

Procedure for monitoring autocorrelated processes using LSTM Autoencoder

Pyoungjin Ji^a, Jaeheon Lee^{1, a}

^aDepartment of Applied Statistics, Chung-Ang University

Abstract

Many studies have been conducted to quickly detect out-of-control situations in autocorrelated processes. The most traditionally used method is a residual control chart, which uses residuals calculated from a fitted time series model. However, many procedures for monitoring autocorrelated processes using statistical learning methods have recently been proposed. In this paper, we propose a monitoring procedure using the latent vector of LSTM Autoencoder, a deep learning-based unsupervised learning method. We compare the performance of this procedure with the LSTM Autoencoder procedure based on the reconstruction error, the RNN classification procedure, and the residual charting procedure through simulation studies. Simulation results show that the performance of the proposed procedure and the RNN classification procedure are similar, but the proposed procedure has the advantage of being useful in processes where sufficient out-of-control data cannot be obtained, because it does not require out-of-control data for training.

Keywords: autocorrelated process, latent vector, LSTM Autoencoder, unsupervised learning

1. 서론

자기상관 공정이란 공정의 특성으로 인해 공정의 양적 지표인 품질특성치(quality characteristic)가 시간에 따라 상관성을 가지는 공정을 말한다. 자기상관 공정의 경우 품질특성치가 서로 독립일 때와 달리 전통적인 관리도(control chart)를 사용한 공정 모니터링 절차를 적용할 경우 오경보율(false alarm rate)이 사전에 설정한 값에 비해 커진다는 문제가 발생한다.

이러한 문제를 해결하는 전통적인 관리도 절차 중 가장 빈번하게 사용되는 것은 잔차(residual) 관리도 절차이다. 이를 위해 자기상관이 존재하는 공정 데이터에 대해 적절한 시계열 모형에 적합시켜 잔차를 계산한 후 이 잔차에 대해 관리도 절차를 적용하는 것이다. 만일 시계열 모형의 적합이 정확하게 진행되었다면 잔차들은 어느 정도 독립성을 만족한다는 사실을 이용한 것이다. 이 논문에서는 잔차들에 대해 Shewhart의 개별 관측값 관리도를 고려하였다. 잔차 관리도에 대한 좀 더 상세한 설명은 Montgomery (2020)의 10.4절을 참고할 수 있다.

이와는 다른 방법으로 자기상관 공정의 모니터링 절차에 대한 문제를 해결하기 위해 최근 통계적 학습(statistical learning)을 사용한 연구들이 활발하게 진행되었다. Ho와 Chang (1999), Hwang (2004), Hwang과 Wang (2010), Yuangyai와 Abrahams (2011), Prajapati와 Singh (2015), 그리고 White와 Safi (2016)는 ANN (artificial neural network) 모형을 사용하여 자기상관 공정을 모니터링하는 절차에 대해 연구하였다. Chen과 Yu (2019)와 Lee와 Lee (2022)는 잔차에 대한 모니터링 절차를 제안하였는데, 각각 딥러닝에 기반한 RNN

¹Corresponding author: Department of Applied Statistics, Chung-Ang University, 84 Heukseok-Ro, Dongjak-Gu, Seoul 06974, Korea. E-mail: jaeheon@cau.ac.kr

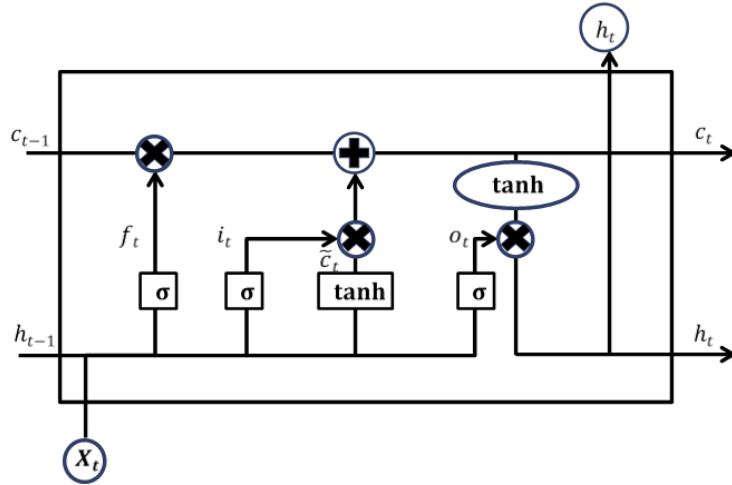


Figure 1: Structure of LSTM cell.

(recurrent neural network) 모형과 LSTM (long short-term memory) 모형을 사용하였다. Ji와 Lee (2023)는 RNN 모형을 사용하여 공정의 데이터가 관리상태와 이상상태로 구분되어 있는 경우, 이에 대한 패턴을 분석하고 공정의 이상상태 여부를 분류하는 지도 학습(supervised learning) 모형의 공정 모니터링 절차를 제안하였다. Maged 등 (2023)은 Variational Autoencoder 모형을 사용하여 차원이 축소된 데이터의 정보를 Hotelling의 T^2 관리도에 적용하는 모니터링 절차에 대해 연구하였다.

일반적으로 학습을 위해 이상상태의 데이터를 충분히 확보하고 있는 경우가 많지 않기 때문에, 최근 비지도 학습(unsupervised learning) 모형은 그 사용 빈도가 급격하게 증가하고 있는 실정이다. 따라서 이 논문은 비지도 학습 모형 중 하나인 LSTM Autoencoder를 사용하여 공정의 특성을 학습시키고 이를 이용하여 공정의 이상상태 여부를 판단하는 모니터링 절차를 제안하고 있다. 제안된 절차의 성능을 평가하기 위해 지도 학습 모형을 사용한 RNN 분류 모니터링 절차 및 전통적인 잔차 관리도 절차의 성능과 비교하였다. 실제 생산 공정에서 발생하는 자기상관 공정은 복잡한 모형이 아니라 비교적 단순한 자기회귀(autoregressive; AR) 모형에 적합되는 경우가 대부분이다. 절차의 성능을 평가하는 모의실험에서는 가장 흔하게 관측되는 AR(1), AR(2), 그리고 ARMA(1, 1) 모형을 고려하였다. 더불어, 실제 공정 데이터에 대해 제안된 모니터링 절차를 적용하여 공정의 변화에 따른 이상상태 탐지 성능을 확인하였다.

이 논문은 2장에서 LSTM Autoencoder에 대한 소개 및 설명을 하고, 3장에서 이 논문에서 제안하는 LSTM Autoencoder의 잠재 벡터를 이용한 모니터링 절차와 모의실험에 대한 설계 및 결과를 제시한다. 그리고 4장에 실제 공정 데이터에 대해 적용한 결과를 제시하고, 마지막으로 5장에 논문의 결과를 요약한다.

2. 이론적 배경

2.1. LSTM

LSTM Autoencoder를 소개하기에 앞서 LSTM에 대한 설명을 하고자 한다. LSTM 모형은 RNN의 단점을 보완하여 Hochreiter와 Schmidhuber (1997)에 의해 처음 제안되었다. LSTM의 구조를 Figure 1에 나타내었다. LSTM 모형은 크게 셀, 망각 게이트(forget gate; f_t), 입력 게이트(input gate; i_t), 그리고 출력 게이트(output gate; o_t)로 구성되어 있으며, 이 게이트들을 통해 학습을 진행한다. Figure 1에서 망각 게이트는 이전 시점에서 학습된 셀 c_{t-1} 의 정보를 보존할지 여부를 정하고, 입력 게이트는 셀 c_t 로부터 추가되는 정보 중 어떤 것을

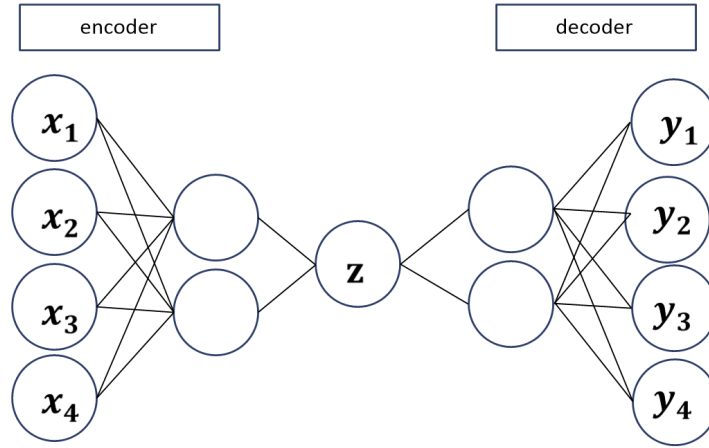


Figure 2: Structure of Autoencoder.

가져갈지 판단하며, 출력 게이트는 c_t 의 정보 중 어떤 정보를 내보낼지 결정한다.

다음은 은닉층 h_t 가 계산되는 과정을 식으로 나타낸 것이다.

$$\begin{aligned}
 f_t &= \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f), \\
 i_t &= \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i), \\
 \tilde{c}_t &= \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c), \\
 o_t &= \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o), \\
 c_t &= (c_{t-1} \times f_t) + (\tilde{c}_t \times i_t), \\
 h_t &= \tanh(c_t) \times o_t.
 \end{aligned}$$

위의 식에서 σ 는 시그모이드(sigmoid) 함수, W 는 가중치 행렬, b 는 각 게이트에 대한 편향, 그리고 연산 '×'는 Hadamard 곱을 나타낸다. 이러한 구조 때문에 LSTM은 자기상관이 존재하는 데이터의 학습에 효율적이라고 알려져 있으며, 따라서 시계열 데이터의 학습에 많이 사용되고 있다.

2.2. Autoencoder

Autoencoder는 데이터의 차원을 축소시켰다 확장하며 다시 원래의 입력 데이터로 복원하는 인공지능망의 한 종류이다 (Kramer, 1991, 1992). Autoencoder의 구조에 대해 Figure 2에 제시하였다. Autoencoder는 입력된 데이터의 정보를 압축하는 부분인 encoder와 압축된 데이터를 다시 복원하는 부분인 decoder로 이루어져 있다. Figure 2의 잠재 벡터(latent vector) z 는 encoder의 데이터의 압축이 완료된 상태로, 데이터가 모형 내에서 가장 추상화된 정보를 가지고 있다. 데이터의 정보를 압축했다 다시 복원하는 과정을 거치는 구조이므로, 데이터 라벨링(data labelling)이 필요하지 않은 비지도 학습의 신경망 모형이다. 따라서 데이터의 상태에 대한 정보가 필요 없고 이상상태의 데이터가 없는 데이터의 학습에 유용하다는 장점이 있다.

Autoencoder는 데이터의 특징을 추출하여 압축한 후, 해당 특징으로 데이터를 다시 복원하는 것에 목적을 두어 학습을 진행한다. 따라서 관리상태의 데이터가 학습된 모형에 이상상태의 데이터가 입력되었을 때 복원 오차(reconstruction error)가 관리상태 데이터의 복원 오차보다 커진다는 점을 이용하여, 데이터의 비정상

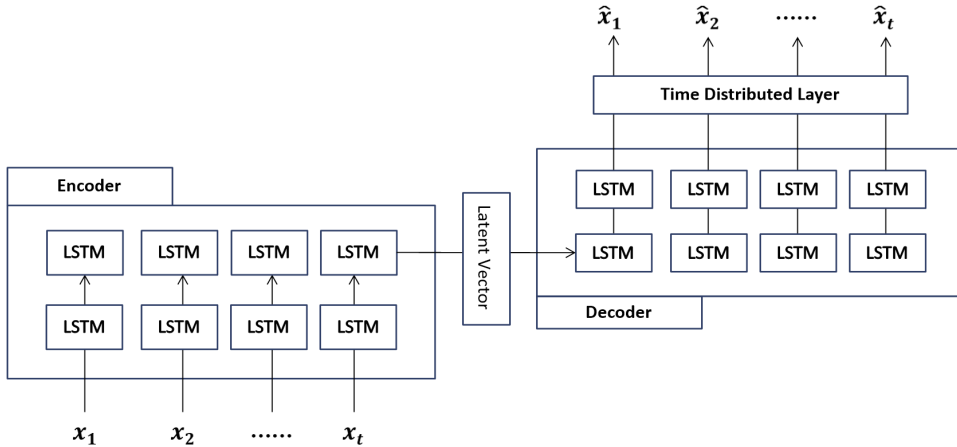


Figure 3: Structure of LSTM Autoencoder.

여부를 탐지하고 있다. 또한 잠재 벡터는 데이터의 가장 추상화된 정보를 압축해서 가지고 있으므로, 차원 축소에도 이용된다.

2.3. LSTM Autoencoder

LSTM Autoencoder는 Srivastava 등 (2015)이 제안한 모형으로 Autoencoder의 셀이 LSTM으로 구성되어 있는 구조이다. LSTM Autoencoder는 시계열 데이터의 학습에 유리한 LSTM과 데이터의 비정상 여부를 빠르게 탐지할 수 있는 Autoencoder의 장점을 모두 가지고 있다. LSTM Autoencoder의 학습은 Autoencoder와 마찬가지로 encoder에서 데이터를 압축하고 decoder에서 데이터를 다시 복원하는 과정을 통해 학습을 진행한다.

LSTM Autoencoder의 구조에 대해 Figure 3에 제시하였다. Figure 3는 시점 t 에서의 입력 데이터 $\mathbf{x}_t = [x_{t1}, x_{t2}, \dots, x_{tm}]$ 가 각각 2개의 LSTM층으로 구성된 encoder와 decoder를 지나면서 압축과 복원을 거쳐 $\hat{\mathbf{x}}_t$ 가 출력되는 과정을 나타낸다. 잠재 벡터는 데이터가 입력되어 encoder를 거쳐 압축된 상태로 decoder를 지나면서 복원된다. Time distributed layer는 입력 데이터의 각 time step마다 계산된 오류를 하위 time step으로 전파하여 모형의 가중치를 업데이트하는 역할을 한다.

3. 절차의 제안 및 모의실험

3.1. 잠재벡터를 이용한 공정 모니터링 절차

최근 LSTM Autoencoder를 이용하여 시계열 데이터의 이상치를 탐지하는 절차에 대한 연구가 진행되었다 (Homayouni 등, 2020; Wei 등, 2022). 이 절차는 encoder와 decoder를 사용하여 압축과 복원을 한 후 복원 오차를 기준으로 이상치를 판단하는 것이다. 그러나 시계열 데이터의 이상치와 공정의 이상상태는 연속적인 상태의 유지 여부에 따라 그 특성에 차이가 있는데, 이 논문에서 제안하는 공정 모니터링 절차는 LSTM Autoencoder에서 encoder로 압축된 잠재 벡터에 기초한 절차이다.

모형의 학습에 이용된 데이터는 관리상태에서 생성된 시계열 데이터를 이동 윈도우 방식으로 구성한 크기가 12인 윈도우 데이터이다. 학습을 위하여 학습 데이터 1,000개와 검증 데이터 500개의 윈도우 데이터를 사용하였고, 손실함수는 평균제곱오차(mean squared error; MSE)를 사용하였다. 모니터링 절차의 간편화를 위해 잠재 벡터의 차원은 1로 지정하였다. 크기가 12인 윈도우 데이터를 크기가 1인 잠재 벡터로 압축했기

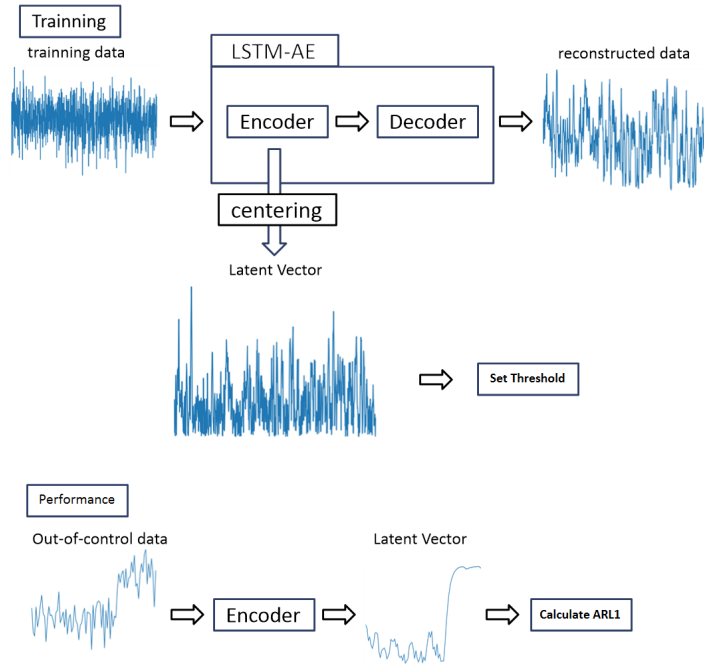


Figure 4: Flowchart of the proposed monitoring procedure.

때문에, 정보의 손실은 있을 수 있지만 관리상태와 이상상태의 판단 기준으로 사용할 때 절차를 간편하게 설계할 수 있다는 장점이 있다. 복원 오차와 잠재 벡터의 비교에 대해서는 이후에 다시 언급할 것이다.

잠재 벡터를 이용한 공정 모니터링 절차의 이상상태 판단 기준은 식 (3.1)과 같다. 식 (3.1)에서 l_t 는 시점 t 에서의 잠재 벡터, $\text{median}(l_{train})$ 은 학습 데이터에서 생성된 잠재 벡터들의 중앙값, 그리고 h 는 이상상태 판단의 임계치(threshold)를 나타낸다. 잠재 벡터의 값으로 이상상태 여부를 좀 더 쉽게 판단하기 위해 우리는 중심화(centering) 과정을 수행하였다. 몇 가지 경우에 대해 중심화 과정을 수행한 결과, 평균을 사용하는 것보다 중앙값을 사용하는 것이 더 효과적이라고 판단되어 중앙값을 사용하여 중심화 과정을 수행하였다. 제안하는 모니터링 절차의 순서를 정리하여 Figure 4에 나타내었다.

$$|l_t - \text{median}(l_{train})| > h. \tag{3.1}$$

앞에서 언급한 바와 같이, 기존의 LSTM Autoencoder를 이용한 이상치 판단은 입력된 데이터와 복원된 데이터의 차이인 복원 오차를 사용한다. 공정 모니터링 절차도 이와 유사하다. 즉, 모형 학습시 관리상태의 공정 데이터를 학습하기 때문에, 모형에서 복원 가능한 범위는 관리상태인 공정 데이터의 범위에 한정되게 된다. 달리 말해, 공정에 변화가 발생한 윈도우 데이터가 모형에 입력되었을 때 모형은 학습된 데이터의 범위를 기반으로 입력된 윈도우와 가장 가까운 윈도우를 출력한다. 따라서 변화의 크기가 큰 경우에는 입력 윈도우와 출력 윈도우의 차이인 복원 오차가 크게 증가하여 이상상태를 빠르게 탐지하지만, 변화의 크기가 작은 경우에는 복원 오차에 큰 변화가 없어서 임계치를 벗어날 때까지 소요되는 시간이 아주 길어지거나 변화를 탐지하지 못하는 경우도 발생할 수 있다.

Figure 5는 AR(1) 모형 (모형에 대해서는 다음 절에서 설명)에서 500 번째 시점에 공정 수준의 변화가 발생한 경우 복원 오차와 잠재 벡터의 변화를 기록한 그래프이다. 큰 변화가 발생한 경우 복원 오차와 잠재 벡터가

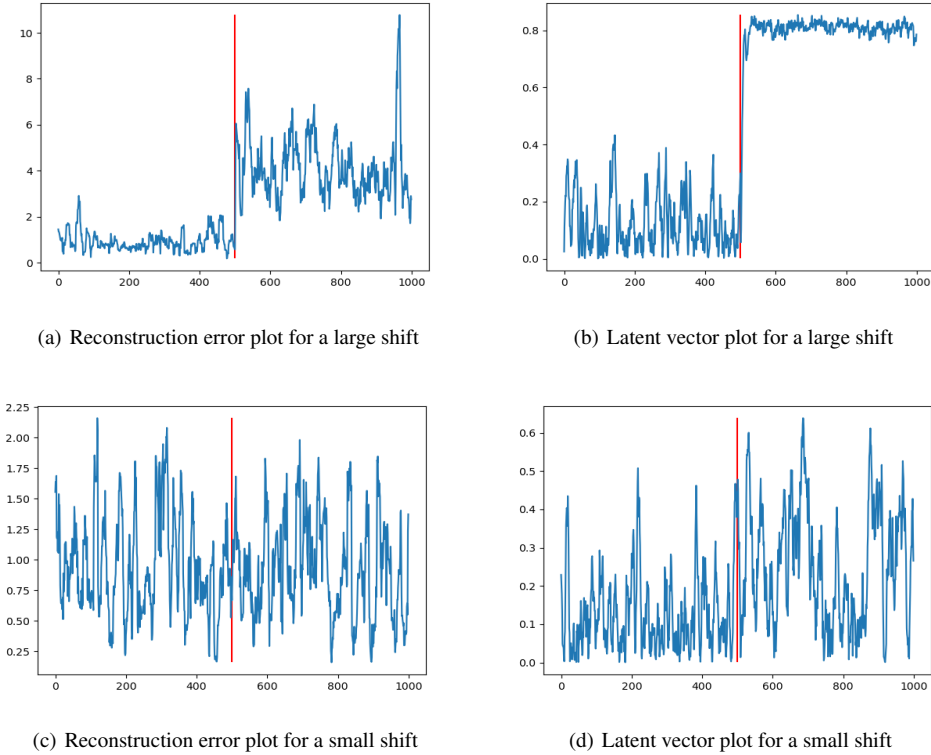


Figure 5: Reconstruction error plots and latent vector plots.

모두 공정의 변화에 민감하게 반응하는 것으로 나타났다. 그러나 작은 변화가 발생한 경우 복원 오차에는 큰 변화가 나타나지 않지만 잠재 벡터는 복원 오차보다 변화에 좀 더 민감하게 반응하는 것을 확인할 수 있다.

3.2. 모의실험의 설계

생산 공정에 표본추출간격이 작은 경우 표본추출간격 내에서 관성적인 요소가 사라지지 않아서 데이터가 시간에 따라 서로 상관될 수 있다. 이때 자기상관된 공정의 데이터에 많이 적합되는 모형은 AR(1), AR(2), 그리고 AR(1) 모형에 랜덤 오차가 추가된 모형이다. 그런데 Lu와 Reynolds (1999)는 AR(1) 모형에 랜덤 오차가 추가된 모형이 ARMA(1, 1) 모형과 동치라는 사실을 유도하였다. 따라서 이 논문의 모의실험에서는 AR(1), AR(2), 그리고 ARMA(1, 1) 모형을 고려하였고, 모형의 식을 각각 다음과 같이 표현하였다.

$$x_t = \xi_t + \phi x_{t-1} + \epsilon_t,$$

$$x_t = \xi_t + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \epsilon_t,$$

$$x_t = \xi_t + \phi x_{t-1} + \epsilon_t - \theta \epsilon_{t-1}.$$

이 논문에서 이상상태는 공정의 수준, 분산, 그리고 수준과 분산이 동시에 변화하는 3가지 경우를 고려한다. 공정의 상태는 시점 $t \leq t^* - 1$ 에서 관리상태이고 시점 $t^* - 1$ 과 t^* 사이에 변화가 발생하여 시점 $t \geq t^*$ 에서 이상상태가 된다고 가정한다. 위의 모형에서 ξ_t 와 ϵ_t 의 분포는 다음과 같이 정의한다.

Table 1: Variance of the autocorrelated models, σ_x^2

AR(1)	$\frac{\sigma_\epsilon^2}{1 - \phi^2}$
AR(2)	$\frac{(1 - \phi_2)\sigma_\epsilon^2}{(1 + \phi_2)(1 - \phi_2 - \phi_1)(1 - \phi_2 + \phi_1)}$
ARMA(1, 1)	$\frac{(1 + 2\phi\theta + \theta^2)\sigma_\epsilon^2}{1 - \phi^2}$

$$\xi_t = \begin{cases} \xi, & t \leq t^* - 1 \text{인 경우,} \\ \xi + \delta\sigma_x, & t \geq t^* \text{인 경우.} \end{cases} \quad (3.2)$$

$$\epsilon_t \sim \begin{cases} N(0, \sigma_\epsilon^2), & t \leq t^* - 1 \text{인 경우,} \\ N(0, \gamma^2\sigma_\epsilon^2), & t \geq t^* \text{인 경우,} \end{cases} \quad (3.3)$$

여기서 σ_x 는 공정이 관리상태일 때 표준편차로서 각 시계열 모형에 대한 σ_x^2 을 Table 1에 정리하였다. 모의실험에서 공정 수준의 변화에 대한 모수 δ 는 0.5, 1, 2, 3, 공정 분산의 변화에 대한 모수 γ 는 1.5, 2, 3을 사용했고, $\xi = 0$ 과 $\sigma_\epsilon^2 = 1$ 을 가정하였다.

일반적으로 공정 모니터링 절차의 성능 평가는 평균 런길이(average run length; ARL)를 사용한다. 런길이(run length)는 모니터링 절차의 통계량이 임계치를 벗어나서 이상상태라는 신호를 발생할 때까지 관측한 표본의 수로 정의되고, ARL은 런길이의 평균을 나타낸다. 런길이의 분포는 일반적으로 산포가 크고 오른쪽으로 꼬리가 긴 분포를 따르기 때문에 대푯값으로 ARL 이외에 다른 측도 (런길이의 중앙값, 백분위수 등)를 사용하는 경우도 있지만, 아직까지 대부분의 연구에서 ARL을 사용하여 성능을 비교하고 있다. ARL은 오경보율과 관련된 관리상태의 ARL인 ARL_0 와 탐지력을 나타내는 이상상태의 ARL인 ARL_1 으로 구분한다. ARL_0 는 임계치를 크게 설정해서 큰 값을 갖는 것이 좋고, ARL_1 은 임계치를 작게 설정해서 작은 값을 갖는 것이 좋다. 따라서 일반적으로 모니터링 절차의 성능은 주어진 ARL_0 를 만족하도록 절차의 모수인 임계치를 설정한 후 ARL_1 을 계산하여 이 값으로 평가한다. 즉, 주어진 ARL_0 를 만족하면서 각 변화에 대해 작은 ARL_1 값을 갖는 절차의 성능이 좋은 것으로 평가된다 (Montgomery, 2020). 이 모의실험에서는 $ARL_0 = 370.4$ 를 만족하도록 임계치를 설정하고, 10,000번의 시행을 반복하여 ARL을 계산하였다.

생산 공정에서 공정은 초기에 관리상태를 유지하다가 일정 시간이 지난 후 이상상태로 변화하는 경우가 일반적이다. 따라서 $t^* = 71$ 로 설정하여 시점 70까지는 관리상태의 데이터, 그리고 시점 71부터 이상상태의 데이터를 생성하였다. 앞에서 언급한 바와 같이 학습 데이터의 윈도우 크기는 12를 사용하였는데, 이는 Ji와 Lee (2023)에서 사용한 24보다 작은 값이다. 윈도우 크기가 24인 경우 학습에 시간이 오래 소요되고 과적합(overfitting)이 자주 발생하여 이 논문에서는 12를 사용하였다. 따라서 시점 12부터 공정의 이상상태 여부를 판단할 수 있고, 시점 71의 공정 데이터가 최초로 포함되는 60 번째 윈도우부터 런길이를 측정하였다. 60 번째 윈도우 이전에 이상상태의 신호가 발생한 경우는 오경보이므로 모의실험의 반복에서 제외시켰다. 이때 각 모형에 대한 데이터는 Python의 Numpy 모듈을 사용한 함수를 따로 정의하여 생성하였고, 데이터 학습 및 모의실험은 Python과 PyTorch 모듈을 사용하여 수행하였다.

Table 2: ρ_1 of the autocorrelated models

AR(1)	ϕ
AR(2)	$\frac{\phi_1}{1 - \phi_2}$
ARMA(1,1)	$\frac{(\phi - \theta)(1 - \phi\theta)}{1 + \theta^2 - 2\phi\theta}$

3.3. 모의실험의 결과

이 논문에서 제안한 LSTM Autoencoder의 절차 (잠재 벡터를 이용한 절차)의 성능을 평가하기 위해 LSTM Autoencoder의 복원 오차를 이용한 절차, Ji와 Lee (2023)가 제안한 RNN에 기초한 분류 모니터링 절차, 그리고 전통적으로 사용하는 잔차 관리도 절차와 비교하였다.

LSTM Autoencoder의 복원 오차를 이용한 절차는 encoder와 decoder를 사용하여 압축과 복원을 한 후 복원 오차를 계산하고, 이 값이 주어진 임계치보다 큰 경우 이상상태라고 판단하는 것이다. 이때 복원 오차는 입력 윈도우와 복원 윈도우 간의 평균제곱오차를 사용하여 계산하였다.

Ji와 Lee (2023)가 제안한 RNN 분류 모니터링 절차에서는 관리상태와 이상상태의 데이터로 구성된 크기가 24인 입력 데이터 표본과 크기가 1인 출력 데이터 표본에 대해 RNN을 사용하여 모형을 학습시켰다. 이때 출력 데이터는 관리상태의 입력 데이터 표본은 0, 그리고 이상상태의 입력 데이터 표본은 1로 부여하였다. 모형을 학습시킨 후, 이 모형을 사용하여 새로 관측된 공정 데이터 표본이 관리상태의 표본인지 이상상태의 표본인지를 분류하는 절차이다.

1장에서 설명한 바와 같이, 잔차 관리도는 관측된 데이터를 시계열 모형에 적합시켜 계산된 잔차에 대해 관리도를 적용하는 것이다. 이 관리도는 미리 설정한 오경보율 또는 ARL_0 를 만족하도록 관리상한(upper control limit; UCL)과 관리하한(lower control limit; LCL)을 설정하고, 이 관리한계를 벗어나면 이상상태로 판단하게 된다. 잔차 관리도의 관리한계 UCL과 LCL은 식 (3.4)와 같은데, 여기서 \bar{e} 와 \overline{MR} 는 각각 관리상태에서 계산된 잔차들의 평균과 이동범위(moving window)의 평균이고, L 은 관리한계의 폭을 나타내는 계수이다.

$$UCL = \bar{e} + L \frac{\overline{MR}}{1.128}, \quad LCL = \bar{e} - L \frac{\overline{MR}}{1.128}. \quad (3.4)$$

이 논문에서는 공정에서 발생하는 이상상태의 종류로서 공정 수준의 변화 (식 (3.2)에서 $\delta > 0$, 식 (3.3)에서 $\gamma = 1$), 공정 수준과 분산의 동시 변화 ($\delta > 0$, $\gamma > 1$), 그리고 공정 분산의 변화 ($\delta = 0$, $\gamma > 1$) 등의 총 3가지 경우를 고려하였다. 시계열 모형에서 ϕ 와 θ 들의 변화에 따른 특성을 살펴보기 위해 1차 자기상관 ρ_1 을 Table 2에 제시했다.

각 절차에 대해 ARL_0 는 약 370.4가 되도록 절차의 모수를 설정했고, 고려한 시계열 모형에서 변화에 대한 ARL_1 을 모의실험을 통해 계산하여 이를 Tables 3–5에 각각 나타내었다. 각 Table에서 ‘LSTM-AE latent vector’는 LSTM Autoencoder의 잠재 벡터를 이용한 절차, ‘LSTM-AE Recons.’는 LSTM Autoencoder의 복원 오차를 이용한 절차, ‘RNN classification’은 RNN에 기초한 분류 모니터링 절차, 그리고 ‘residual’은 잔차 관리도 절차를 나타낸다. 비교를 용이하게 하기 위해 Table에서 각 경우에 대해 ARL 이 가장 작은 값을 볼드체로 표시하였다.

Tables 3–5의 결과를 살펴보면, 각 이상상태의 유형에 대해 다음과 같은 사실을 알 수 있다. 먼저 Table 3의 AR(1) 모형에서 $\phi = 0.25$ 일 때 공정의 수준만 변화한 경우 ($\delta > 0$, $\gamma = 1$)에는 잠재 벡터를 이용한 절차의 성능이 좋았지만, 수준과 분산이 모두 변화한 경우 ($\delta \geq 0$, $\gamma > 1$)에는 RNN 분류, 분산만 변화한 경우 ($\delta = 0$, $\gamma > 1$)에는 복원 오차를 이용한 절차의 성능이 좋게 나타났다. $\phi = 0.5$ 의 경우에는 RNN 분류와 복원 오차를 이용한 절차의 성능이 좋았으며, 자기상관이 커질수록 (ϕ 가 커질수록) 잔차 관리도 절차의 성능이 좋은 것을

Table 3: ARL values of AR(1) process

AR(1) process			LSTM-AE latent vector		LSTM-AE Recons.		RNN classification		Residual	
ϕ	δ	γ	ARL	Threshold	ARL	Threshold	ARL	Threshold	ARL	L
0.25 $\sigma_x = 1.0328$	0	1	374.90	0.4555	378.77	2.18	376.31	0.926	376.35	2.964
	0.5	1	35.82		288.55		39.60		178.90	
	1	1	10.08		194.41		12.20		59.59	
	2	1	4.48		16.13		4.55		9.36	
	3	1	3.31		2.15		2.80		2.44	
	0.5	1.5	25.49		13.25		18.40		18.20	
	1	1.5	11.34		11.53		9.85		12.13	
	2	1.5	4.97		5.29		4.52		4.82	
	3	1.5	3.29		2.17		2.80		2.45	
	0	1.5	60.16		13.71		45.22		22.62	
0	2	32.59		5.55		21.09		7.63		
0	3	19.05		2.37		10.54		3.18		
0.5 $\sigma_x = 1.1547$	0	1	375.00	0.341	372.78	2.81	374.40	0.9485	372.80	2.997
	0.5	1	72.12		231.03		57.37		237.24	
	1	1	19.31		107.26		17.85		106.30	
	2	1	5.64		16.98		5.83		19.66	
	3	1	3.91		3.43		3.71		3.46	
	0.5	1.5	35.79		16.99		25.25		19.73	
	1	1.5	17.54		14.04		13.74		14.91	
	2	1.5	6.60		7.10		5.91		6.52	
	3	1.5	3.93		3.36		3.42		3.63	
	0	1.5	54.90		18.36		57.89		22.44	
0	2	27.46		7.29		28.79		7.50		
0	3	15.22		3.09		14.41		3.23		
0.75 $\sigma_x = 1.5119$	0	1	374.16	0.5296	374.31	4.48	379.60	0.9797	371.23	2.98
	0.5	1	134.38		305.19		89.53		348.37	
	1	1	41.37		258.22		31.70		245.30	
	2	1	10.81		55.90		8.66		59.10	
	3	1	7.05		5.79		4.54		4.34	
	0.5	1.5	56.98		26.93		39.50		21.18	
	1	1.5	33.83		24.09		21.87		18.49	
	2	1.5	13.07		13.26		9.15		8.65	
	3	1.5	7.08		5.79		4.50		4.41	
	0	1.5	66.22		28.65		78.86		22.69	
0	2	36.43		10.81		41.99		7.53		
0	3	21.42		4.44		22.67		3.16		
0.95 $\sigma_x = 3.2026$	0	1	375.41	0.64	378.61	8.07	377.64	0.99931	376.94	2.96
	0.5	1	163.47		294.47		149.40		321.65	
	1	1	75.08		213.43		65.63		122.09	
	2	1	18.66		25.85		16.71		1.05	
	3	1	8.25		1.59		6.17		1.00	
	0.5	1.5	70.53		35.09		79.18		19.68	
	1	1.5	49.76		28.52		47.52		11.05	
	2	1.5	20.74		9.07		17.55		1.22	
	3	1.5	8.19		1.66		6.02		1.00	
	0	1.5	87.93		37.23		133.17		22.45	
0	2	48.57		15.70		83.00		7.73		
0	3	28.21		6.58		48.73		3.26		

알 수 있다. 전반적으로 공정의 수준만 변화한 경우에는 통계적 학습에 기초한 잠재 벡터를 이용한 절차와 RNN 분류 절차의 성능이 좋았고, 분산이 변화하는 경우에는 복원 오차를 이용한 절차와 간차 관리도 절차의 성능이 좋게 나타났다.

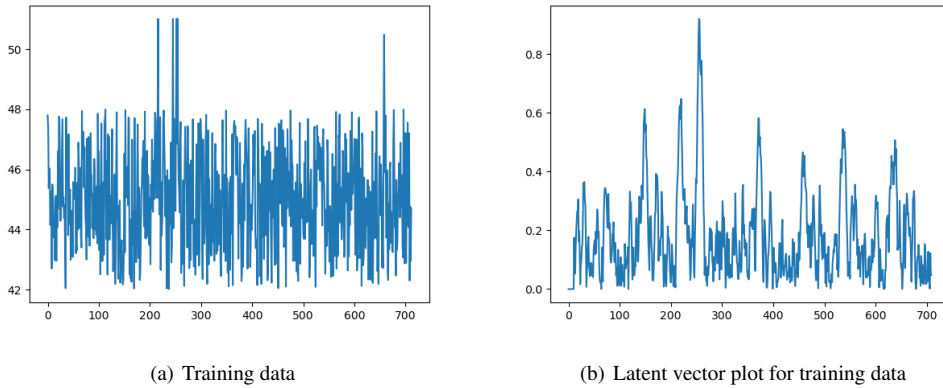
Table 4의 AR(2) 모형에서 $\rho_1 = 0$ 일 때 수준이 변화한 경우에는 대체적으로 잠재 벡터를 이용한 절차의 성능이 좋았지만, 분산만 변화한 경우에는 복원 오차를 이용한 절차의 성능이 가장 좋았다. ρ_1 이 커질수록 수준의 변화에 대해 잠재 벡터를 이용한 절차와 RNN 분류 절차가 다른 절차에 비해 성능이 좋았고, 분산의

Table 4: ARL values of AR(2) process

AR(2) process		LSTM-AE latent vector		LSTM-AE Recons.		RNN classification		Residual		
ϕ_1/ϕ_2	δ	γ	ARL	Threshold	ARL	Threshold	ARL	Threshold	ARL	L
0 / 0.1 $\sigma_x = 1.005$ $\rho_1 = 0$	0	1	378.64	0.3676	370.60	2.13	374.54	0.953	370.33	2.97
	0.5	1	30.78		256.97		30.77		229.36	
	1	1	7.78		107.89		9.62		83.80	
	2	1	2.76		5.86		4.06		12.36	
	3	1	1.86		1.47		2.57		2.61	
	0.5	1.5	15.37		11.56		16.40		19.25	
	1	1.5	6.98		8.90		8.58		13.37	
	2	1.5	2.95		3.51		4.02		5.13	
	3	1.5	1.84		1.45		2.57		2.64	
	0	1.5	39.29		12.52		42.41		22.24	
	0	2	17.68		5.04		19.82		7.60	
	0	3	8.18		2.19		10.08		3.14	
0.25 / 0.2 $\sigma_x = 1.0744$ $\rho_1 = 0.3125$	0	1	372.20	0.613	370.09	2.277	377.76	0.97357	372.32	3.01
	0.5	1	72.74		297.12		53.30		323.57	
	1	1	18.71		189.06		16.51		221.08	
	2	1	5.89		12.81		5.66		69.80	
	3	1	3.89		2.03		3.46		13.25	
	0.5	1.5	35.26		14.64		25.73		21.54	
	1	1.5	16.74		12.09		13.40		18.04	
	2	1.5	6.53		5.36		5.94		9.88	
	3	1.5	3.93		1.97		3.48		13.07	
	0	1.5	60.84		15.15		59.35		22.48	
	0	2	31.83		5.87		28.57		7.77	
	0	3	17.70		2.47		14.46		3.27	
0.4 / 0.3 $\sigma_x = 1.2773$ $\rho_1 = 0.5714$	0	1	377.79	0.4104	371.99	2.94	370.63	0.9898	377.70	3.02
	0.5	1	83.82		297.55		83.91		318.24	
	1	1	28.36		245.30		28.92		203.31	
	2	1	8.30		41.54		8.31		53.08	
	3	1	6.05		3.35		4.55		6.54	
	0.5	1.5	44.66		21.39		38.50		20.75	
	1	1.5	24.93		17.99		21.59		17.54	
	2	1.5	10.20		9.14		8.70		8.83	
	3	1.5	6.02		3.46		4.58		6.49	
	0	1.5	73.43		21.60		78.23		22.61	
	0	2	39.59		8.18		39.87		7.53	
	0	3	23.88		3.27		20.56		3.17	
0.6 / 0.2 $\sigma_x = 1.5403$ $\rho_1 = 0.7501$	0	1	375.02	0.5627	374.87	3.98	379.54	0.9962	371.96	3.05
	0.5	1	125.39		307.56		101.81		358.96	
	1	1	44.80		252.46		37.98		302.61	
	2	1	10.64		45.98		10.54		115.13	
	3	1	5.68		4.02		5.30		9.62	
	0.5	1.5	46.13		27.38		46.89		21.69	
	1	1.5	28.78		24.04		27.30		19.01	
	2	1.5	11.61		12.25		11.01		10.05	
	3	1.5	5.56		4.03		5.27		9.12	
	0	1.5	61.05		29.02		89.16		22.41	
	0	2	31.57		10.88		48.27		7.60	
	0	3	17.75		4.29		24.54		3.19	
0.8 / 0.1 $\sigma_x = 2.1938$ $\rho_1 = 0.8889$	0	1	377.23	0.457	370.25	5.9	375.40	0.9996	374.03	3.01
	0.5	1	118.89		295.34		124.01		352.43	
	1	1	50.15		195.99		51.96		255.45	
	2	1	13.41		29.33		14.26		19.69	
	3	1	6.75		2.23		6.15		1.03	
	0.5	1.5	59.60		31.97		62.61		20.25	
	1	1.5	37.28		27.14		37.58		16.16	
	2	1.5	15.14		12.00		15.21		4.38	
	3	1.5	6.71		2.31		6.23		1.03	
	0	1.5	92.81		33.39		113.65		22.01	
	0	2	59.73		13.73		63.27		7.56	
	0	3	45.45		5.48		32.12		3.14	

Table 5: ARL values of ARMA(1, 1) process

ARMA(1,1) process			LSTM-AE latent vector		LSTM-AE Recons.		RNN classification		Residual	
ϕ/θ	δ	γ	ARL	Threshold	ARL	Threshold	ARL	Threshold	ARL	L
0.2 / -0.2 $\sigma_x = 1$ $\rho_1 = 0.3714$	0	1	371.63	0.4174	370.69	2.428	373.17	0.9254	375.49	3.8
	0.5	1	45.00		280.17		38.95		151.47	
	1	1	11.83		190.94		13.11		43.25	
	2	1	3.79		17.18		4.82		7.26	
	3	1	2.34		2.12		3.80		2.56	
	0.5	1.5	23.53		14.56		19.43		23.25	
	1	1.5	11.06		12.25		10.43		13.92	
	2	1.5	4.21		5.44		4.71		5.29	
	3	1.5	2.38		2.10		2.78		2.57	
	0	1.5	46.44		15.12		48.29		29.7	
0	2	21.71		6.06		22.18		10.55		
0	3	11.14		2.56		10.45		4.31		
0.2 / -0.8 $\sigma_x = 1.0013$ $\rho_1 = 0.5918$	0	1	379.43	0.4273	372.62	4.54	379.44	0.9556	374.85	4.79
	0.5	1	49.26		326.28		46.29		234.06	
	1	1	15.22		259.04		15.33		94.32	
	2	1	5.22		21.65		5.46		18.18	
	3	1	3.92		2.93		3.13		4.89	
	0.5	1.5	30.92		19.50		22.62		36.37	
	1	1.5	14.82		17.09		12.23		24.09	
	2	1.5	6.31		7.67		5.42		9.33	
	3	1.5	3.90		3.02		3.10		4.95	
	0	1.5	74.45		20.40		54.93		42.93	
0	2	37.03		8.16		26.02		14.63		
0	3	19.10		3.59		12.51		5.51		
0.4 / -0.4 $\sigma_x = 1$ $\rho_1 = 0.627$	0	1	370.30	0.426	371.96	3.9	377.64	0.9541	376.18	3.34
	0.5	1	48.57		304.39		51.76		210.61	
	1	1	15.10		246.12		17.22		85.59	
	2	1	5.40		32.29		5.90		14.35	
	3	1	3.88		3.53		3.37		2.69	
	0.5	1.5	30.37		20.84		25.13		21.51	
	1	1.5	14.88		17.82		13.61		15.37	
	2	1.5	6.23		8.68		6.01		6.24	
	3	1.5	3.86		3.46		3.36		2.56	
	0	1.5	73.96		21.73		60.41		25.56	
0	2	36.11		8.41		28.90		8.64		
0	3	19.02		3.63		13.74		3.56		
0.8 / -0.2 $\sigma_x = 1.4142$ $\rho_1 = 0.8529$	0	1	372.79	0.444	373.93	7.3	375.67	0.9896	371.13	3.049
	0.5	1	87.02		247.03		92.98		322.69	
	1	1	33.94		113.49		34.23		203.06	
	2	1	10.39		16.46		10.39		21.37	
	3	1	6.83		3.07		4.96		1.09	
	0.5	1.5	46.61		29.22		46.20		20.85	
	1	1.5	26.92		23.53		25.24		16.01	
	2	1.5	11.54		10.37		10.59		5.44	
	3	1.5	6.87		3.20		5.02		1.11	
	0	1.5	90.14		32.90		94.66		22.18	
0	2	55.47		13.37		51.23		7.77		
0	3	37.43		5.60		26.86		3.22		
0.8 / -0.8 $\sigma_x = 1$ $\rho_1 = 0.8986$	0	1	373.06	0.57	372.08	15.13	370.12	0.99815	376.91	4.5
	0.5	1	87.98		318.49		94.42		350.83	
	1	1	33.97		266.86		35.67		252.80	
	2	1	9.52		51.97		10.51		39.75	
	3	1	5.40		6.08		5.20		1.48	
	0.5	1.5	44.36		34.09		47.16		39.13	
	1	1.5	25.73		30.15		27.12		31.88	
	2	1.5	10.79		16.04		11.19		9.40	
	3	1.5	5.31		5.93		5.17		1.47	
	0	1.5	87.67		36.18		95.17		40.70	
0	2	53.68		14.91		51.87		14.00		
0	3	32.31		6.60		28.73		5.21		

Figure 6: *Electrolytic degreasing data.*

변화, 특히 분산만 변화한 경우에는 복원 오차를 이용한 절차와 잔차 관리도 절차에 비해 성능이 좋지 않게 나타났다. 전반적으로 모든 경우에서 잠재 벡터를 이용한 절차와 RNN 분류 절차의 성능은 크게 차이가 없음을 알 수 있다.

Table 5의 ARMA(1, 1) 모형의 경우에도 유사한 결과를 확인할 수 있었다. 수준이 변화한 경우에는 잠재 벡터를 이용한 절차의 성능이 좋은 경우가 많았지만, RNN 분류 절차의 성능과 크게 차이가 없음을 알 수 있었다. 또한 분산만 변화한 경우에는 여전히 복원 오차를 이용한 절차와 잔차 관리도의 성능이 가장 좋게 나타났다.

모의실험에서 LSTM Autoencoder를 이용한 2가지 모니터링 절차의 성능은 시계열 모형에 상관없이 비슷한 경향을 보였다. 공정의 수준 변화가 작은 경우 ($\delta < 3$) 잠재 벡터를 이용한 절차의 성능이 우수하였고, 공정의 변화가 크거나 ($\delta = 3$) 공정의 분산에 변화가 발생한 경우 복원 오차를 이용한 절차의 성능이 더 좋게 나타났다. 그러나 복원 오차를 이용한 절차는 3.1절에서도 언급한 바와 같이 공정 수준의 변화가 작은 경우 성능이 많이 떨어진다는 사실에 유념해야 한다.

모의실험 결과, 이상상태의 데이터가 필요한 지도 학습 방법인 RNN 분류 절차와 이상상태의 데이터가 따로 필요하지 않은 비지도 학습 방법인 LSTM Autoencoder의 잠재 벡터를 이용한 절차의 성능이 유사했으며, 복원 오차를 이용한 절차와 전통적인 잔차 관리도 절차의 성능이 유사한 것을 확인할 수 있었다.

4. 실제 공정 데이터에 대한 적용

이 장에서는 제안된 모니터링 절차를 실제 공정 데이터에 적용하고 그 성능을 평가하고자 한다. 실제 공정 데이터는 2가지로서, 4.1절의 전해탈지 공정 데이터와 4.2절의 건조구동장치 공정 데이터이다.

4.1. 예제 1 : 전해탈지 공정 데이터

제안한 모니터링 절차를 인공지능 중소벤처 제조 플랫폼에서 제공하는 전해탈지 공정 데이터에 적용하고자 한다 (출처: 중소벤처기업부, Korea AI Manufacturing Platform (KAMP), 진단(전해탈지) AI 데이터셋, KAIST ((주)에이비에이치, (주)임픽스), 2021.12.27, <https://mdata.kamp-ai.kr>). 해당 데이터는 이노징크(innozinc) 전기도금 분야의 데이터로서 제품 생산 시 설비에 설치된 센서를 통해 수집된 설비 데이터이고, 2021년 9월 6일부터 2021년 10월 27일까지 5초 간격으로 수집된 데이터이다. 3장에서 실시한 모의실험과 유사한 수준의

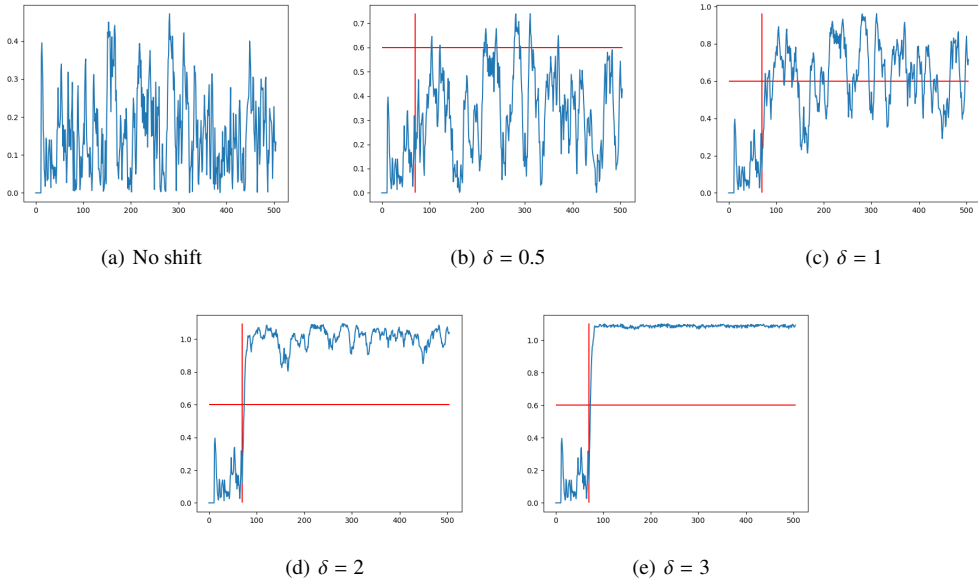


Figure 7: Results for monitoring electrolytic degreasing data.

학습을 위해 수집된 전체 데이터가 아닌 10월 21일 하루에 관측된 길이 1576의 전처리 설비 내 공정 온도 데이터만 사용하고자 한다. 전해탈지에서 온도가 너무 낮으면 오염물 제거 및 표면 활성화가 약하고 너무 높으면 제품에 변색 등의 손상이 발생할 수 있기 때문에, 온도는 공정 품질에 영향을 미치는 중요한 변수 중 하나이다. 사용한 데이터는 자기상관이 존재하고 정상성(stationarity)을 만족하는 시계열 데이터로 확인되었기 때문에, 이 논문에서 제안한 모니터링 절차의 적용이 가능한 데이터라고 판단하였다.

모형의 입력 데이터는 제안된 절차와 동일하게 크기가 12인 시계열 윈도우를 만들어 700개를 학습 데이터, 300개를 검증 데이터, 나머지 데이터를 평가 데이터로 나누어 학습을 진행하였다. Figure 6은 온도 데이터와 출력된 잠재 벡터의 그래프로서 대부분 안정적인 값을 갖지만 시점 216, 245, 252, 254, 658에서 큰 값을 갖는 것으로 나타났는데, 해당 시점의 데이터를 제거하지 않고 학습을 진행하되 임계치의 결정에서는 배제하였다.

학습 데이터에서 출력된 잠재 벡터값들을 기준으로 하여 임계치는 0.6으로 설정하였고, 평가 데이터에서 시점 71에 공정 수준의 변화를 인위적으로 발생시켜 이상상태를 탐지할 수 있는지 확인하였다. δ 의 크기에 따른 탐지 여부를 Figure 7에 나타내었다. Figure 7을 살펴보면, $\delta = 0.5$ 의 경우 공정 변화가 발생한 후 34 번째, $\delta = 1$ 의 경우 7 번째, $\delta = 2$ 의 경우 5 번째, $\delta = 3$ 의 경우 4 번째 시점에서 이상상태를 탐지했으며, 오경보는 발생하지 않은 것으로 나타났다.

4.2. 예제 2 : 건조구동장치 공정 데이터

이 절에서는 제안된 모니터링 절차를 인공지능 중소벤처 제조 플랫폼에서 제공하는 건조구동장치 데이터에 적용하고자 한다 (출처: 중소벤처기업부, Korea AI Manufacturing Platform (KAMP), 예지보전(건조구동장치) AI 데이터셋, KAIST ((주)아쿠라소프트, (주)임팩스, 한양대학교 산학협력단), 2020.12.14, <https://mdata.kamp-ai.kr>). 해당 데이터는 분무건조공법을 이용한 분말유크림 제조 공정의 장비인 건조구동장치의 데이터로서 해당 장치의 부품 중 하나인 모터 베어링의 진동 데이터이다. 데이터는 건조구동장치에 부착된 무선 진동센서를 통해 수집되며, 수집 기간은 2021년 11월 7일부터 2021년 11월 20일까지 약 10분 간격으로 수집된 데이터이

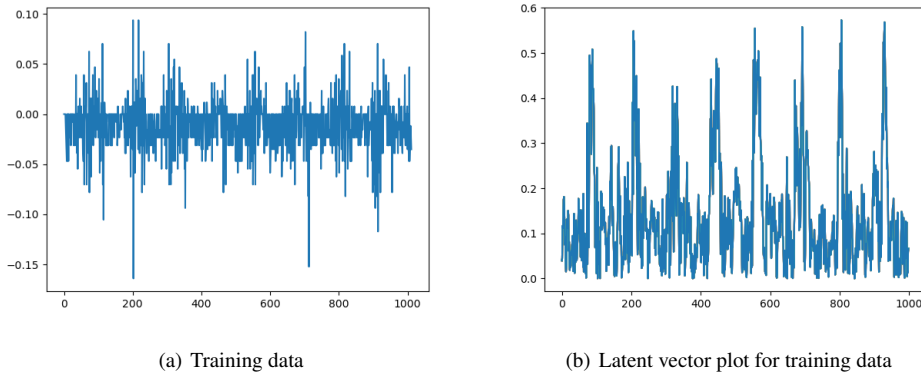


Figure 8: Drier vibration data.

다. 4.1절과 같이 3장에서 실시한 모의실험과 유사한 수준의 학습을 위해 수집된 전체 데이터가 아닌 학습 데이터 1,000개, 검증 데이터 500개, 그리고 평가 데이터 500개의 데이터만 사용하였다. 전해탈지 공정 데이터와 마찬가지로, 건조구동장치 공정 데이터는 자기상관이 존재하는 정상 시계열 데이터임을 확인하였다. Figure 8은 진동 데이터와 출력된 잠재 벡터의 그래프를 나타낸다.

학습 데이터에서 출력된 잠재 벡터값들을 기준으로 하여 임계치는 0.58으로 설정하였고, 평가 데이터에서 시점 71에 공정 수준의 변화를 인위적으로 발생시켜 이상상태를 탐지할 수 있는지 확인하였다. δ 의 크기에 따른 탐지 여부를 Figure 9에 나타내었다. Figure 9를 살펴보면, $\delta = 0.5$ 의 경우 공정 변화가 발생한 후 86 번째, $\delta = 1$ 의 경우 11 번째, $\delta = 2$ 의 경우 4 번째, $\delta = 3$ 의 경우 3 번째 시점에서 이상상태를 탐지했으며, 오경보는 발생하지 않은 것으로 나타났다.

5. 결론

이 논문에서는 LSTM Autoencoder를 이용하여 자기상관 공정의 특성을 학습하고 학습된 모형의 잠재 벡터에 기초한 모니터링 절차를 제안하였다. 또한 모의실험을 통해 제안된 절차의 성능을 LSTM Autoencoder의 복원 오차를 이용한 절차, RNN에 기초한 분류 모니터링 절차, 그리고 전통적인 잔차 관리도 절차와 비교하였고, 전해탈지 공정의 온도 데이터와 건조구동장치 공정의 진동 데이터에 적용하여 실제 공정 데이터에 대한 성능을 확인하였다.

모의실험을 통해 얻은 결론은 다음과 같다. 공정의 수준 변화에 대해 제안된 비지도 학습 방법인 LSTM Autoencoder와 지도 학습 방법인 RNN 분류 모니터링 절차의 성능은 유사했으며, 1차 자기상관인 ρ_1 값이 작은 경우 제안된 LSTM Autoencoder의 성능이 더 좋은 것으로 나타났다. 공정의 분산 변화에 대해서는 복원 오차를 이용한 절차와 잔차 관리도 절차의 성능이 더 좋은 것으로 나타났다. 그러나 복원 오차를 이용한 절차는 공정 수준의 작은 변화에 성능이 많이 떨어지고, 잔차 관리도 절차는 시계열 모형에 적합시켜야 하기 때문에 적합된 모형이 정확하지 않을 경우에 이상상태의 탐지 성능이 나빠진다는 단점이 있다. RNN 분류 모니터링 절차의 경우 LSTM Autoencoder를 사용한 절차와 동일하게 시계열 모형의 적합이 불필요하지만 학습을 위해 이상상태인 데이터가 필수적으로 필요하다는 단점이 있다. 따라서 관리상태의 데이터만으로 학습과 공정의 모니터링이 가능하고 이상상태의 탐지 성능 또한 나쁘지 않다는 점에서 잠재 벡터에 기초한 LSTM Autoencoder 모니터링 절차는 충분한 장점을 가지고 있다고 판단된다.

향후 LSTM Autoencoder를 다변량 자기상관 공정 데이터에 적용하는 절차를 제안하고 그 특성과 성능을

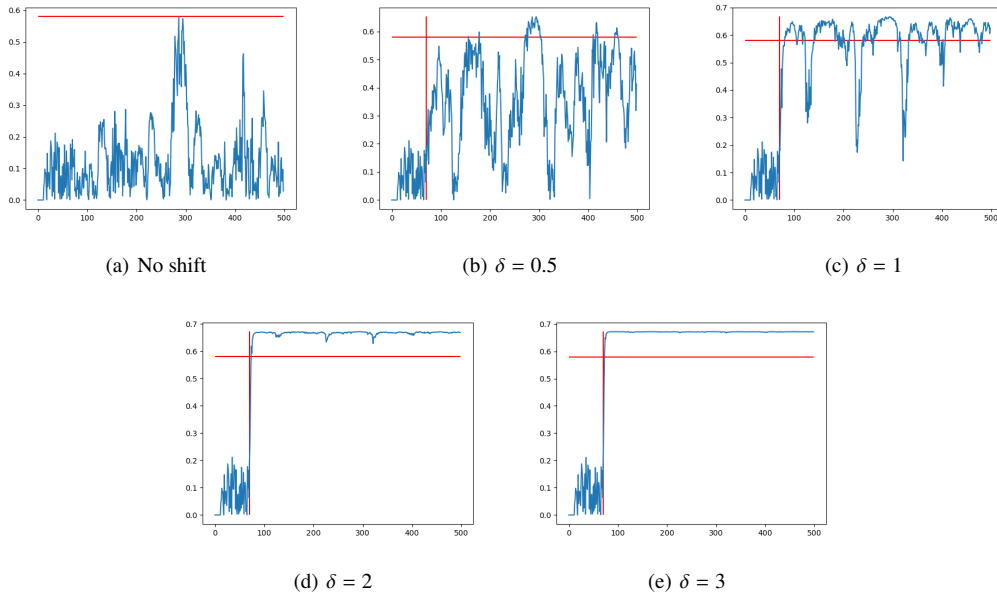


Figure 9: Results for monitoring drier vibration data.

평가하는 연구를 진행하고자 한다.

References

- Chen S and Yu J (2019). Deep recurrent neural network-based residual control chart for autocorrelated processes, *Quality and Reliability Engineering International*, **35**, 2687–2708.
- Ho ES and Chang SI (1999). An integrated neural network approach for simultaneous monitoring of process mean and variance shifts a comparative study, *International Journal of Production Research*, **37**, 1881–1901.
- Hochreiter S and Schmidhuber J (1997). Long short-term memory, *Neural Computation*, **9**, 1735–1780.
- Homayouni H, Ghosh S, Ray I, Gondalia S, Duggan J, and Kahn MG (2020). An autocorrelation-based LSTM-Autoencoder for anomaly detection on time-series data. In *Proceedings of 2020 IEEE International Conference on Big Data*, Atlanta, GA, 5068–5077.
- Hwang HB (2004). Detecting process mean shift in the presence of autocorrelation: A neural-network based monitoring scheme, *International Journal of Production Research*, **42**, 573–595.
- Hwang HB and Wang Y (2010). Shift detection and source identification in multivariate autocorrelated processes, *International Journal of Production Research*, **48**, 835–859.
- Ji P and Lee J (2023). Performance comparison of procedures for monitoring autocorrelated processes based on classification and forecasting, *Journal of the Korean Data & Information Science Society*, **34**, 775–789.
- Kramer MA (1991). Nonlinear principal component analysis using autoassociative neural networks, *AIChE Journal*, **37**, 233–243.
- Kramer MA (1992). Autoassociative neural networks, *Computers & Chemical Engineering*, **16**, 313–328.
- Lee S and Lee J (2022). Monitoring procedure of autocorrelated processes using the deep learning-based LSTM

- model, *Journal of the Korean Data & Information Science Society*, **33**, 835–842.
- Lu CW and Reynolds MR JR (1999). EWMA control charts for monitoring the mean of autocorrelated processes, *Journal of Quality Technology*, **31**, 166–188.
- Maged A, Lui CF, Haridy S, and Xie M (2023). Variational AutoEncoders-LSTM based fault detection of time-dependent high dimensional processes, *International Journal of Production Research*, **62**, 1–16.
- Montgomery DC (2020). *Introduction to Statistical Quality Control* (8th ed), John Wiley & Sons, Inc., Hoboken, NJ.
- Prajapati DR and Singh S (2015). Application of ANN to monitor the correlated process using higher sample size, *International Journal of Performability Engineering*, **11**, 395–404.
- Srivastava N, Mansimov E, and Salakhutdinov R (2015). Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 843–852.
- Wei Y, Jang-Jaccard J, Xu W, Sabrina F, Camtepe S, and Boulic M (2022). LSTM-Autoencoder based anomaly detection for indoor air quality time series data, Available from: arXiv:2204.06701 [cs.LG]
- White AK and Safi SK (2016). The efficiency of artificial neural networks for forecasting in the presence of autocorrelated disturbances, *International Journal of Statistics and Probability*, **5**, 51–58.
- Yuanyai C and Abrahams R (2011). Statistical process control with autocorrelated data using neural networks. In *Proceedings of 2011 IEEE International Conference on Quality and Reliability*, Bangkok, Thailand, 283–287.

Received December 23, 2023; Revised February 6, 2024; Accepted February 28, 2024

LSTM Autoencoder를 이용한 자기상관 공정의 모니터링 절차

지평진^a, 이재현^{1,a}

^a중앙대학교 응용통계학과

요약

자기상관 공정에서 이상상태를 빠르게 탐지하는 절차에 대해 많은 연구가 진행되어 왔다. 가장 전통적인 절차는 관측된 데이터에 대해 적합한 시계열 모형에서 계산된 잔차를 이용하는 잔차 관리도이다. 그러나 최근에는 통계적 학습 방법을 이용하여 자기상관 공정을 모니터링하는 절차가 많이 제안되었다. 이 논문에서는 딥러닝에 기반한 비지도 학습 방법인 LSTM Autoencoder의 잠재 벡터를 이용한 모니터링 절차를 제안하고, 이를 모의실험을 통해 LSTM Autoencoder의 복원 오차를 이용한 절차, RNN 분류 모니터링 절차, 그리고 잔차 관리도 절차의 성능과 비교하였다. 모의실험 결과, 제안된 절차와 RNN 분류 모니터링 절차의 성능은 유사하지만, 제안된 절차는 학습에 이상상태의 데이터가 필요하지 않기 때문에 이상상태의 데이터를 충분히 확보할 수 없는 공정에 유용하게 적용할 수 있다는 장점이 있다.

주요용어: 자기상관 공정, 잠재 벡터, LSTM Autoencoder, 비지도 학습
