이학석사학위논문

Detecting Structural Change Point in Autoregressive Time Series via Transformer

Transformer을 이용한 자기회귀 시계열의 구조적 변화점 탐지

2023 년 8 월

서울대학교 대학원

통계학과

조 항 범

Detecting Structural Change Point in Autoregressive Time Series via Transformer

Transformer을 이용한 자기회귀 시계열의 구조적 변화점 탐지

지도교수 이 상 열

이 논문을 이학석사 학위논문으로 제출함

2023 년 4 월

서울대학교 대학원

통계학과

조 항 범

조항범의 이학석사 학위논문을 인준함

2023 년 6 월

위 원 장	오희석	(인)
부위원장	이상열	(인)
위 원	이권상	(인)

Abstract

In this paper, we discuss a method for detecting structural change point in autoregressive time series using transformer based deep learning model. Detecting structural changes can be achieved using the LSCUSUM test, which is one of the most popular methods for change point detection. A crucial aspect of constructing the LSCUSUM test is to adequately estimate the residuals, and choosing an appropriate model is of paramount importance. Given that many time series exhibit nonlinear characteristics, it becomes imperative to employ deep learning methods for capturing and effectively modeling these nonlinearities. Therefore, in this context, we utilize a transformer-based deep learning model that leverages the powerful self-attention mechanism. In the process, we compute empirical size and power about our method and apply to two real datasets.

Keywords: Structural break, Deep learning, Tranformer, LSCUSUM, Autoregressive

Student Number: 2021-26067

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Change point detection is widely studied in the field of time series analysis because ignoring change points can lead to significant errors in modeling the underlying patterns of the data. Many domains that require time series analysis often involve situations where structural change can occur. For example, changes in government policies or sudden social events can introduce change points in economic time series data.

The cumulative sum (CUSUM) test, introduced by Page (1955), is widely used as a convenient method for change point detection. However, the conventional estimate-based CUSUM test proposed by Lee et al. (2003) can suffer from size distortions and low power in certain cases. To address this issue, Oh and Lee (2019) and Oh and Lee (2018) suggested a score vector-based CUSUM test and a modified residual-based CUSUM test. Furthermore, Lee (2020) proposed the location and scale-based CUSUM (LSCUSUM) test, which demonstrated improved stability adn power compared to previous estimate-, residual-, and score vector-based CUSUM tests through simulation studies. This test is advantageous because it can detect changes in both the location and scale of the data and only requires observations and residuals for analysis.

Accurate computation of residuals is crucial for the LSCUSUM test, as it relies on test statistics based on residuals. This requires fitting an appropriate model, especially for time series data with nonlinear autoregressive structures. Therefore, utilizing machine learning and deep learning models becomes essential to capture such complexities. Lee et al. (2020) proposed a hybrid change point detection method using SVR (Support Vector Regression) combined with the CUSUM method. Similarly, Ri et al. (2023) introduced an NNR (Neural Network Regression) approach for detecting structural change points in ARMA models using LSCUSUM methods.

Various methods are being used in analyzing data with sequential structures, such as time series, using deep learning techniques. Initially, RNN-based methodologies gained prominence as they were well-suited for capturing the relationship between past and current data in time series analysis. However, RNNs faced challenges such as the vanishing gradient problem. To partially address this, models such as LSTM by Hochreiter and Schmidhuber (1997) and GRU by Cho et al. (2014) were introduced, which consider long-term dependencies in time series.

In tasks involving seq2seq models like machine translation, which shares similarities with time series analysis, many issues are resolved using an encoder-decoder structure. The input sentence to be translated is read, encoded into a fixed-length vector, and then the decoder generates the output sentence recurrently.

However, the seq2seq models used in this approach store all input sequences into a single vector, leading to inevitable loss of information. To mitigate this, the Attention mechanism, introduced in Bahdanau et al. (2014), allows the

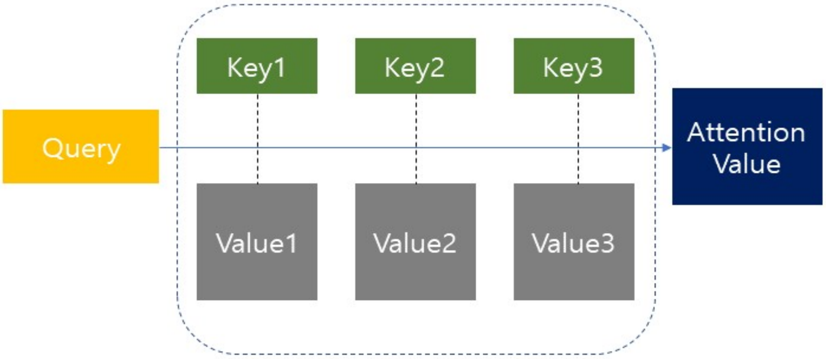

Figure 1.1 Description of Attention Function

decoder to refer back to the relevant parts of the source during each step of prediction. This enables the model to learn which parts of the encoding process are most relevant for accurate predictions. In time series analysis, this mechanism can be helpful in predicting correlations within irregular past patterns. To provide necessary explanation of Attention:

For a query (context) and key-value pairs (references), attention value is the weighted average of values, where each weight is proportional to the relevance between the query and the corresponding key. In time series structure, the past time points are used as references to predict the current time point by considering it as the query. Figure 2.1 illustrates the schematic representation of how Attention values are computed. The similarity between the query and the keys (1,2,3) is measured, and the weighted average of the values based on these similarities is used as the attention value. This attention operation plays a crucial role in the Transformer, which we introduce next.

Vaswani et al. (2017) proposed Transformer, an architecture that uses attention mechanism in encoder-decoder structure, not for the correction of RNN

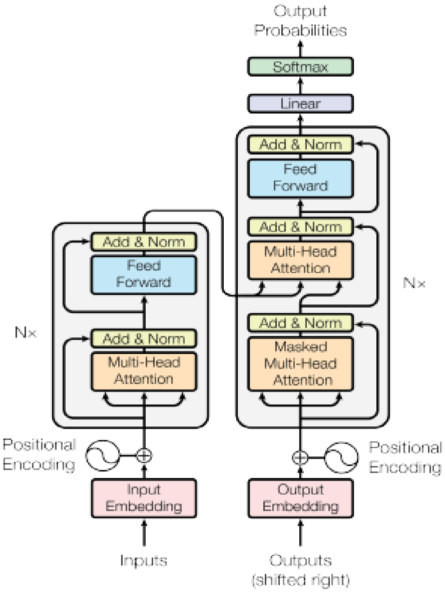

Figure 1.2 The Transformer - model architecture in Vaswani et al. (2017)

based encoder-decoder structure. Figure 1.2 describes Transformer architecture in Vaswani et al. (2017). The transformer architecture has shown remarkable performance in predicting data with sequential structures, particularly in tasks such as machine translation and computer vision. we aim to utilize the transformer to accurately compute the residuals to meet the requirements of the LSCUSUM test.

The remaining parts of this paper are as follows: In Chapter 2, an explanation of the transformer and LSCUSUM test is provided, along with a brief overview of the attention mechanism underlying the transformer. Chapter 3 conducts a simulation study, performing a grid search for tuning parameters and calculating the empirical size of the two models and power of various changes through Monte Carlo simulations. In Chapter 4, the methods are applied to two real datasets for illustration. Lastly, Chapter 5 concludes the paper with final remarks.

Chapter 2

Model Description

For time series $\{y_t\}$, we use a sliding window method to construct a training dataset. To align with the seq2seq model used in the transformer and to fix an input length to the model, we put input variable as $y_t, y_{t-1}, \cdots, y_{t-\text{iw}+1}$ where iw is a length of input window, out put variable as $y_{\text{ow}}, \cdots, y_{t+1}$ where ow is a length of output window. The functional relationship learned by Transformer model is below:

$$(y_{\text{ow}}, \cdots, \hat{y}_{t+1}) = f(y_t, y_{t-1}, \cdots, y_{t-\text{iw}+1})$$

where $\hat{y}_{t+k}$ is prediction for time $t+k$. To capture autoregressive structure for time series, we only use $\hat{y}_{t+1}$ in $(y_{\text{ow}}, \cdots, \hat{y}_{t+1})$ to predict.The data were normalized using the Min-Max normalization technique.

2.1 Transformer

The Figure 2.2 describes our Transformer based prediction structure. Mathematically,

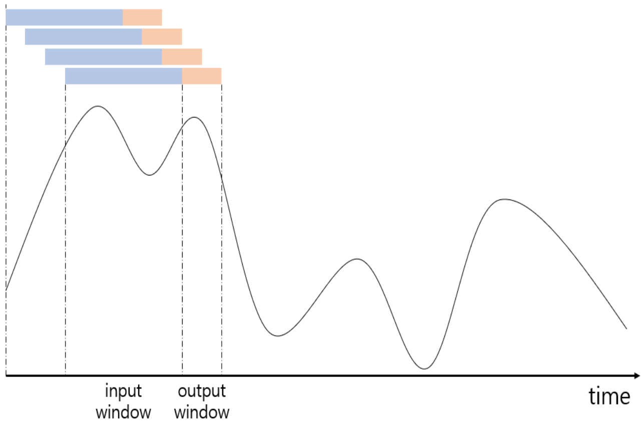

Figure 2.1 Using a sliding window to construct a dataset

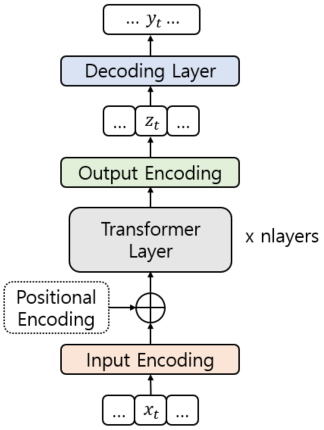

Figure 2.2 Description of the Transformer based prediction model

$$
\begin{aligned}
d_t &= W_2 \max(W_1 x_t + b_1, 0) + b_2 \\
q_t &= d_t + p_t \\
o_t &= \text{transformerEncoder}^n(q_t) \\
z_t &= W_4 \max(W_3 o_t + b_3, 0) + b_4 \\
\mathbf{y} &= W_6 \max(W_5 \mathbf{z} + b_5, 0) + b_6
\end{aligned}
$$

where $x_t \in \mathbf{R}$ be input time series at time t. $W_1 \in \mathbf{R}^{d/2\times 1},\ W_2 \in \mathbf{R}^{d\times d/2},\ W_3 \in \mathbf{R}^{d/2\times d},\ W_4 \in \mathbf{R}^{d\times d/2},\ W_5 \in \mathbf{R}^{(ow+iw)/2\times iw},\ W_6 \in \mathbf{R}^{ow\times (ow+iw)/2}$ are parameter matrices to learn, and $b_1 \in \mathbf{R}^{d/2},\ b_2 \in \mathbf{R}^{d},\ b_3 \in \mathbf{R}^{d/2},\ b_4 \in \mathbf{R}^{d},\ b_5 \in \mathbf{R}^{(ow+iw)/2},\ b_6 \in \mathbf{R}^{ow}$ are bias vectors to learn. $\mathbf{x} \in \mathbf{R}^{iw}$ is an input vector that cut out of time series using sliding window method. for time t, we embedded univariate time series to $\mathbf{R}^d$ using feed-forward network, called d_t. We choose a dimension of embedding space as a hyperparameter. And, we added positional encoding p_t, as described in Vaswani et al. (2017), to inject some information about the position in the sequence. TransformerEncoder is similar to an Encoder part of the transformer in Figure 1.2. After that, we applied another feed-forward network to tranform the enbedded time series back to its univariate representation. Finally, we used a feed-forward network to decode our input window-length time series into an output window-length time series.

2.2 Location and scale-based CUSUM test

In this section, we introduce a method of detecting change point in time series. In Oh and Lee (2019), LSCUSUM test uses the mean-zero stationary location-scale time series model as below:

$$y_t = g_t(\mu_0) + \sqrt{h_t(\theta_0)}\eta_t,\ t \in \mathbf{Z}$$

where $g_t(\mu) = g(y_{t-1}, y_{t-2}, \cdots ; \mu)$ and $h_t(\theta) = h(y_{t-1}, y_{t-2}, \cdots ; \theta)$ denote autoregressive model with unknown parameter $\theta = (\mu, \theta)^T$. η_t are iid random variables with mean zero and unit variance. In particular, we consider autoregressive homogeneous time series model, so we only consider $h_t = \sigma^2$. Thus, the model discussed in this paper is

$$y_t = g_t(\mu_0) + \sigma\eta_t, \ t \in \mathbf{Z}$$

To detecting change point based on observations $y_1, \cdots, y_n$, we are interesting in testing the null hypothesis:

$$H_0 : \theta \text{ remains the same for the whole series} \quad vs. \quad H_1 : \text{not } H_0$$

To conduct a test, we consider a LSCUSUM test in Lee (2020). $\tilde{g}_t(\hat{\mu}_n)$ is our estimator of $g_t(\mu_0)$ via transformer and putting $\tilde{\epsilon}_t = y - \tilde{g}_t(\hat{\mu}_n)$ as residual, we can use

$$\begin{aligned}
\hat{T}_n^{LS} = \max_{1 \le k \le n} \Bigg\{ & \frac{1}{n\hat{\tau}_{1,n}^2} \left| \sum_{t=1}^{k} (y_t - \tilde{\epsilon}_t)\tilde{\epsilon}_t - \left(\frac{k}{n}\right) \sum_{t=1}^{n} (y_t - \tilde{\epsilon}_t)\tilde{\epsilon}_t \right|^2 \\
& + \frac{1}{n\hat{\tau}_{2,n}^2} \left| \sum_{t=1}^{k} \tilde{\epsilon}_t^2 - \left(\frac{k}{n}\right) \sum_{t=1}^{n} \tilde{\epsilon}_t^2 \right|^2 \Bigg\} \\
\hat{T}_n^{max} = \max_{1 \le k \le n} \max \Bigg\{ & \frac{1}{n\hat{\tau}_{1,n}^2} \left| \sum_{t=1}^{k} (y_t - \tilde{\epsilon}_t)\tilde{\epsilon}_t - \left(\frac{k}{n}\right) \sum_{t=1}^{n} (y_t - \tilde{\epsilon}_t)\tilde{\epsilon}_t \right|^2 \\
& , \frac{1}{n\hat{\tau}_{2,n}^2} \left| \sum_{t=1}^{k} \tilde{\epsilon}_t^2 - \left(\frac{k}{n}\right) \sum_{t=1}^{n} \tilde{\epsilon}_t^2 \right| \Bigg\}
\end{aligned}$$

where

$$\begin{aligned}
\hat{\tau}_{1,n}^2 &= \frac{1}{n} \sum_{t=1}^{n} (y_t - \tilde{\epsilon}_t)^2 \tilde{\epsilon}_t^2 - \left(\frac{1}{n} \sum_{t=1}^{n} (y_t - \tilde{\epsilon}_t)\tilde{\epsilon}_t \right)^2, \\
\hat{\tau}_{2,n}^2 &= \frac{1}{n} \sum_{t=1}^{n} \tilde{\epsilon}_t^4 - \left(\frac{1}{n} \sum_{t=1}^{n} \tilde{\epsilon}_t^2 \right)^2.
\end{aligned}$$

In Lee (2020), $\hat{T}_n^{LS}$ and $\hat{T}_n^{max}$ converges to a function of Brownian bridges in distribution. So, we obtain critical values of test statistics above through Monte Carlo simulations. We rejected H_0 if $\hat{T}_n^{LS} > 2.4503$ or $\hat{T}_n^{max} > 1.4596$ at the level of 0.05.

Chapter 3

Simulation Study

3.1 Selecting Optimal Tuning Parameters

In this section, we select optimal tuning parameters of transformer model via grid search method. First, we generate length-1000 time series $\{y_t\}_{t=1}^{1000}$ following the AR(2) model. Among them, we selected different parameter values for the first 500 observations and the last 500 observations, allowing for the selection of transformer model tuning parameters that can fit adequately to a model with a changing structure. We selected the tuning parameter with the smallest L1 loss, and the following tuning parameters became the candidates for selection.

- dmodel : the number of expected features in the input.
- nhead : the number of heads in the multi-head-attention models.
- nlayers : the number of sub-encoder-layers in the encoder.
- input window : a period of time that is used as input to predict the future values.

- output window : a period of time for which the model generat predictions.

Tuning Parameter	Transformer	Chosen
dmodel	(128, 256, 512)	512
nhead	(4, 6, 8)	8
nlayers	(2, 3, 4)	4
input window	(2, 5, 10, 20)	10
output window	(1, 2, 5, 10)	2

Table 3.1 Set of tuning parameter for grid search

Although not selected through grid search, the following tuning parameters were used in the model.

- epoch : An epoch refers to an iteration that is completed once for the entire data set. We choose the number of epochs to be 200, which is a suitable value obtained by observing the train loss and test loss. This choice helps prevent both overfitting and underfitting by monitoring the loss.

- batch size : A size of the data sample assigned for each batch. We choose 64 in our model.

- scaler : A preprocessing step that aims to normalize the input data. We choose min-Max scaling in our model.

- optimizer : A method that is utilized to adjust the parameters of a neural network model during the training process. It aims to minimize the loss by iteratively updating the weights and biases based on the computed gradients of the loss function. We choose Adam optimizer in our model.

3.2 Monte Carlo Simulation

We conducted a Monte Carlo simulation to evaluate the performance of the LSCUSUM test using a transformer on a time series with an autoregressive structure. We measured the size of both the classical method and our proposed method for a linear AR(2) model and linear ARMA(1,1) model. To measure the size of two models, we generated time series from the model below;

$$(M1) \quad y_t = 0.6y_{t-1} + 0.3y_{t-2} + \epsilon_t$$

$$(M2) \quad y_t = 0.6y_{t-1} + \epsilon_t + 0.6\epsilon_{t-1}$$

where ϵ_t are IID normal random variables with mean 0 and variance σ^2. We generated 1000 sample size time series data under no changes. With 1000 repetitions, we counted the number of rejections of the null hypothesis. Table3.2 and Table 3.3 reports the empirical sizes for two model.

M1	Transformer	Classical
T_n^{LS}	0.100	0.047
T_n^{max}	0.178	0.049

Table 3.2 Empirical size of AR(2) Model

M2	Transformer	Classical
T_n^{LS}	0.122	0.048
T_n^{max}	0.166	0.049

Table 3.3 Empirical size of ARMA(1,1) Model

Additionally, we assessed the power of the test for a linear AR(1) model under two scenarios: one with varying autoregressive coefficients and another with varying error term variance. Under the alternative, we generate 1000 sample size time series changing coefficient in the middle. we counted the number of rejections of the null hypothesis. Table 3.3 and Table 3.4 reports the empirical power for AR(1) model with $\phi = 0.3$ and $\sigma^2 = 1$ under alternatives.

$\phi = 0.7$	Transformer	Classical
T_n^{LS}	0.878	0.872
T_n^{max}	0.906	0.888

Table 3.4 Empirical power of AR(1) Model with varying autoregressive coefficient

$\sigma^2 = 2$	Transformer	Classical
T_n^{LS}	0.980	0.916
T_n^{max}	0.981	0.934

Table 3.5 Empirical power of AR(1) Model with varying error term variance

Chapter 4

Empirical Applications

In this chapter, we applied our Transformer based LSCUSUM test to two sets of real data: stock price index SnP500 from 2017 to 2019 and exchange rate dollar to won from 2018.6.25 to 2023.6.23. The models were trained using 70 percent of the data, and predictions were made for the entire dataset. We used log-returns of the SnP500 data for stationarity. The exchange rate data was fitted after applying first-order difference.

First, the Figure 4.1 shows the fitting results for the SnP data. We obtained $T_n^{LS} = 13.57 (> 2.45)$, $T_n^{max} = 8.66 (> 1.46)$, both indicated rejection of the null hypothesis. Both statistics point to a change point around mid-January 2018 for the exchange rate data, which is figured as solid and dashed line.

Second, the Figure 4.2 shows the fitting results of the differenced exchange rate data. We obtained $T_n^{LS} = 19.88 (> 2.45)$, $T_n^{max} = 16.57 (> 1.46)$, both indicated rejection of the null hypothesis. Both statistics point to a change point around mid-Fabuary 2022 for the exchange rate data, which is figured as solid and dashed line.

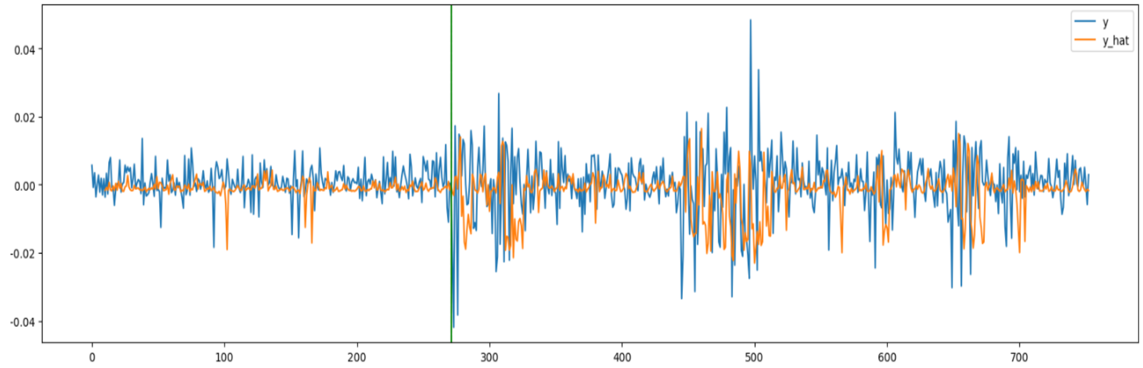

Figure 4.1 SnP index, 2017-2019

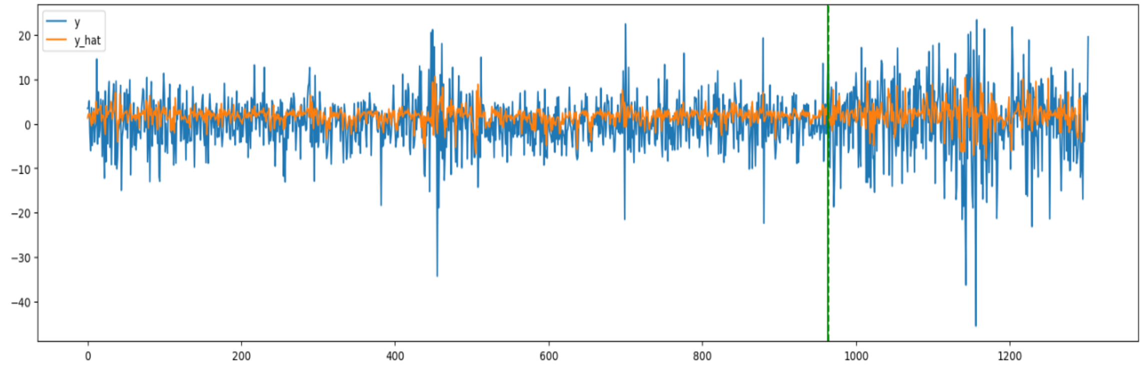

Figure 4.2 USD/KRW, 2018.6.25-2023.6.23

Chapter 5

Conclusion and Discussion

We developed a prediction model for time series forecasting based on Transformers. Using this model, we obtained residuals and applied them to the LSCUSUM test. By performing a grid search, we determined the optimal tuning parameter and calculated the size and power of the Transformer-based LSCUSUM test. We then applied our method to both SnP500 data and exchange rate data, and in both cases, it identified suitable change points.

The size obtained through Monte Carlo simulation exhibited a certain level of distortion, which appears to be attributed to autocorrelation among the residuals. During the simulation, we were able to observe plots showing autocorrelation among the residuals, indicating a lack of independence and potentially leading to inflated values of the test statistics. To address this issue, it is necessary to either apply an autoregressive process to the residuals or adjust the values of both test statistics, $hat\tau_1, n^2l, hat\tau_2, n^2l$, which are used. In Lee et al. (2021), an approach considering long run variance incorporating autocovariance was attempt to as a means to address this issue.

In Shi et al. (2022), The Transformer model has shown better results in modeling long-term irregular patterns compared to traditional time series modeling approaches. As a result, it is expected to be able to detect more complex and irregular patterns of change, surpassing the capabilities of simulations.

Bibliography

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lee, S. (2020). Location and scale-based cusum test with application to autoregressive models. *Journal of Statistical Computation and Simulation*, 90(13):2309–2328.

Lee, S., Kim, D., and Seok, S. (2021). Modeling and inference for counts time series based on zero-inflated exponential family ingarch models. *Journal of Statistical Computation and Simulation*, 91(11):2227–2248.

Lee, S., Lee, S., and Moon, M. (2020). Hybrid change point detection for time series via support vector regression and cusum method. *Applied Soft Computing*, 89:106101.

Lee, S., Na, O., and Na, S. (2003). On the cusum of squares test for variance change in nonstationary and nonparametric time series models. *Annals of the Institute of Statistical Mathematics*, 55:467–485.

Oh, H. and Lee, S. (2018). On score vector-and residual-based cusum tests in arma–garch models. *Statistical Methods & Applications*, 27:385–406.

Oh, H. and Lee, S. (2019). Modified residual cusum test for location-scale time series models with heteroscedasticity. *Annals of the Institute of Statistical Mathematics*, 71(5):1059–1091.

Page, E. (1955). A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527.

Ri, X.-h., Chen, Z., and Liang, Y. (2023). Detecting structural change point in arma models via neural network regression and lscusum methods. *Entropy*, 25(1):133.

Shi, J., Jain, M., and Narasimhan, G. (2022). Time series forecasting (tsf) using various deep learning models. *arXiv preprint arXiv:2204.11115*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

국문초록

본 논문에서는 트랜스포머 기반 딥러닝 모델을 이용하여 자기회귀 시계열에서 구조적 변화 지점을 탐지하는 방법에 대해 논의한다. 변화 지점 감지를 위한 방법 중 하나인 LSCUSUM 검정을 사용하여 구조적 변화를 감지할 수 있는데, LSCUSUM 검정의 가장 중요한 측면은 잔차를 정확하게 추정하는 것이므로, 적절한 모델을 선택하는 것이 가장 중요하다. 많은 시계열이 비선형 특성을 나타내므로 이러한 비선형성을 포착하고 효과적으로 모형화하기 위해 딥 러닝 방법을 사용하는 것이 필수적이다. 따라서 우리는 self-attention 메커니즘을 활용하는 transformer 기반 딥 러닝 모델을 활용한다. 또한, 본 연구에서는 transformer 기반의 변화점 탐지 방법에 대한 크기와 검정력을 Monte Carlo simulation으로 계산하고, 두 실제 데이터 세트에 적용한다.

주요어: Structural break, Deep learning, Tranformer, LSCUSUM

학번: 2021-26067