 이전 시점의 cell state 중에서

잇을 정보를 제거하고 현재 시점의 입력 중에서 기억할 정보를 추가하여 식 (26)과 같이 cell state를 업데이트한다.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (26)$$

출력 게이트는 업데이트 된 cell state에서 다음 LSTM 계층으로 전달하고 출력할 정보를 결정하는 게이트이며, 현재 시점의 입력 벡터 x_t 와 이전 시점의 hidden state 벡터 h_{t-1} 에 대해서 sigmoid 함수를 적용하여 식 (27)과 같이 표현할 수 있다.

$$o_t = \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o) \quad (27)$$

최종적으로 \tanh 함수를 사용하여 식 (28)과 같이 현재 시점의 hidden state를 계산한다.

$$h_t = o_t \odot \tanh(c_t) \quad (28)$$

이와 같은 LSTM의 구조는 역전과 과정에서 기울기의 흐름이 덧셈 노드를 통과하고, 행렬 곱 연산이 아닌 곱셈 연산의 역전파를 사용하기 때문에 기울기 소실이 발생할 가능성이 낮아진다. 하지만 LSTM은 학습할 파라미터가 많기 때문에 계산량이 많다는 단점을 가지고 있다. 이를 보완하기 위해서 GRU가 제안되었다.

Gated recurrent unit (GRU)는 LSTM의 구조를 간소화하여 개선한 RNN 모델로, LSTM에서 사용된 hidden state와 cell state를 하나의 hidden state로 표현한다 [70]. GRU는 리셋 게이트와 업데이트 게이트의 두 게이트로 구성된 구조로 형태를 변경하여 기존의 LSTM과 비슷한 성능을 유지하면서 적은 파라미터를 사용하도록 한다. 리셋 게이트는 이전 시점의 hidden state의 정보를 얼마나 제거할지 결정하며, 식 (29)와 같이 LSTM의 게이트와 유사한 방법으로 계산한다.

$$r_t = \sigma(W_{x_r}x_t + W_{h_r}h_{t-1} + b_r) \quad (29)$$

GRU의 업데이트 게이트는 LSTM의 입력 게이트와 망각 게이트의 역할을 통합하여 이전 시점의 hidden state와 현재 시점의 입력 벡터에서 필요한 정보만을 선택하도록 한다. 업데이트 게이트는 식 (30)과 같이 계산한다.

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (30)$$

이후에 리셋 게이트를 활용하여 이전 시점의 hidden state에서 얼마나 많은 정보를 유지하고 무시할지 결정한다. Hidden state의 업데이트에 사용할 임시적인 hidden state \tilde{h}_t 는 식 (31)과 같이 계산한다.

$$\tilde{h}_t = \tanh(W_{x\tilde{h}}x_t + W_{h\tilde{h}}(r_t \odot h_{t-1}) + b_g) \quad (31)$$

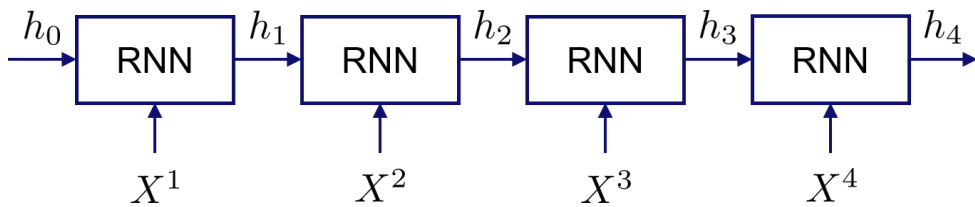
마지막으로 업데이트 게이트를 가중치로 사용하여 h_{t-1} 과 \tilde{h}_t 의 가중 평균을 계산하여 식 (32)와 같이 현재 시점의 hidden state를 결정한다.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (32)$$

제 2 항 Multi RNN 모델

배치 공정에서 생산되는 제품의 품질은 공정의 운전 내용에 따라 변화한다. 운전 시간 동안 누적된 고차원의 정보로부터 제품의 품질을 예측하는 모델을 만들기 위해서는 고차원의 데이터에 대해서 차원 축소를 통해 제품 품질을 추정하는 데에 필요한 낮은 차원의 feature를 추출해야 한다. 또한 비정상상태에서 운전되는 배치 공정이 보이는 비선형성과 공정 데이터가 가지는 시계열성을 고려하기 위해서 RNN을 활용하는 방안을 고려할 수 있다. 본 연구에서는 RNN을 사용하여 각 배치 공정의 hidden state를 추출해내어 품질 변수를 추정하는 모델을 구성했다.

배치 공정이 가지는 multi-phase 특성을 고려하여 각 phase의 데이터에 대해서 각각 RNN 셀을 학습시키는 구조로 제품 품질 변수를 추정하는 모델을 구성했다. [58]의 연구에서 제시된 RNN 모델의 구조를 활용했으며, 해당 RNN 모델의 구조는 그림 3와 같다. 해당 모델은 이전 phase의 RNN 셀에서 계산되어 출력된 hidden state를 다음 셀에 입력되는 hidden state로 설정하였다. 이를 통해 여러 개의 RNN 셀을 유기적으로 연결하였으며, 각 phase의 공정 정보가 연속적으로 다음 phase의 RNN 셀에 전달되는 형태를 구성했다. 각각의 RNN 셀은 입력 값으로 주어지는 각 phase의 공정 데이터를 활용하여 각 phase의 공정 정보를 담고 있는 낮은 차원의 hidden state를 계산한다. 각 RNN 셀에서 계산된 hidden state를 순차적으로 연결한 벡터를 입력 값으로 가지는 affine function을 통해 제품의 품질을 예측하도록 하였다. 최종적으로 모델로부터 예측된 제품 품질 변수와 실제 제품 품질 변수의 차이에 대해서 평균 제곱 오차 (mean squared error, MSE) 값을 손실 함수로 설정하여 RNN 모델의 학습을 진행했다.



$$[h_1, h_2, h_3, h_4] \rightarrow \hat{y}$$

그림 3. Multi RNN의 구조.

제 3 장 페니실린 생산 공정

페니실린 (*Penicillium chrysogenum*)을 생산하는 fed-batch fermentation bioreactor는 공정 모니터링 및 제어를 위한 벤치마크 공정으로써 많은 연구에서 사용되고 있다 [71~73]. 특히 [74,75]의 연구에서는 산업용 생물반응기의 벤치마크 모델이 연구되었고, 100,000L 크기의 페니실린 생산 공정에 대해서 실제 반응기의 공정 운전 정보로 검증된 반응기 모델이 개발되었다. 해당 반응기의 개략적인 모식도는 그림 4와 같다. 해당 배치 공정의 시뮬레이터 IndPenSim과 시뮬레이터로 생성된 공정 데이터는 오픈 소스로 공개되어 있으며, www.industrialpenicillinsimulation.com에서 찾아볼 수 있다. 본 연구에서는 IndPenSim으로부터 생성된 100개의 배치에 대한 공정 데이터를 활용했다. 해당 데이터셋은 공정 레시피에 따라서 운전된 30개의 배치, 오퍼레이터의 제어에 따른 30개의 배치, advanced process control (APC)에 따라서 제어된 30개의 배치, 그리고 공정 이상을 포함하고 있는 마지막 10개의 배치에 관한 공정 데이터로 구성되어 있다. 전체 100개의 배치 데이터 중에서 80개의 배치 데이터는 모델의 학습에 사용되었고, 20개의 배치 데이터는 모델의 테스트를 위해 사용되었다. 학습 데이터와 테스트 데이터는 무작위로 선별되었다. 해당 데이터셋은 각 배치 마다 공정 운전 시간이 동일하지 않으며, 공정 운전 시간의 분포는 그림 5의 히스토그램과 같다. 공정 운전 시간은 평균 228시간이며, 최소 167시간에서 최대 290시간까지 분포되어 있다. 각 공정 변수는 0.2시간 단위로 측정되었다. 공정 분석에 사용된 공정 변수는 총 19개이고, 18개의 공정 변수와 1개의 품질 변수로 구성되어 있다. 전체 공정 변수는 표 1과 같다. 100개의 배치 공정에 대해서 공정 변수가 오프라인에서 모두 측정되어 활용할 수

있음을 가정했다. 또한 공정의 생산품인 페니실린의 농도를 제품 품질 변수로 설정했고, 페니실린의 농도는 배치 공정이 마무리된 이후에 측정 가능하다고 가정했다.

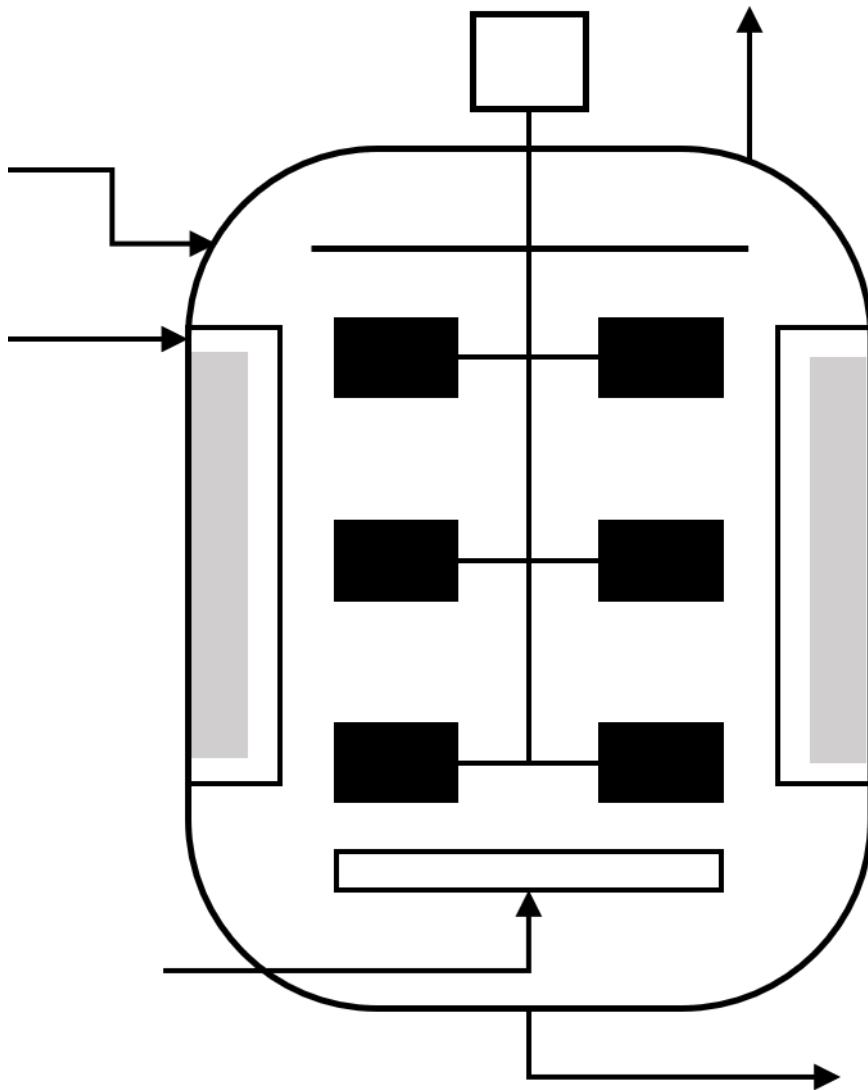


그림 4. 페니실린 생산 bioreactor 모식도.

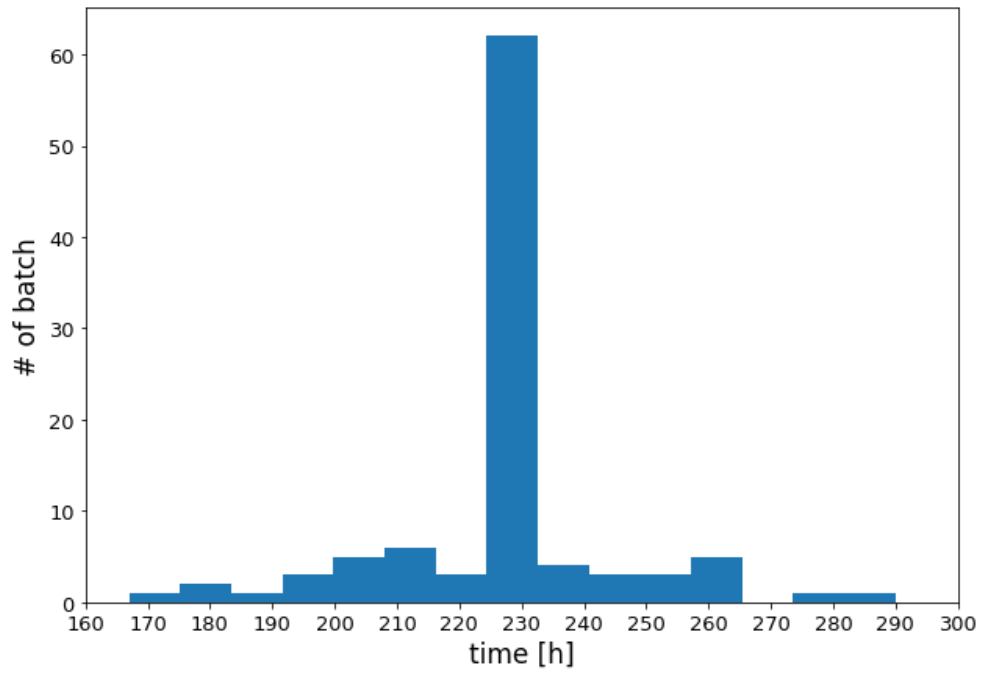


그림 5. 공정 운전 시간 분포.

표 1. 페니실린 생산 bioreactor의 공정 변수.

번호	변수	단위	분류
1	폭기 속도	L/h	공정 변수
2	기질 유량	L/h	공정 변수
3	산 유량	L/h	공정 변수
4	염기 유량	L/h	공정 변수
5	냉각 용수 유량	L/h	공정 변수
6	가열 용수 유량	L/h	공정 변수
7	반응기 회석 용수 유량	L/h	공정 변수
8	반응기 내부 압력	bar	공정 변수
9	배출 유량	L/h	공정 변수
10	PAA 유량	L/h	공정 변수
11	오일 유량	L/h	공정 변수
12	기질 농도	g/L	공정 변수
13	용존 산소량	mg/L	공정 변수
14	반응물 부피	L	공정 변수
15	pH	pH	공정 변수
16	반응기 내부 온도	K	공정 변수
17	배출 가스의 이산화탄소 비율	%	공정 변수
18	배출 가스의 산소 비율	%	공정 변수
19	페니실린 농도	g/L	품질 변수

제 4 장 결과 및 분석

품질 예측 모델의 예측 성능 비교를 위한 metric으로는 root mean squared error (RMSE), mean absolute percentage error (MAPE), 그리고 coefficient of determination (R^2 score)을 사용했다. RMSE와 MAPE는 각각 에러의 제곱과 절댓값을 활용한 지표이며 회귀 모델의 오차를 표현하고, R^2 score는 독립변수가 종속변수를 얼마나 잘 설명하는지 나타내는 지표로써 회귀 모델의 설명력을 나타낸다. RMSE, MAPE, 그리고 R^2 score는 각각 식 (33), (34), (35)와 같이 정의된다.

$$RMSE = \sqrt{\frac{1}{I} \sum_{i=1}^I (y_i - \hat{y}_i)^2} \quad (33)$$

$$MAPE = 100 \times \frac{1}{I} \sum_{i=1}^I \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (34)$$

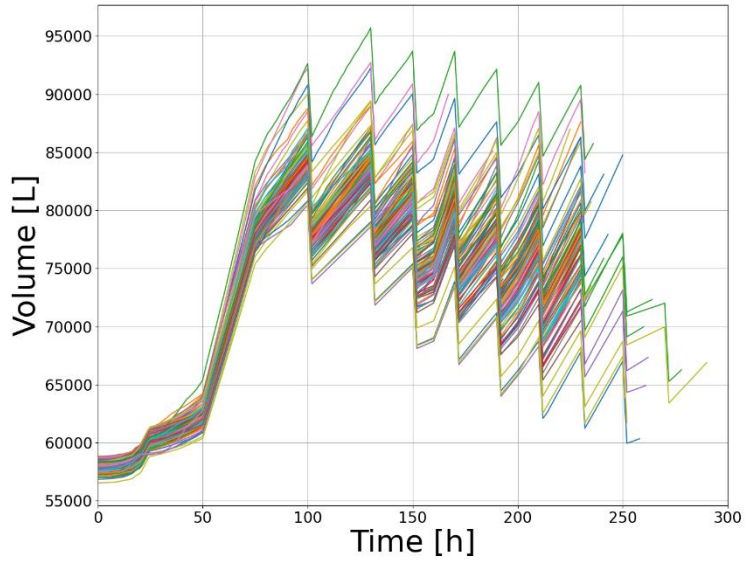
$$R^2 = 1 - \frac{\sum_{i=1}^I (y_i - \hat{y}_i)^2}{\sum_{i=1}^I (y_i - \bar{y})^2} \quad (35)$$

이때 y_i 는 i 번째 배치의 실제 품질 변수, \hat{y}_i 는 모델로부터 추정된 i 번째 배치의 품질 변수, 그리고 \bar{y} 는 실제 품질 변수의 평균값을 의미한다. RMSE와 MAPE는 0에 가까울수록 모델의 예측 정확도가 높다는 것을 나타내며, R^2 score는 1에 가까울수록 모델의 설명력이 좋다는 것을 나타낸다.

제 1 절 DTW-MPLS 모델

운전 시간이 일정하지 않은 배치 공정의 공정 데이터에 대해서 MPLS 기법을 적용한 품질 예측 모델을 구성하기 위해서 DTW 기법을 적용하여 공정 데이터의 시간 축을 정렬했다. 공정 운전 시간의 평균값인 228시간의 운전 시간을 가지는 배치를 시간 축의 기준이 될 운전 경로로써 사용했다. 배치 공정의 정렬은 [22]에서 제안된 방법과 같이 symmetric DTW와 asymmetric DTW의 두 단계를 거쳐 진행되었다. 먼저 symmetric DTW 알고리즘을 활용하여 모든 배치 공정의 반응 경로를 기준이 되는 운전 경로와 비교하여 DTW를 적용함으로써 시간 축을 정렬했다. 이때 symmetric DTW를 적용하여 서로 다른 길이로 정렬된 공정 데이터에 대해서 asymmetric DTW 알고리즘을 적용하여 모든 배치 데이터의 길이가 기준이 되는 운전 경로의 길이와 동일하도록 정렬해주었다. 정렬되기 전의 공정 데이터와 정렬된 이후의 공정 데이터는 그림 6과 같이 표현할 수 있다.

(a)



(b)

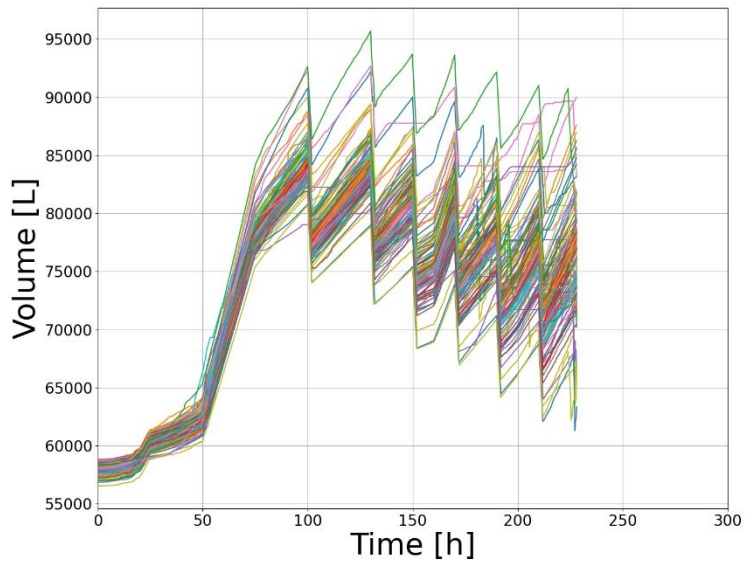


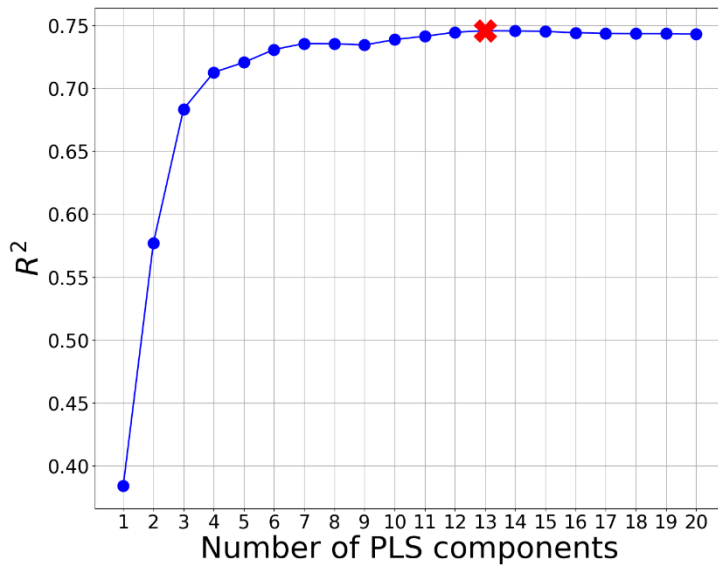
그림 6. DTW 기법을 통한 정렬 (a)전, (b)후의 반응기 내부 부피 변화.

운전 시간이 동일하도록 정렬된 공정 데이터에 대해서 [10]에서 제안한 MPLS 기법을 적용하여 제품 품질 변수를 추정하는 모델을 구성했다. 먼저 MPLS 모델의 주성분 개수를 결정하기 위해서 주성분의 개수를 증가시키며 validation을 진행했다. 주성분 개수의 변화에 따라서 그림 7과 같이 R^2 score와 MSE 값이 변화하는 것을 확인할 수 있었다. 초기에 주성분의 개수가 적을 때는 주성분의 개수가 늘어남에 따라서 오차가 빠르게 줄어들지만, 주성분의 개수가 일정 값을 넘어서면 오차가 줄어들지 않고 점차 증가하는 것을 확인할 수 있다. 따라서 오차가 최소가 되며 이후로 오차가 줄어들지 않는 주성분 개수인 13개의 주성분을 활용하여 MPLS 모델을 구성했다. 테스트 데이터에 대해서 MPLS 모델을 활용하여 품질 변수를 추정한 결과는 그림 8과 같다. 각 모델에 대해서 실제 페니실린 농도와 예측된 페니실린 농도를 2차원으로 표현한 parity plot에서 표기된 점들이 $y = x$ 선에 가깝게 분포할수록 예측 모델의 정확하다고 할 수 있다. DTW-MPLS 모델의 품질 예측 과정에서 발생한 오차는 RMSE가 3.5419, MAPE가 13.3698%였고, R^2 score는 0.7257이었다.

DTW 및 MPLS 기법을 활용한 품질 변수 추정은 좋은 성능을 보였지만 몇 가지 한계점을 가지고 있다. 공정 데이터에 대해서 DTW를 사용하여 시간 축을 정렬하면 기준으로 선정한 배치에 맞추어 공정 데이터를 늘리거나 줄이면서 공정 정보의 왜곡이 일어나고 이로 인해서 공정 정보의 손실이 있을 수 있다. 또한 DTW를 사용하기 위해서는 배치 공정이 종료될 때까지의 모든 공정 데이터를 측정해야 하므로 공정이 진행되는 동안 온라인으로 활용할 수 없기 때문에 확장성이 떨어진다. 한편 MPLS는 DTW와 같이 시간 축을 정렬하는 기법이 선행되지 않으면 운전 시간이 동일하지 않은 배치 공정의 공정 데이터에

사용할 수 없다는 한계가 있다. 또한 MPLS는 선형 모델이기 때문에 비정상상태에서 운전되어 강한 비선형성을 보이는 배치 공정에 활용하기에는 한계가 있다. 따라서 공정 변수의 왜곡 없이 서로 다른 길이를 가지는 배치 공정의 데이터를 입력 값으로 사용할 수 있고, 공정의 동특성이 보이는 비선형성을 다룰 수 있는 RNN을 활용한 제품 품질 예측 모델을 개발해보았다.

(a)



(b)

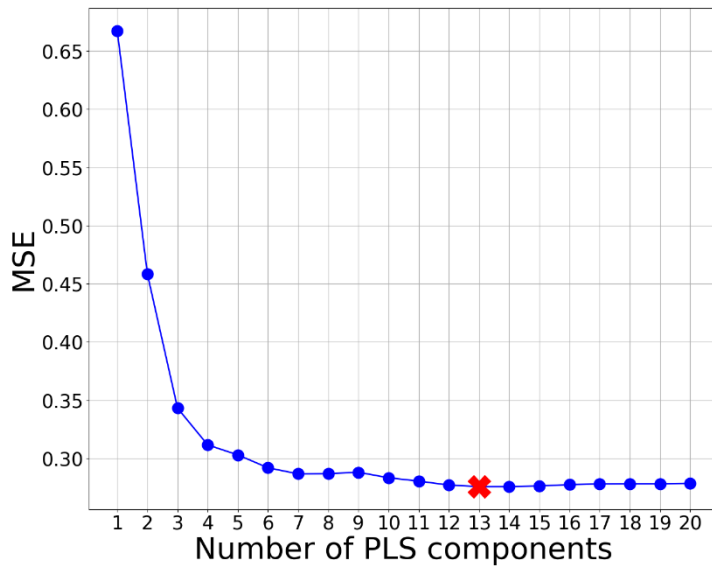
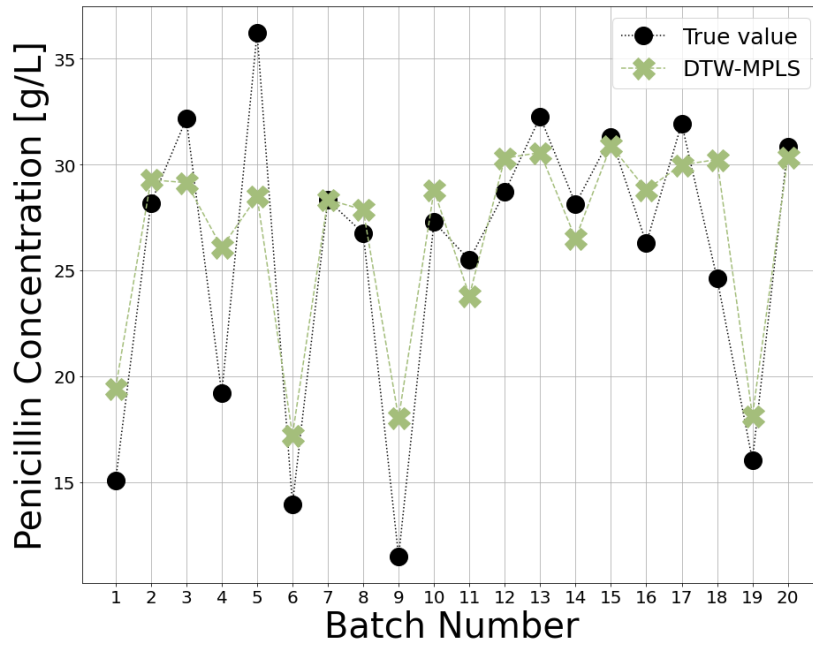


그림 7. MPLS 모델의 주성분 개수에 따른 (a) R^2 score, (b) MSE 값의 변화.

(a)



(b)

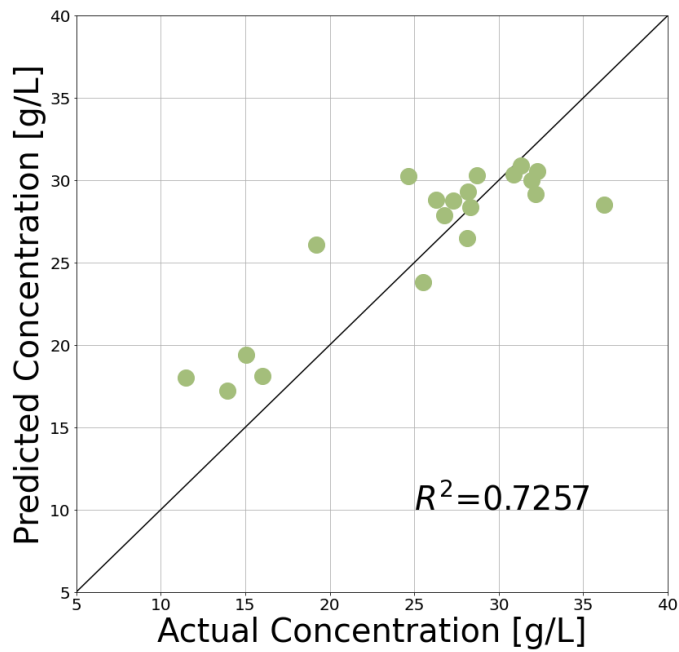


그림 8. 테스트 데이터셋에 대한 DTW-MPLS 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

제 2 절 Single RNN 모델

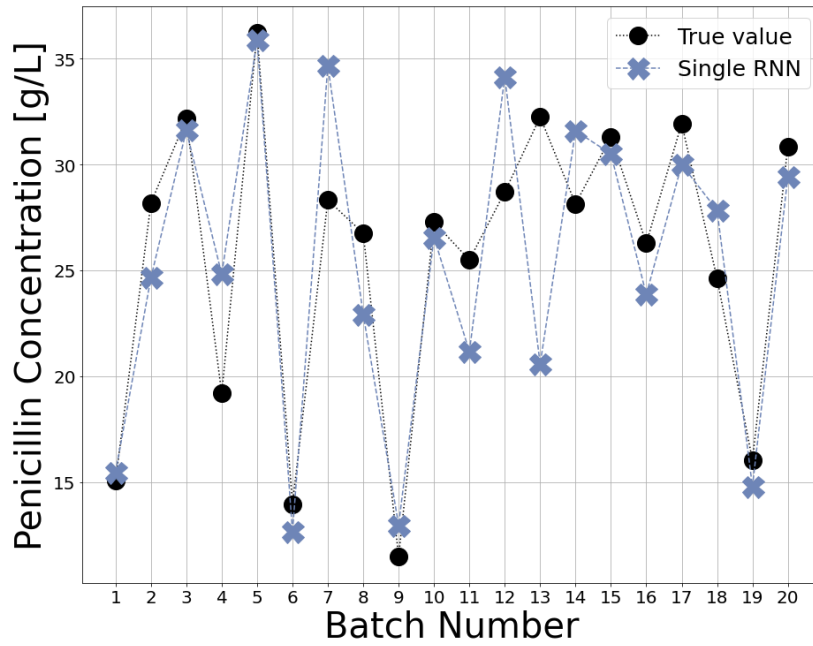
RNN은 hidden state를 다음 시점으로 전달하는 순차적인 형태로 구성되어 있기 때문에 입력 값으로 사용되는 시계열 데이터의 시퀀스 길이가 모두 동일하지 않아도 사용할 수 있다. 또한 RNN은 인공 신경망의 일종으로, 비선형성을 가지는 고차원의 입력 데이터를 저차원의 hidden state의 형태로 축소할 수 있다. 따라서 본 연구에서는 대상 공정의 공정 데이터를 활용하여 대표적인 RNN 종류인 Vanilla RNN, LSTM, 그리고 GRU로 구성된 품질 예측 모델의 학습을 진행하였다.

RNN 모델의 학습을 위해 사용한 hyperparameter의 설정은 다음과 같다. 2층으로 구성된 RNN을 사용했고, hidden state의 차원은 16차원으로 구성했다. 최적화 모델로는 Adam optimizer를 사용했으며, 모델의 학습 속도는 0.0003으로 설정했다. RNN의 기울기 소실을 방지하여 학습의 안정성을 향상시키기 위해서 gradient의 L2 norm 최대값을 3으로 설정한 gradient clipping을 사용했다. 미니 배치의 크기는 8로 설정했으며, 1000 epoch까지 학습을 진행했다.

RNN 모델의 학습이 완료된 이후에 DTW-MPLS의 결과와 동일한 테스트 데이터에 대해서 Vanilla RNN, LSTM, 그리고 GRU 기반의 품질 예측 모델의 성능을 측정했다. 각각의 결과는 그림 9, 10, 11와 같고, 이때 발생한 오차는 표 2와 같다. 각 모델의 품질 예측 과정에서 발생한 오차와 DTW-MPLS 모델에서 발생한 오차는 그림 12로부터 비교할 수 있다. Vanilla RNN, LSTM, GRU 중에서 MAPE 지표에서는 Vanilla RNN이 좋은 성능을 보였지만, 이외의 지표에서는 GRU가 가장 좋은 성능을 보였다. 하지만 예상과는 다르게 RNN을 활용한 모델이

대부분의 지표에서 DTW-MPLS보다 더 낮은 성능을 보이는 것을 확인할 수 있었다. 이때 RNN 모델의 학습을 위해 주어진 배치 공정의 데이터는 평균 1,140개의 데이터 포인트로 구성되어 있기 때문에 배치 데이터에서 입력 값으로 주어지는 시퀀스의 길이가 지나치게 길어서 RNN의 기울기 소실 현상이 나타났을 가능성이 존재한다. 또한 RNN 모델은 각 배치의 공정 데이터가 내포하고 있는 품질 변수와 관련된 특성을 하나의 hidden state 형태로 차원을 축소하는데, 하나의 RNN 셀로는 배치 공정이 가지는 여러 phase의 다양한 동특성을 반영하지 못하기 때문에 낮은 성능을 보였을 가능성이 있다. 따라서 배치 공정의 공정 데이터를 phase 별로 분할하여 각 phase의 데이터에 대해서 개별적으로 RNN의 셀을 학습시키는 형태의 모델을 적용해보았다.

(a)



(b)

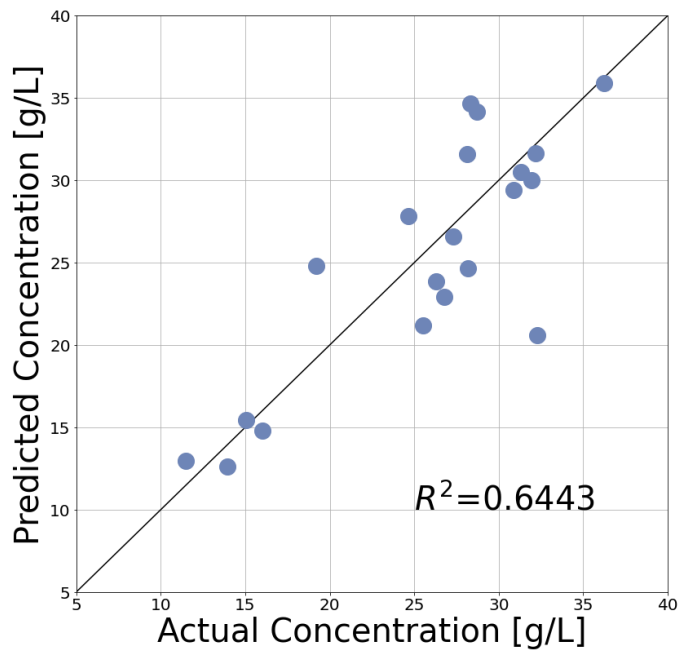
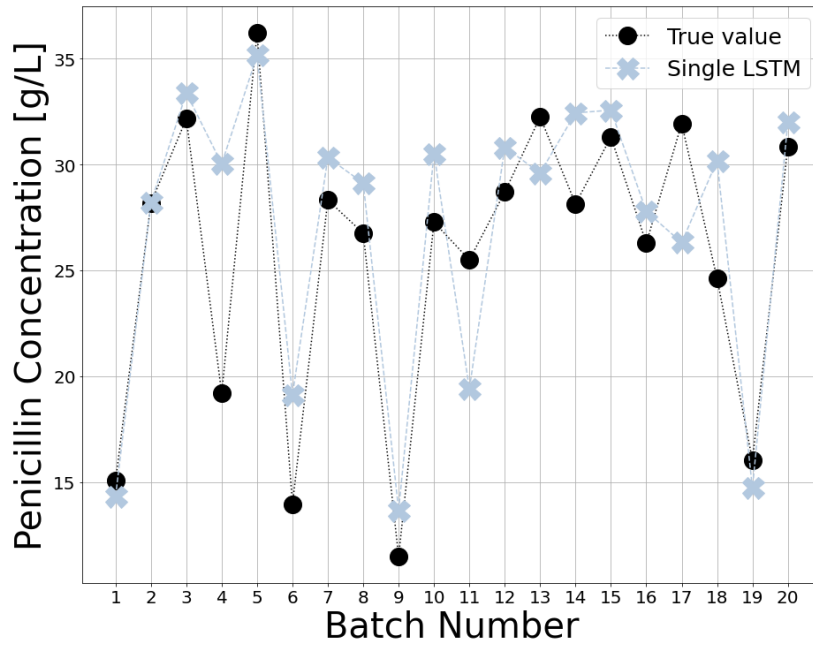


그림 9. 테스트 데이터에 대한 single RNN 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

(a)



(b)

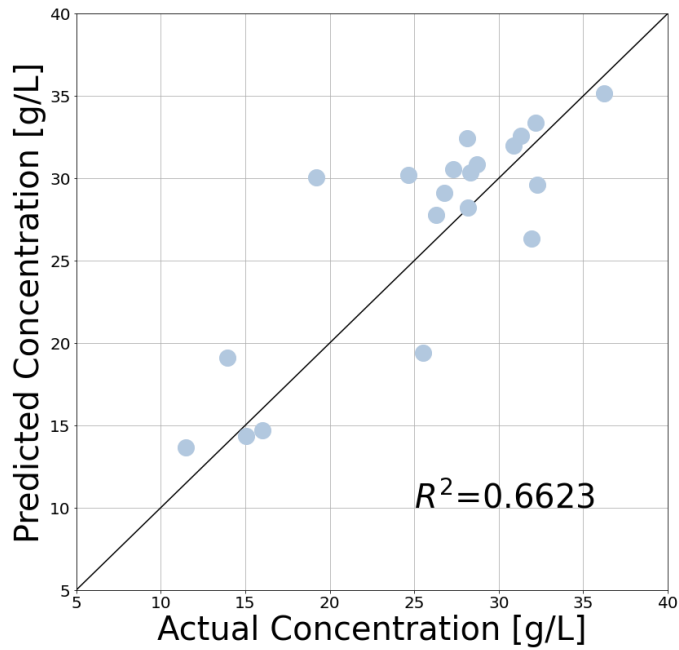
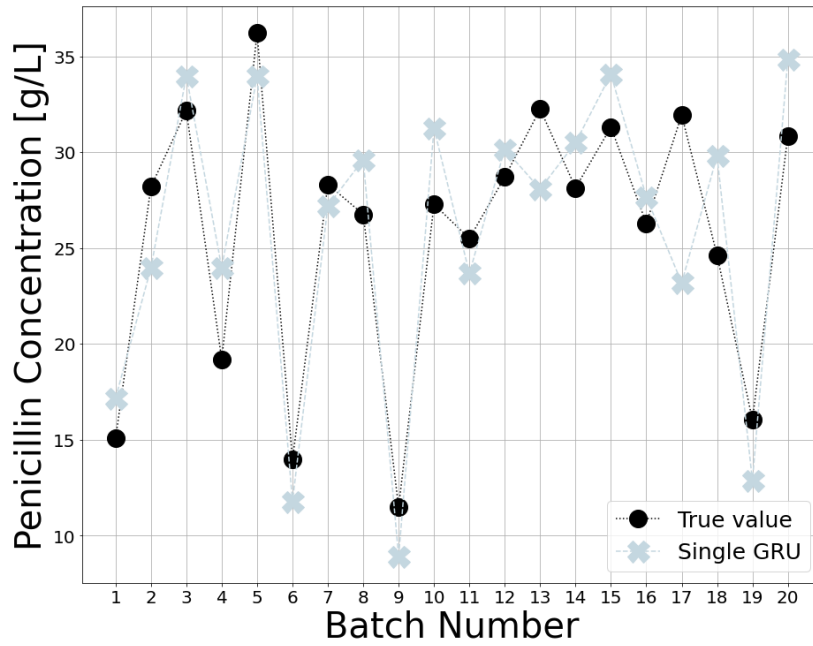


그림 10. 테스트 데이터에 대한 single LSTM 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

(a)



(b)

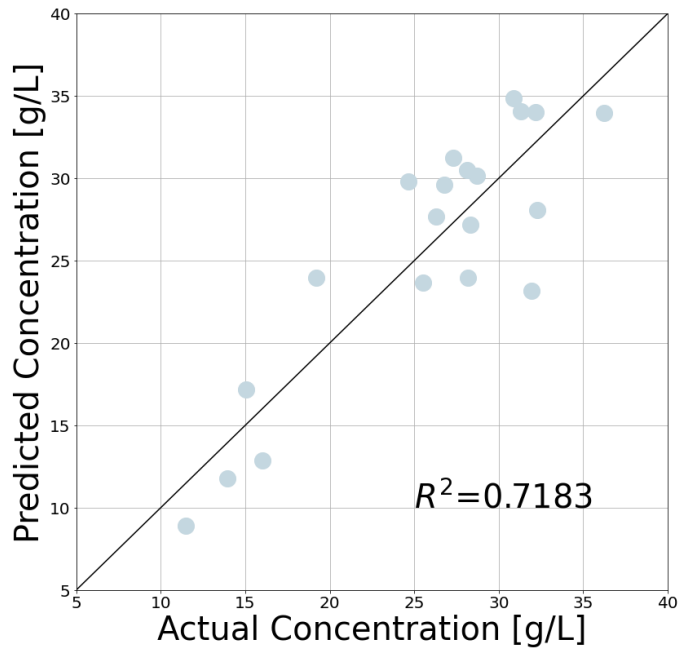
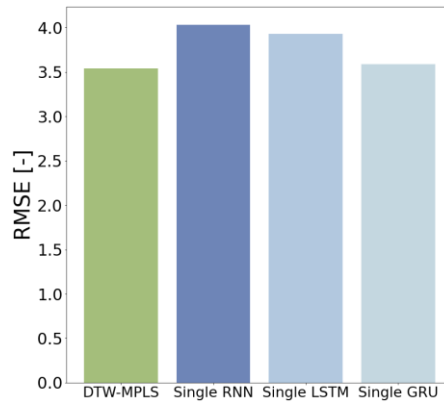


그림 11. 테스트 데이터에 대한 single GRU 모델의 (a)품질 예측 결과, (b)예측 결과에 따른 parity plot.

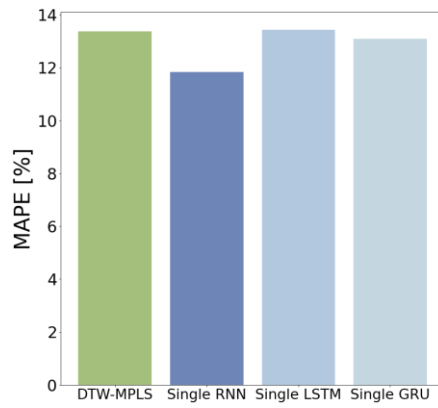
표 2. Single RNN 모델의 예측 오차.

RNN 종류	RMSE	MAPE [%]	R^2 score
Vanilla RNN	4.0334	11.8335	0.6443
LSTM	3.9303	13.4326	0.6623
GRU	3.5894	13.0920	0.7183

(a)



(b)



(c)

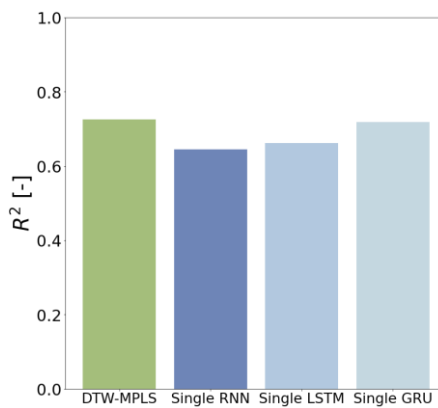


그림 12. Single RNN 모델의 예측 결과에 따른 (a)RMSE, (b) MAPE, (c) R^2 score 값의 비교.

제 3 절 Multi RNN 모델

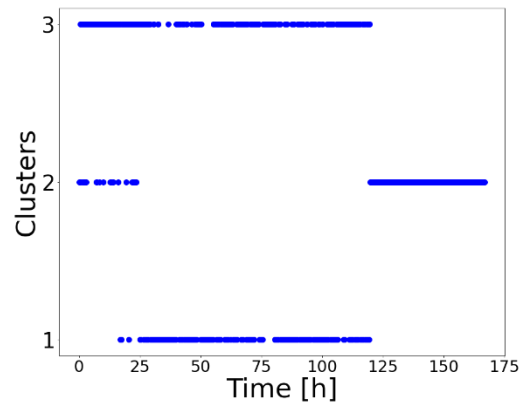
제 1 항 Phase Partition

각각의 phase 별로 RNN 셀을 학습시키기에 앞서서, 각 배치의 공정 데이터를 phase 별로 분할해주었다. [33]에서 제안한 각 배치 별 공정 데이터의 상관관계를 활용한 수정된 형태의 sub-PCA 모델을 사용하여 K-means clustering 기법을 적용하면 그림 13과 같은 결과를 얻을 수 있다. K-means clustering 알고리즘은 시계열 데이터의 순차성을 고려하지 않기 때문에 클러스터의 개수와는 상관없이 각 클러스터에 포함된 데이터가 시간과 무관하게 군집화 되어 있는 것을 확인할 수 있다. 따라서 sub-PCA 기법을 사용하기 위해서는 각각의 클러스터에 대해서 임의로 후처리를 적용하여 phase를 분리해야 한다. 또한 그림 13에 도시된 결과는 2장 4절 1항에 언급된 바와 같이 가장 짧은 운전 시간을 가지는 배치 공정을 기준으로 하여 3차원 형태의 배치 데이터에 적용한 것이다. 따라서 phase를 분리하기 위해서는 사용자가 설정한 임의의 SPE 기준 값을 사용하여 각각의 배치 데이터에서 첫번째 phase 데이터를 분리해주고, 같은 과정을 반복해야 한다. 하지만 각각의 배치 데이터가 가지는 SPE 값의 크기가 큰 차이를 보이고 phase마다 SPE 값의 변화가 크게 변화하기 때문에 설정한 기준 값에 대해서 phase가 분리되지 않는 경우가 발생했다.

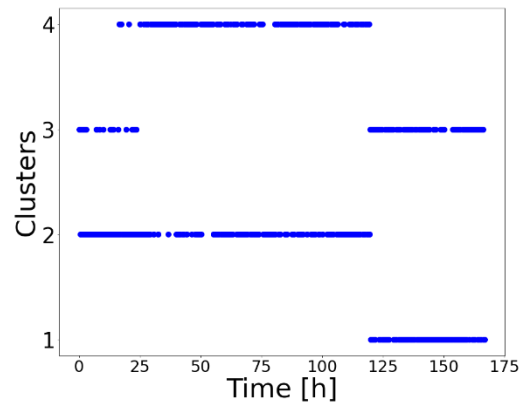
운전 시간이 균일하지 않은 배치 공정의 phase 분리를 위해 변형된 sub-PCA 기법을 사용할 때 다음과 같은 문제를 확인할 수 있다. 해당 방법론은 시계열 데이터가 가지는 순차성을 고려하지 않기 때문에 인위적인 후처리가 필요하다는 문제가 나타난다. 또한 모든 공정 변수를 활용하여 sub-PCA 기법을 적용할 경우, n^2 차원의 고차원 데이터에

대한 K-means clustering을 적용해야 하는데, K-means clustering을 통한 분류가 잘 되었는지 평가할 수 있는 기준이 없고, 고차원의 loading matrix에서 나타나는 높은 비대칭도로 인해 유클리드 거리 기반의 K-means clustering은 유의미한 결과를 내기 어렵다. 마지막으로 군집의 개수 P, 기준이 되는 SPE 값과 같이 사용자 임의로 설정하는 값에 따라서 phase 분리 결과가 크게 달라질 수 있다는 문제가 나타난다.

(a)



(b)



(c)

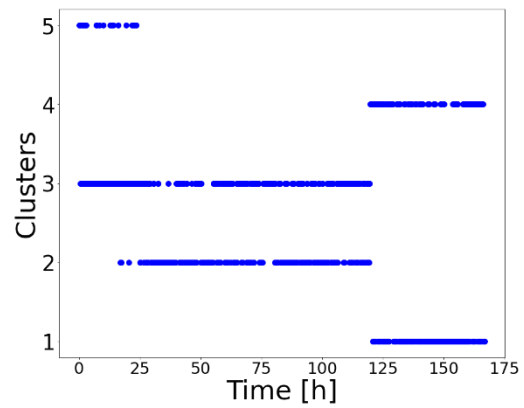


그림 13. Sub-PCA 기법을 사용하여 클러스터의 개수를 (a)3개, (b)4개, (c)5개로 설정했을 때 군집화 결과.

따라서 공정 데이터가 가지는 시계열성을 고려하여 phase를 분리하기 위해서 주어진 공정 데이터에 대해서 WKM 알고리즘을 적용하여 phase를 분리하였다. 이때 SQE 값은 군집화의 성능을 표현하며, phase의 개수에 따른 SQE 값의 변화는 그림 14와 같다. 그림으로부터 phase의 개수가 늘어날수록 SQE 값이 점점 낮아지는 경향성을 확인할 수 있다. 하지만 앞서 언급한 바와 같이 WKM 알고리즘의 군집화 성능과 계산의 복잡도는 trade-off 관계에 놓여있으므로, 높은 성능을 보이면서 낮은 복잡도를 가지는 최적의 phase 개수를 탐색해야 한다. 따라서 1에서 10까지의 phase 개수에 대해서 WKM 알고리즘을 적용하여 phase를 분할하고, 각각의 경우에 대해서 2장 4절 2항에서 정의한 PPCI 값을 계산하여 최적의 phase 개수를 탐색해보았다. $\gamma = 0.6$ 일 때 phase 개수에 따른 PPCI 값은 그림 15와 같다. 그림으로부터 PPCI를 최소화하는 최적의 phase 개수는 4개인 것을 확인할 수 있다. 따라서 WKM 알고리즘을 적용하여 배치 공정의 phase를 4개로 분리했고, 일부 배치의 경우에 phase를 분리한 결과는 그림 16과 같이 나타낼 수 있다. 첫번째 phase는 대체로 비슷한 시점에 종료되지만, 이후 3개의 phase의 시점과 기간은 배치별로 크게 차이를 보이는 것을 확인할 수 있다.

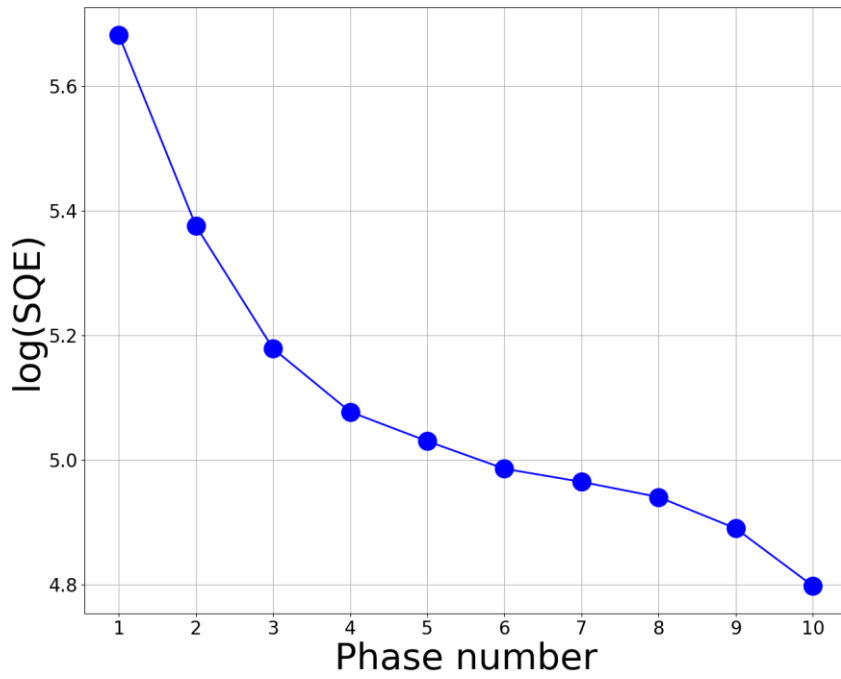


그림 14. 분할하는 phase의 개수에 따른 $\log(SQE)$ 값의 변화.

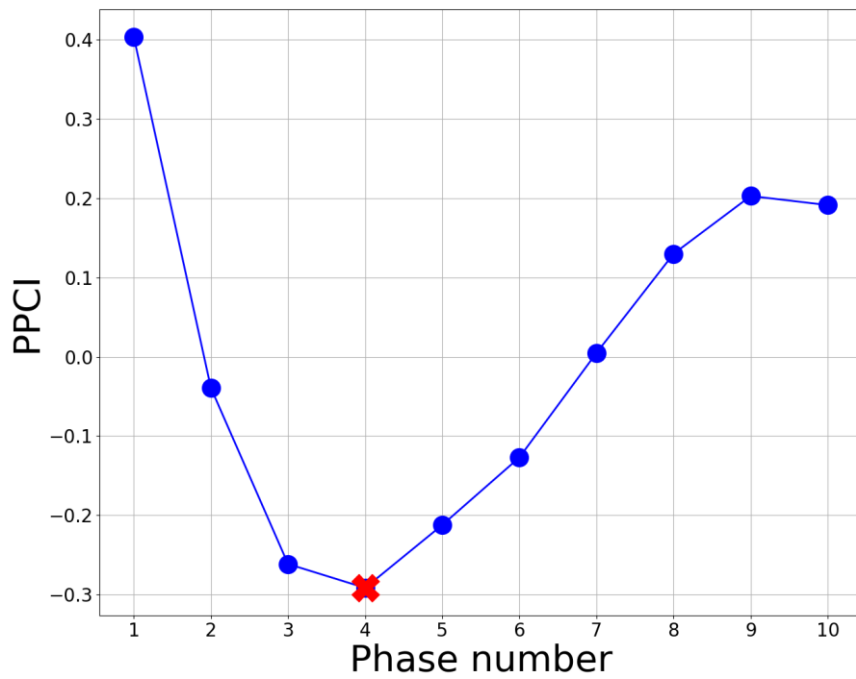


그림 15. 분할하는 phase의 개수에 따른 PPCI 값의 변화.

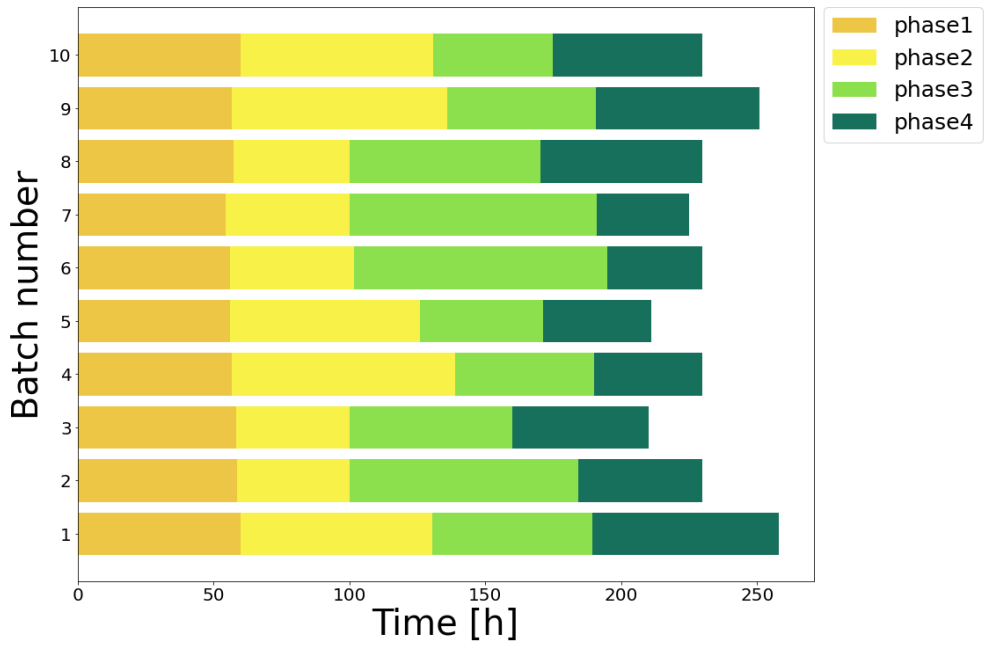


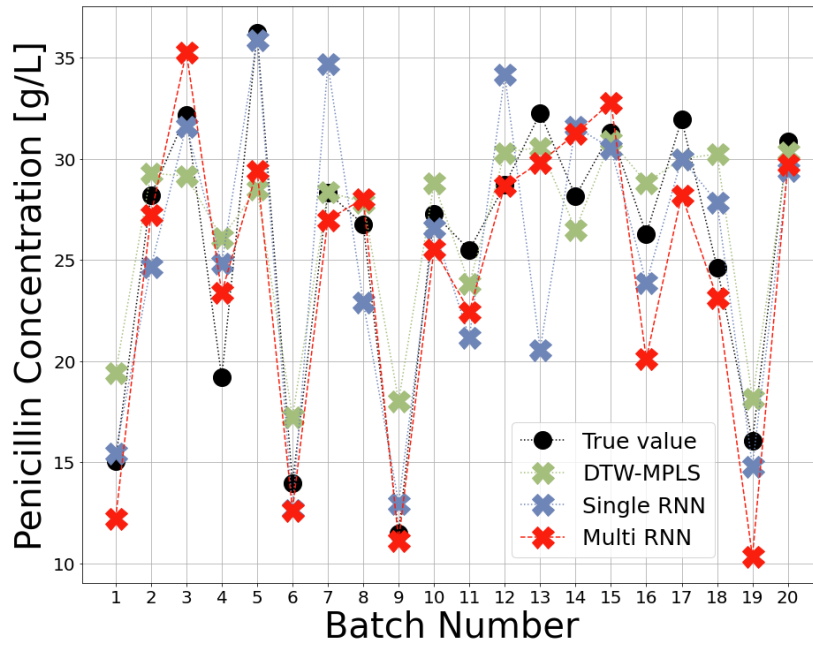
그림 16. WKM 알고리즘을 적용하여 phase를 분할한 결과.

제 2 항 Multi RNN 모델

그림 3과 같이 각 phase에 대해서 서로 다른 RNN 셀을 활용하는 형태의 multi RNN 구조를 적용하여 모델의 학습을 진행했다. 하나의 RNN 셀을 활용한 모델과 동일하게 16차원의 hidden state를 가지는 2층으로 구성된 RNN 모델을 구성했다. 최적화 모델로는 Adam optimizer를 사용했으며, 모델의 학습 속도, gradient clipping의 정도, 미니 배치의 크기, 그리고 학습 epoch는 모두 하나의 RNN 셀로 구성된 이전의 모델과 동일하게 설정했다.

학습을 완료한 모델에 대해서 앞선 경우와 동일한 테스트 데이터를 활용하여 제품의 품질 변수를 추정해보았다. 그림 17, 18, 19에서 Vanilla RNN, LSTM, 그리고 GRU 기반의 multi RNN 모델의 품질 변수 추정 결과를 DTW-MPLS 모델과 single RNN 모델의 품질 변수 추정 결과와 비교하여 표기하였고, 각 multi RNN 모델의 추정 결과로부터 parity plot을 도시하였다. 그림으로부터 multi RNN 모델이 single RNN 모델보다 대체로 좋은 품질 변수 추정 성능을 보이는 것을 확인할 수 있다.

(a)



(b)

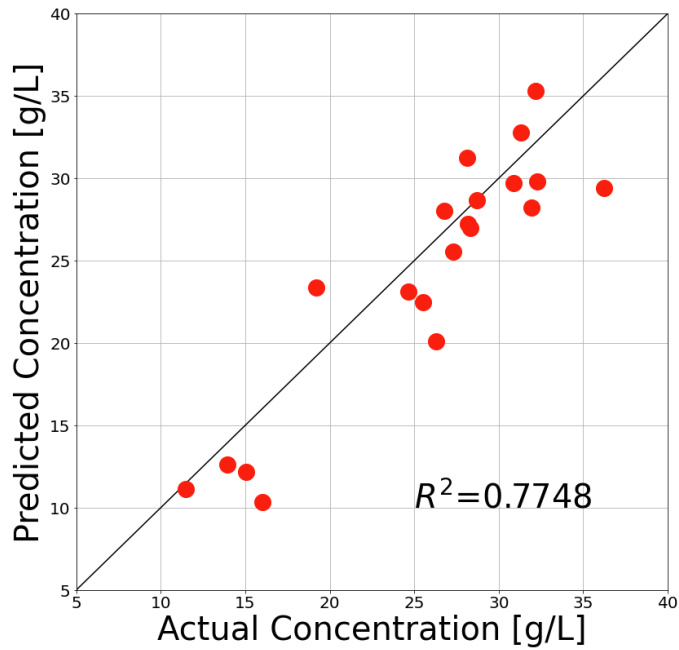
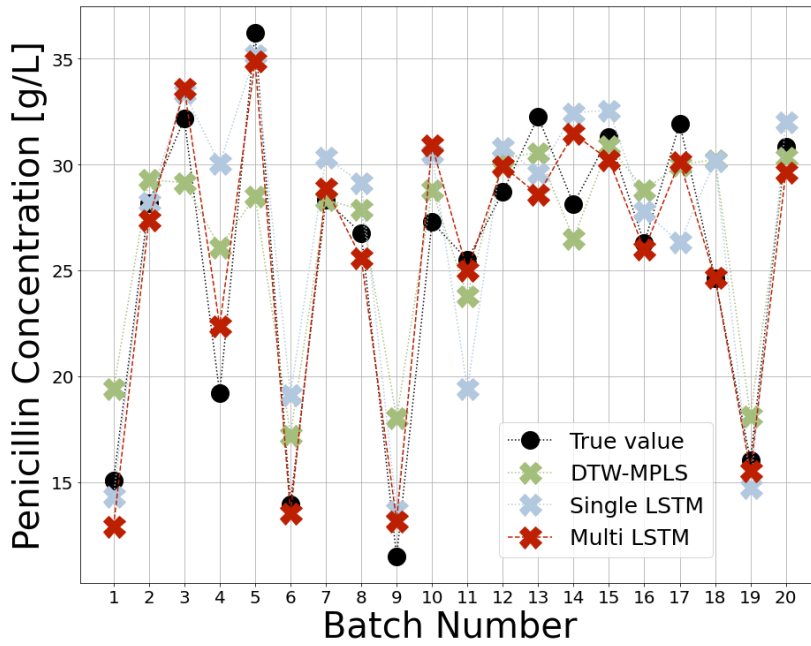


그림 17. 테스트 데이터에 대한 multi RNN 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

(a)



(b)

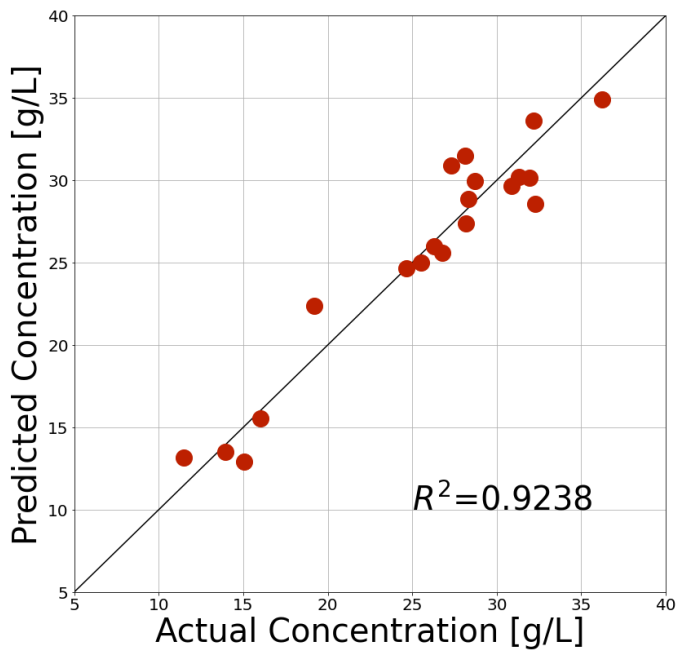
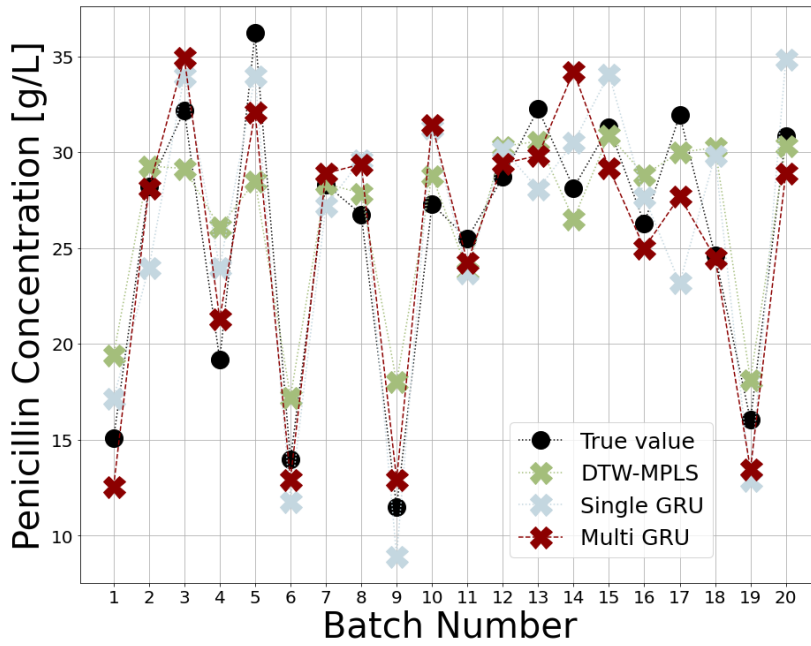


그림 18. 테스트 데이터에 대한 multi LSTM 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

(a)



(b)

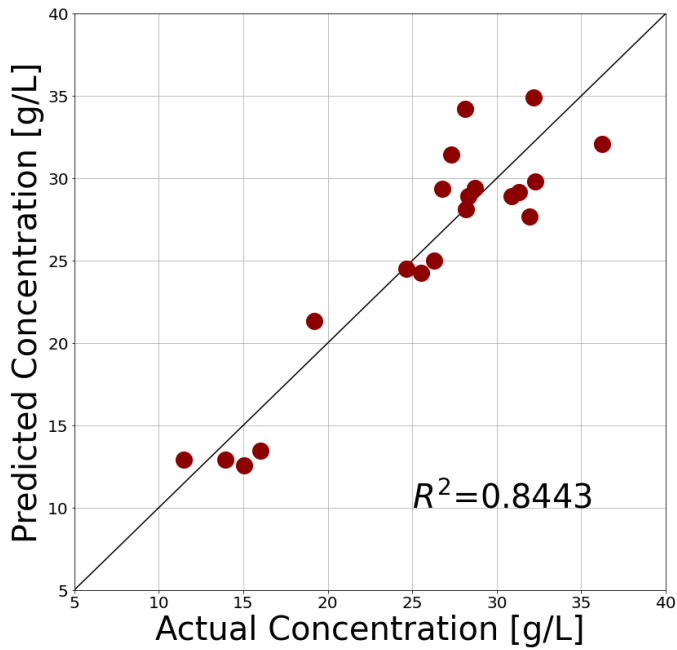


그림 19. 테스트 데이터에 대한 multi GRU 모델의 (a) 품질 예측 결과, (b) 예측 결과에 따른 parity plot.

테스트 데이터에 대해서 DTW-MPLS 모델, single RNN 모델, 그리고 multi RNN 모델의 품질 추정에서 발생한 오차는 표 3과 그림 20에서 비교할 수 있다. Vanilla RNN, LSTM, 그리고 GRU를 사용한 경우 모두에 대해서 phase 별로 다른 RNN 셀을 사용하여 학습한 multi RNN 모델이 DTW-MPLS 모델과 하나의 RNN 셀을 사용한 single RNN 모델보다 낮은 오차와 좋은 추정 성능을 보이는 것을 확인할 수 있다. Vanilla RNN을 사용한 multi RNN 모델의 품질 추정 오차는 DTW-MPLS 모델과 single RNN 모델 대비 RMSE는 각각 9.38%, 20.42%, MAPE는 각각 2.51%, 0.97% 감소하였고, R^2 score는 6.76%, 20.24% 증가하였다. LSTM을 활용한 multi LSTM 모델의 품질 추정 오차는 DTW-MPLS 모델과 single LSTM 모델 대비 RMSE는 각각 47.28%, 52.49%, MAPE는 각각 7.07%, 7.13% 감소하였고, R^2 score는 27.29%, 39.48% 증가하였다. 또한 GRU를 활용한 multi GRU 모델의 품질 추정 오차는 DTW-MPLS 모델과 single GRU 모델 대비 RMSE는 각각 24.65%, 25.65%, MAPE는 각각 4.41%, 4.14% 감소하였고, R^2 score는 16.34%, 17.53% 증가하였다.

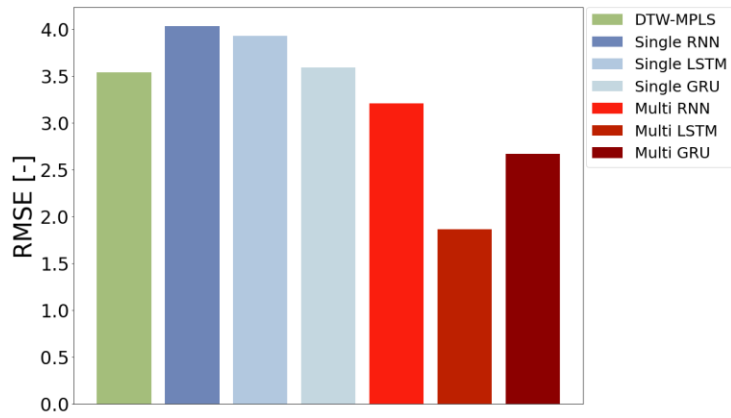
전체 모델의 품질 변수 추정 성능을 비교했을 때, phase를 분할하여 phase 별 공정 데이터를 별도로 고려했을 때 상대적으로 더 좋은 성능을 가지는 것을 확인하였다. 특히 Vanilla RNN은 single RNN 모델과 multi RNN 모델에 적용한 경우 모두 LSTM, GRU 기반의 모델 대비 낮은 성능을 보였다. 이는 Vanilla RNN이 LSTM과 GRU보다 장기 기억 의존성 문제에 취약하기 때문에 나타나는 차이인 것으로 생각된다. 또한 single RNN 모델에서는 GRU가 LSTM보다 좋은 성능을 보였지만, multi RNN 모델에서는 LSTM이 GRU보다 좋은 성능을 보인다. LSTM과 GRU는 매우 비슷한 구조를 가지고 있으며, 다양한 분야에서

비슷한 성능을 보이는 것이 보고되었다 [76,77]. 따라서 상황에 따라서 LSTM과 GRU의 성능을 비교해보고 사용할 필요가 있을 것으로 생각된다.

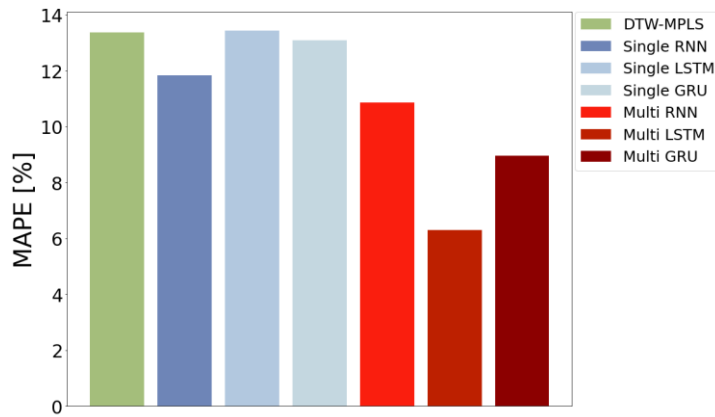
표 3. 각 모델의 품질 예측 과정에서 발생한 오차 비교.

모델	RMSE	MAPE [%]	R ² score
DTW-MPLS	3.5419	13.3698	0.7257
Single RNN	4.0334	11.8335	0.6443
Single LSTM	3.9303	13.4326	0.6623
Single GRU	3.5894	13.0920	0.7183
Multi RNN	3.2097	10.8607	0.7748
Multi LSTM	1.8672	6.3045	0.9238
Multi GRU	2.6688	8.9568	0.8443

(a)



(b)



(c)

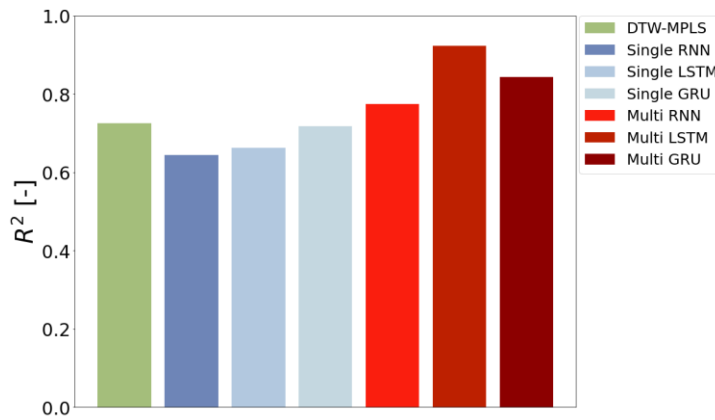


그림 20. 모델 별 (a)RMSE, (b)MAPE, (c) R^2 score의 비교

해당 모델은 hidden state로부터 품질 변수를 추정하므로 hidden state의 크기를 통해 품질 변수에 대한 각 phase의 영향력을 확인할 수 있다. 따라서 가장 좋은 추정 성능을 보인 multi LSTM 모델로부터 20개의 테스트 데이터에 대해서 hidden state를 추출해보았다. 각각의 hidden state는 식 (36)과 같은 16차원의 벡터로 표현될 수 있고, 그림 21은 각 phase에서 16차원의 hidden state의 값을 표현한 그래프이다. 또한 식 (37)과 같이 각 hidden state 벡터의 L2 norm을 계산할 수 있으며, 그림 22는 각 phase의 hidden state가 가지는 L2 norm의 평균값을 나타낸 그래프이다.

$$h_p = (h_{p,1}, \dots, h_{p,16}) \quad (366)$$

$$\|h_p\|_2 = \left(\sum_{i=1}^{16} h_{p,i}^2 \right)^{\frac{1}{2}} \quad (377)$$

그림 21, 22에서 다른 phase와 비교했을 때 phase1의 hidden state 벡터의 각 성분과 L2 norm 값이 0에 가까운 것을 확인할 수 있다. 따라서 페니실린 생산 공정의 품질 변수는 phase1 이후의 공정 변수의 변화에 영향을 많이 받는다고 생각할 수 있다. phase1은 공정 운전이 시작된 이후로 평균 57.8시간까지의 기간이며, 이 기간은 생물 반응기의 기질이 레시피에 따라 주입되는 기간이기 때문에 대부분의 배치의 공정 변수의 변화가 비슷한 경향성을 보인다. 이로 인해서 phase1의 공정 데이터가 상대적으로 품질 변수의 변화에 대해서 적은 영향력을 가지게 된 것으로 생각된다.

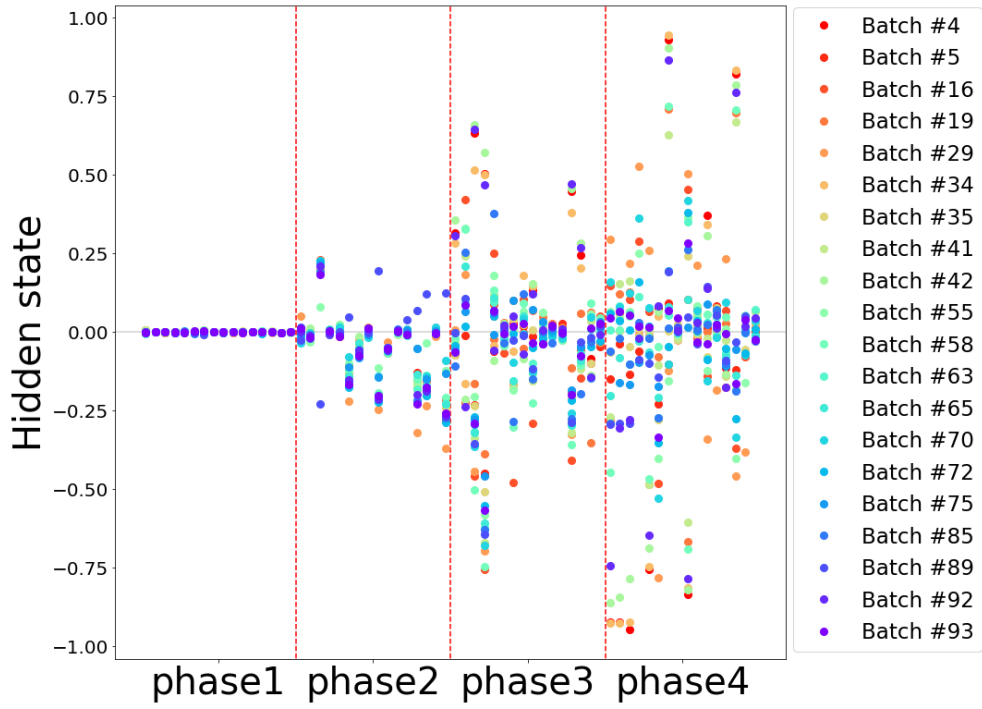


그림 21. 전체 배치 데이터에 대해서 multi LSTM 모델로부터 구한 각 phase 별 hidden state의 값.

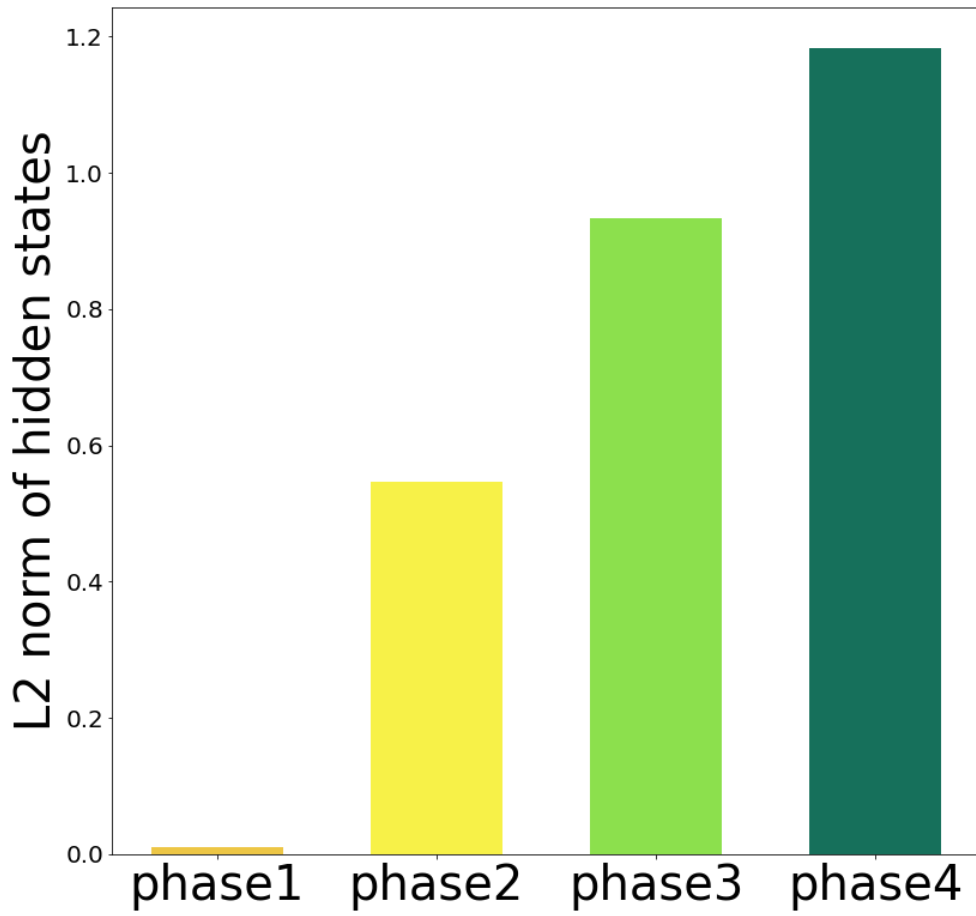


그림 22. 전체 배치 데이터에 대해서 multi LSTM 모델로 구한 phase 별 hidden state의 L2 norm의 평균값

제 5 장 결 론

본 연구에서는 운전 시간이 동일하지 않은 배치 공정의 공정 데이터에 대해서 시계열을 고려한 WKM 알고리즘을 적용하여 phase를 분할하고, phase 별로 다른 RNN 셀을 적용하여 최종적으로 생산되는 제품의 품질을 추정하는 기계 학습 기반 모델을 개발했다. 개발된 품질 변수 추정 방법론은 공정 데이터에 변형을 가하지 않음으로써, 시간 축을 정렬하여 선형 모델을 활용했던 기존의 DTW-MPLS 모델보다 좋은 추정 성능을 보였다. 또한 각 phase에서 나타나는 동특성을 개별적으로 고려함으로써, 하나의 RNN 셀을 사용하는 경우보다 좋은 성능을 보이는 것을 확인했다. 특히 Vanilla RNN, GRU에 비해서 LSTM을 활용한 경우에 가장 높은 성능을 보였으며, 학습된 LSTM에서 추출된 hidden state로부터 각 phase가 품질 변수에 미치는 영향이 다른 것을 확인했다.

본 연구에서 제안된 방법을 실제 배치 공정의 공정 데이터에 적용하여 품질 변수 추정 모델을 활용하기 위해서는 몇 가지 추가적인 연구가 필요하다. 먼저, 본 연구에서는 오프라인으로 주어진 공정 데이터만을 활용하여 학습 및 검증을 진행했지만, 실제 공정에 적용하기 위해서 온라인으로 주어지는 공정 데이터를 활용하는 방법을 고려해야 한다. 특히 온라인으로 주어지는 공정 데이터의 phase를 구분할 수 있는 방법을 고안해야 한다. 또한 대부분의 배치 공정은 주어진 레시피에 따라서 운전되기 때문에 사용할 수 있는 데이터의 양이 한정적이다. 따라서 적은 데이터로 과적합 없이 학습할 수 있는 모델의 개발이 필수적이다. 마지막으로, 개발된 모델을 기반으로 하여 공정 운전 과정에서 나타나는 이상 진단 및 최적 운전 조건을 찾아내는 연구를 수행해야 한다.

참고 문헌

- [1] P. Mhaskar, A. Garg, and B. Corbett, *Modeling and Control of Batch Processes*. Springer, 2018.
- [2] D. Wang, "Robust Data-Driven Modeling Approach for Real-Time Final Product Quality Prediction in Batch Process Operation," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 371-377, May 2011.
- [3] C. Ündey, S. Ertunç, and A. Çınar, "Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4645-4658, Jul. 2003.
- [4] C. Zhao, F. Wang, Z. Mao, N. Lu, and M. Jia, "Improved Batch Process Monitoring and Quality Prediction Based on Multiphase Statistical Analysis," *Industrial & Engineering Chemistry Research*, vol. 47, no. 3, pp. 835-849, Jan. 2008.
- [5] L. Luo, S. Bao, J. Mao, and D. Tang, "Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 150, pp. 9-22, Jan. 2016.

[6] Q. Jiang, X. Yan, H. Yi, and F. Gao, "Data-Driven Batch-End Quality Modeling and Monitoring Based on Optimized Sparse Partial Least Squares," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4098-4107, May 2020.

[7] T. Yamaguchi and Y. Yamashita, "Quality prediction for multi-grade batch process using sparse flexible clustered multi-task learning," *Computers & Chemical Engineering*, vol. 150, p. 107320, Jul. 2021.

[8] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, no. 8, pp. 1361-1375, Aug. 1994.

[9] P. Nomikos and J. F. MacGregor, "Multivariate SPC Charts for Monitoring Batch Processes," *Technometrics*, vol. 37, no. 1, pp. 41-59, Feb. 1995.

[10] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 1, pp. 97-108, Nov. 1995.

[11] J. Chen and K.-C. Liu, "On-line batch process monitoring using dynamic PCA and dynamic PLS models," *Chemical Engineering*

Science, vol. 57, no. 1, pp. 63–75, Jan. 2002.

[12] H.–J. Ramaker, E. N. M. van Sprang, J. A. Westerhuis, and A. K. Smilde, "Fault detection properties of global, local and time evolving models for batch process monitoring," *Journal of Process Control*, vol. 15, no. 7, pp. 799–805, Oct. 2005.

[13] J. Chen and H.–H. Chen, "On–line batch process monitoring using MHMT–based MPCA," *Chemical Engineering Science*, vol. 61, no. 10, pp. 3223–3239, May 2006.

[14] E. N. M. van Sprang, H.–J. Ramaker, J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Critical evaluation of approaches for on–line batch process monitoring," *Chemical Engineering Science*, vol. 57, no. 18, pp. 3979–3991, Sep. 2002.

[15] T. Chen and J. Zhang, "On–line multivariate statistical monitoring of batch processes using Gaussian mixture model," *Computers & Chemical Engineering*, vol. 34, no. 4, pp. 500–507, Apr. 2010.

[16] S. Stubbs, J. Zhang, and J. Morris, "BioProcess performance monitoring using multiway interval partial least squares," *Computer Aided Chemical Engineering*, vol. 41, pp. 243–259, 2018.

- [17] D. Neogi and C. E. Schlags, "Multivariate Statistical Analysis of an Emulsion Batch Process," *Industrial & Engineering Chemistry Research*, vol. 37, no. 10, pp. 3971–3979, Sep. 1998.
- [18] S. G. Rothwell, E. B. Martin, and A. J. Morris, "Comparison of Methods for Dealing with Uneven Length Batches," *IFAC Proceedings Volumes*, vol. 31, no. 8, pp. 387–392, May 1998.
- [19] N.-P. V. Nielsen, J. M. Carstensen, and J. Smedsgaard, "Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping," *Journal of Chromatography A*, vol. 805, no. 1–2, pp. 17–35, May 1998.
- [20] G. Tomasi, F. van den Berg, and C. Andersson, "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data," *Journal of Chemometrics*, vol. 18, no. 5, pp. 231–241, May 2004.
- [21] M. Fransson and S. Folestad, "Real-time alignment of batch process data using COW for on-line process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 84, no. 1–2, pp. 56–61, Dec. 2006.
- [22] A. Kassidas, J. F. MacGregor, and P. A. Taylor, "Synchronization

of batch trajectories using dynamic time warping," *AIChE Journal*, vol. 44, no. 4, pp. 864–875, Apr. 1998.

[23] X.-T. Doan and R. Srinivasan, "Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control," *Computers & Chemical Engineering*, vol. 32, no. 1–2, pp. 230–243, Jan. 2008.

[24] J. M. González-Martínez, A. Ferrer, and J. A. Westerhuis, "Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping," *Chemometrics and Intelligent Laboratory Systems*, vol. 105, no. 2, pp. 195–206, Feb. 2011.

[25] Y. Yao and F. Gao, "A survey on multistage/multiphase statistical modeling methods for batch processes," *Annual Reviews in Control*, vol. 33, no. 2, pp. 172–183, Dec. 2009.

[26] C. Zhao, S. Mo, F. Gao, N. Lu, and Y. Yao, "Statistical analysis and online monitoring for handling multiphase batch processes with varying durations," *Journal of Process Control*, vol. 21, no. 6, pp. 817–829, Jul. 2011.

[27] K. A. Kosanovich, M. J. Piovoso, K. S. Dahl, J. F. MacGregor,

and P. Nomikos, "Multi-way PCA applied to an industrial batch process," *Proceedings of 1994 American Control Conference - ACC '94*, vol. 2, Jun. 1994.

[28] D. Dong and T. J. McAvoy, "Multi-stage batch process monitoring," *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, Jun. 1995.

[29] C. Undey and A. Cinar, "Statistical monitoring of multistage, multiphase batch processes," *IEEE Control Systems*, vol. 22, no. 5, pp. 40–52, Oct. 2002.

[30] N. Lu, F. Gao, and F. Wang, "Sub-PCA modeling and on-line monitoring strategy for batch processes," *AIChE Journal*, vol. 50, no. 1, pp. 255–259, Jan. 2004.

[31] J. Camacho and J. Picó, "Multi-phase principal component analysis for batch processes modelling," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 2, pp. 127–136, Apr. 2006.

[32] J. Yu and S. J. Qin, "Multiway Gaussian Mixture Model Based Multiphase Batch Process Monitoring," *Industrial & Engineering*

Chemistry Research, vol. 48, no. 18, pp. 8585–8594, Aug. 2009.

[33] N. Lu, F. Gao, Y. Yang, and F. Wang, "PCA–Based Modeling and On–line Monitoring Strategy for Uneven–Length Batch Processes," *Industrial & Engineering Chemistry Research*, vol. 43, no. 13, pp. 3343–3352, May 2004.

[34] L. Zhao, C. Zhao, and F. Gao, "Inner–Phase Analysis Based Statistical Modeling and Online Monitoring for Uneven Multiphase Batch Processes," *Industrial & Engineering Chemistry Research*, vol. 52, no. 12, pp. 4586–4596, Mar. 2013.

[35] Z. Ge and Z. Song, "Online Monitoring and Quality Prediction of Multiphase Batch Processes with Uneven Length Problem," *Industrial & Engineering Chemistry Research*, vol. 53, no. 2, pp. 800–811, Jan. 2014.

[36] W. Li, C. Zhao, and F. Gao, "Sequential Time Slice Alignment Based Unequal–Length Phase Identification and Modeling for Fault Detection of Irregular Batches," *Industrial & Engineering Chemistry Research*, vol. 54, no. 41, pp. 10020–10030, Oct. 2015.

[37] L. Luo, S. Bao, J. Mao, and D. Tang, "Phase Partition and Phase–Based Process Monitoring Methods for Multiphase Batch Processes

with Uneven Durations," *Industrial & Engineering Chemistry Research*, vol. 55, no. 7, pp. 2035–2048, Feb. 2016.

[38] J.-M. Lee, C. Yoo, and I.-B. Lee, "Fault detection of batch processes using multiway kernel principal component analysis," *Computers & Chemical Engineering*, vol. 28, no. 9, pp. 1837–1847, Aug. 2004.

[39] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223–234, Jan. 2004.

[40] Y. Zhang and Z. Hu, "On-line batch process monitoring using hierarchical kernel partial least squares," *Chemical Engineering Research and Design*, vol. 89, no. 10, pp. 2078–2084, Oct. 2011.

[41] Y. Hu, H. Ma, and H. Shi, "Enhanced batch process monitoring using just-in-time-learning based kernel partial least squares," *Chemometrics and Intelligent Laboratory Systems*, vol. 123, pp. 15–27, Apr. 2013.

[42] A. Y.-D. Tsen, S. S. Jang, D. S. H. Wong, and B. Joseph, "Predictive control of quality in batch polymerization using hybrid

ANN models," *AIChE Journal*, vol. 42, no. 2, pp. 455–465, Feb. 1996.

[43] J. Zhang, A. J. Morris, E. B. Martin, and C. Kiparissides, "Prediction of polymer quality in batch polymerisation reactors using robust neural networks," *Chemical Engineering Journal*, vol. 69, no. 2, pp. 135–143, Mar. 1998.

[44] Z. Yan, C.-C. Chiu, W. Dong, and Y. Yao, "A LASSO-based batch process modeling and end-product quality prediction method," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 6704–6709, 2014.

[45] C. Yang and J. Hou, "Fed-batch fermentation penicillin process fault diagnosis and detection based on support vector machine," *Neurocomputing*, vol. 190, pp. 117–123, May 2016.

[46] M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas, and E. N. Pistikopoulos, "Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection," *Computers & Chemical Engineering*, vol. 115, pp. 46–63, Jul. 2018.

[47] J. Zhu and F. Gao, "Improved Nonlinear Quality Estimation for Multiphase Batch Processes Based on Relevance Vector Machine with Neighborhood Component Variable Selection," *Industrial &*

Engineering Chemistry Research, vol. 57, no. 2, pp. 666–676, Jan. 2018.

[48] K. Qiu, J. Wang, R. Wang, Y. Guo, and L. Zhao, "Soft sensor development based on kernel dynamic time warping and a relevant vector machine for unequal-length batch processes," *Expert Systems with Applications*, vol. 182, p. 115223, Nov. 2021.

[49] H. Jin, X. Chen, L. Wang, K. Yang, and L. Wu, "Adaptive Soft Sensor Development Based on Online Ensemble Gaussian Process Regression for Nonlinear Time-Varying Batch Processes," *Industrial & Engineering Chemistry Research*, vol. 54, no. 30, pp. 7320–7345, Jul. 2015.

[50] K. Yang, H. Jin, X. Chen, J. Dai, L. Wang, and D. Zhang, "Soft sensor development for online quality prediction of industrial batch rubber mixing process using ensemble just-in-time Gaussian process regression models," *Chemometrics and Intelligent Laboratory Systems*, vol. 155, pp. 170–182, Jul. 2016.

[51] C. Mei, Y. Su, G. Liu, Y. Ding, and Z. Liao, "Dynamic soft sensor development based on Gaussian mixture regression for fermentation processes," *Chinese Journal of Chemical Engineering*, vol. 25, no. 1, pp. 116–122, Jan. 2017.

- [52] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, May 2009.
- [53] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.
- [54] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
- [55] Y. Tian, J. Zhang, and J. Morris, "Optimal control of a fed-batch bioreactor based upon an augmented recurrent neural network model," *Neurocomputing*, vol. 48, no. 1–4, pp. 919–936, Oct. 2002.
- [56] Z. Xiong and J. Zhang, "A batch-to-batch iterative optimal control strategy based on recurrent neural network models," *Journal of Process Control*, vol. 15, no. 1, pp. 11–21, Feb. 2005.
- [57] W. Guo, T. Pan, Z. Li, S. Ding, and M. Wang, "Modelling for

Multi-Phase Batch Processes using Steady State Identification and Deep Recurrent Neural Network," *2019 12th Asian Control Conference (ASCC)*, Jun. 2019.

[58] K. Wang, R. B. Gopaluni, J. Chen, and Z. Song, "Deep Learning of Complex Batch Process Data and Its Application on Quality Prediction," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7233–7242, Dec. 2020.

[59] F. Shen, J. Zheng, L. Ye, and X. Ma, "LSTM Soft Sensor Development of Batch Processes with Multivariate Trajectory-Based Ensemble Just-in-Time Learning," *IEEE Access*, vol. 8, pp. 73855–73864, 2020.

[60] J. Ren and D. Ni, "A batch-wise LSTM-encoder decoder network for batch process monitoring," *Chemical Engineering Research and Design*, vol. 164, pp. 102–112, Dec. 2020.

[61] J. Zheng, F. Shen, and L. Ye, "Improved Mahalanobis Distance Based JITL-LSTM Soft Sensor for Multiphase Batch Processes," *IEEE Access*, vol. 9, pp. 72172–72182, 2021.

[62] R. Srinivasan and M. Qian, "Online Temporal Signal Comparison Using Singular Points Augmented Time Warping," *Industrial &*

Engineering Chemistry Research, vol. 46, no. 13, pp. 4531–4548, Jun. 2007.

[63] P. Geladi and B. R. Kowalski, "Partial least–squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, pp. 1–17, 1986.

[64] L. A. Leiva and E. Vidal, "Warped K–Means: An algorithm to cluster sequentially–distributed data," *Information Sciences*, vol. 237, pp. 196–210, Jul. 2013.

[65] M. Kuhn, H. Tomaschewski, and H. Ney, "Fast nonlinear time alignment for isolated word recognition," *ICASSP '81. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, Mar. 1981.

[66] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State–of–the–art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018.

[67] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, Dec. 2018.

[68] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification:

An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, Sep. 2020.

[69] S. Hochreiter and J. Schmidhuber, "Long Short–Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[70] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, Dec. 2014.

[71] H. J. Lee, S. Lee, and J. M. Lee, "Online Synchronization in Latent Variable Model Predictive Control for Trajectory Tracking of an Uneven Batch Process," *Industrial & Engineering Chemistry Research*, vol. 61, no. 1, pp. 594–604, Dec. 2021.

[72] H. J. Lee, J. Bae, D. H. Jeong, and J. M. Lee, "Real–time synchronization with expected distribution of synchronized index for on–line monitoring of uneven multiphase batch process," *Computers & Chemical Engineering*, vol. 154, p. 107490, Nov. 2021.

[73] T. H. Oh, H. M. Park, J. W. Kim, and J. M. Lee, "Integration of reinforcement learning and model predictive control to optimize semi-batch bioreactor," *AIChE Journal*, vol. 68, no. 6, Mar. 2022.

[74] S. Goldrick, A. Ştefan, D. Lovett, G. Montague, and B. Lennox,

"The development of an industrial-scale fed-batch fermentation simulation," *Journal of Biotechnology*, vol. 193, pp. 70–82, Jan. 2015.

[75] S. Goldrick, C. A. Duran–Villalobos, K. Jankauskas, D. Lovett, S. S. Farid, and B. Lennox, "Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process," *Computers & Chemical Engineering*, vol. 130, p. 106471, Nov. 2019.

[76] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019.

[77] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pp. 98–101, Jun. 2020.

Abstract

Machine Learning Based Approaches to Estimation of Quality Variables in Batch Processes

Joonsoo Park

School of Chemical and Biological Engineering

The Graduate School

Seoul National University

It is necessary to develop a product quality estimation model to operate batch processes in a safe and economical way. However, since the batch processes are operated in an unsteady state and show strong nonlinearity, it is difficult to build a first principle model. In addition, the batch processes show uneven operating duration for each batch and consist of several phases according to operating conditions, which makes the development of a quality estimation model more difficult. In this study, a machine learning based approach to estimation of quality variables considering the uneven duration and the multi-phase of batch processes was developed. The phases were divided by applying the warped K-means clustering (WKM) algorithm considering the sequentiality of process data. The quality

estimation model in the form of a separate Recurrent Neural Network (RNN) cell for each phase was used. The developed quality estimation model showed better performance than the model using dynamic time warping (DTW) and multiway partial least squares (MPLS). In addition, using different RNN cells for each phase shows better performance than using one RNN cell without distinguishing the phases. By developing the quality estimation model into a form that can be used online, it will be possible to use the developed model for diagnosing abnormalities and searching for optimal operating conditions.

Keywords: Batch process, Quality estimation, Warped K-means clustering, Recurrent neural network

Student Number: 2021-25124