

Forecasting volatility index by temporal convolutional neural network

Ji Won Shin^a, Dong Wan Shin^{1,b}

^aInstitute of Mathematical Sciences, Ewha Womans University;

^bDepartment of Statistics, Ewha Womans University

Abstract

Forecasting volatility is essential to avoiding the risk caused by the uncertainties of a financial asset. Complicated financial volatility features such as ambiguity between non-stationarity and stationarity, asymmetry, long-memory, sudden fairly large values like outliers bring great challenges to volatility forecasts. In order to address such complicated features implicitly, we consider machine learning models such as LSTM (1997) and GRU (2014), which are known to be suitable for existing time series forecasting. However, there are the problems of vanishing gradients, of enormous amount of computation, and of a huge memory. To solve these problems, a causal temporal convolutional network (TCN) model, an advanced form of 1D CNN, is also applied. It is confirmed that the overall forecasting power of TCN model is higher than that of the RNN models in forecasting VIX, VXD, and VXN, the daily volatility indices of S&P 500, DJIA, Nasdaq, respectively.

Keywords: deep learning, TCN, volatility forecasting, volatility index, time series

1. 서론

이 연구에서는 자산의 다양한 변동성 지표 중 하나로 인지도가 높은 시카고옵션거래소(CBOE)의 변동성 지수 VIX, VXD, VXN을 다루었다. VIX는 S&P 500 지수 옵션의 변동성에 대한 시장의 기대를 나타내는 지수이고, VXD는 DJIA에 대한 지수, VXN은 나스닥에 대한 지수이다. 변동성 지수(volatility index)가 커지면 그에 대응하는 자산 지수 옵션의 변동성이 커질 것 이라는 기대가 높아짐을 의미하며, 따라서 투자자들의 불안 심리가 커지고 자산 가격의 등락폭이 커져 자산 가격의 리스크(risk)가 커진다고 할 수 있다. 변동성은 투자자들이 체감하는 리스크의 지표로 주가, 환율, 옵션 등의 금융 자산의 위험관리에 있어서 매우 중요한 역할을 하기 때문에 변동성의 정확한 예측은 중요하다. 또한 다른 여러 가지 변동성 측도에 비해 변동성 지수는 이 지수 그 자체로도 또 하나의 금융 자산이기 때문에 이 지수로부터 파생되는 다양한 금융 상품이 존재하므로 변동성 지수를 정확하게 예측 하는 것은 여러 측면으로 중요하다.

변동성 데이터는 비정상성(non-stationarity)과 정상성(stationarity)의 경계적 특성이 있으며, 자산 가격 하방 쇼크에 더 큰 폭으로 상승하는 비대칭성과 상당한 장기 기억성, 시장에 큰 사건이 발생할 때 기존의 값들에 비해 이상치라 할 수 있을 정도의 예측할 수 없는 큰 값이 발생하는 특성들이 존재한다. 이렇게 여러 가지 복잡한 특성들은 하나의 모형으로 구조화되기 어려워서 전통적인 방식의 모형으로는 변동성에 대한 예측력을

This research was supported by a Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2019R1A6A1A11051177).

¹ Corresponding author: Department of Statistics, Ewha Womans University, Seoul 03760, Korea. E-mail: shindw@ewha.ac.kr

Table 1: Basic statistics for volatility data sets

| | n | Mean | Sd | Min | Median | Max | Skewness | Kurtosis |
|-----|------|-------|------|-------|--------|-------|----------|----------|
| VIX | 8186 | 19.58 | 8.02 | 9.14 | 17.68 | 82.69 | 2.16 | 8.20 |
| VXD | 3218 | 17.60 | 6.62 | 2.71 | 15.69 | 67.07 | 2.45 | 9.93 |
| VXN | 3215 | 21.09 | 7.29 | 10.31 | 19.05 | 80.08 | 1.79 | 5.85 |

높이기 어려운 면이 있다. 변동성의 정확한 예측은 자산의 리스크에 대비하는 데에 중요한 역할을 하기 때문에 필수적이다. 이러한 문제를 해결하기 위해 이 연구에서 최근 각광 받고 있는 딥러닝 방법의 하나인 TCN (temporal convolutional network) 모형이 좋은 방안이 될 것이라 보고, causal TCN 모형을 변동성 예측에 적용하고 예측력을 최대화 할 수 있는 TCN 구조를 설계하고자 하였다. 기존 시계열 분석에 적합한 딥러닝 방법인 RNN 계열의 LSTM (long short-term memory) 모형과 그 발전된 모형들에 비해서도 TCN 모형이 여러 가지 면에서 더 우수할 수 있는지를 분석하고자 한다.

이후 본 논문은 다음과 같이 구성된다. 2절은 논문에 쓰인 변동성 지수 데이터의 기본적인 통계와 특성을 다루고, 3절은 TCN 딥러닝 모형의 소개, 4절은 TCN 모형과 RNN 모형들의 예측력 비교, 5절은 요약 및 결론으로 마무리 한다.

2. 데이터

자산 가격의 변동성은 투자자들이 체감하는 리스크(risk)의 지표로 주가나 환율, 옵션 등의 금융 자산의 위험 관리에 있어서 매우 중요한 요소이다. 변동성 데이터는 비정상성과 정상성의 경계적 특징을 가지고 있으며, 전날의 하방 쇼크에 더 큰 폭으로 반응하는 비대칭성과 상당한 장기 기억성 등의 복잡한 특성을 지니고 있다. 변동성 데이터의 특징은 Poon과 Granger (2003), McAleer와 Medeiros (2008), Bucci (2017), Shin과 Shin (2019), Shin과 Shin(2022) 등의 여러 연구에서 확인 할 수 있다.

S&P 500, DJIA, Nasdaq 100에 해당하는 IV의 인덱스는 VIX, VXD, VXN 이며 시카고 선물 거래소의 데이터를 사용하였다. 데이터는 시카고 선물 거래소에서 제공하는 기간부터 사용하여 VIX는 1990.01 ~ 2022.06 기간의 데이터를 사용하였고, VXD와 VXN은 2009.09 ~ 2022.06 기간의 데이터를 사용하였다. Table 1은 VIX, VXD, VXN의 기초 통계량이다.

Table 1에서 세 가지 변동성 지수의 첨도(kurtosis)가 5 ~ 8 정도로 강한 중심 집중 분포를 나타내며 왜도(skewness)가 2 정도로 비대칭성이 있는 것을 알 수 있고, 최댓값(max)과 중앙값(median) 차이가 중앙값과 최솟값(min)의 차이보다 4 ~ 8배 정도로 변동성 지수의 분포는 우측으로 상당히 큰 비대칭성을 나타내는 것을 확인 할 수 있다. Figure 1은 3가지 변동성 지수의 움직임을 나타낸 시도표(time series plot)와 자기 상관 함수(autocorrelation function; ACF)이다.

Figure 1의 시도표에서 2008년 세계 금융 위기 이후와 2020년 부분 코로나(COVID-19)로 인한 금융 시장의 불안정성이 커지는 부분에서 변동성 지수가 크게 상승하는 것을 확인할 수 있으며, 크게 상승한 이후에는 장기적으로 계속 큰 값을 갖으며 서서히 감소하는 추세를 나타냄을 확인할 수 있다. 이처럼 변동성 지수는 큰 값이 발생한 후 바로 다시 평균치로 돌아가지 않고 장시간에 걸쳐 큰 값을 가지며 천천히 감소하는 장기 기억 특성이 있으며, 이는 Figure 1의 ACF 그림에서 100 차시가 넘어가도록 자기 상관 함수가 유의함을 통해서도 확인할 수 있다. 그림에서 100차시까지만 나타내었지만 실제로 200차시가 넘어가도록 자기 상관 함수가 유의한 자기 상관관이 아주 강한 데이터이다.

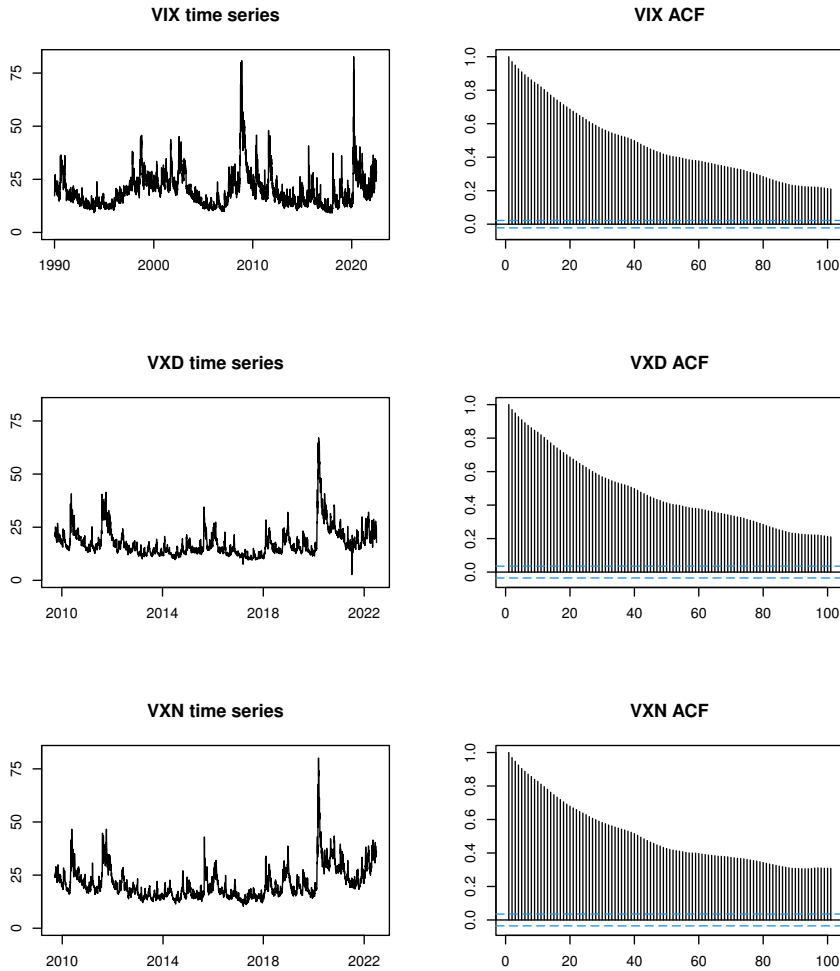


Figure 1: Time series plot and ACF plot of volatility data sets.

3. 변동성 예측에 적합한 TCN 모형과 RNN 계열 모형의 구조 설계

TCN 모형은 CNN (convolution neural network)이 발전된 형태로 1D-CNN의 변형층들이 여러층 쌓여있는 구조를 지닌다. CNN 모형은 사진이나 영상 등에 많이 사용되었다가 시계열 데이터에도 점점 적용되고 있는 추세이다. 기존에는 시계열 데이터에 대한 딥러닝 방법으로 RNN (recurrent neural network) 계열의 LSTM, GRU가 적용가능하여 많이 이용되었으며, 그 방법의 확장으로 Bi-LSTM, CNN-LSTM 등이 있다. 그러나 최근 다양한 데이터에서 TCN 모형이 RNN 계열의 모형에 비해 예측력이 더 우수하다는 Bai 등 (2018), Liu 등 (2019), Lara 등 (2020), Hewage 등 (2020), Bi 등 (2021), Li 등 (2022) 등의 연구들이 발표되고 있다. 논문에서의 결과들은 TCN 구조가 교통 예측이나 연설, 기계 번역, 동작 인지 등의 다양한 시계열 자료에서 기존의 lstm 같은 RNN 모형보다 연산 속도나 예측력면에서 뛰어나고, 또한 RNN 모형을 능가하는 장기 기억력을 가짐을 입증하였다.

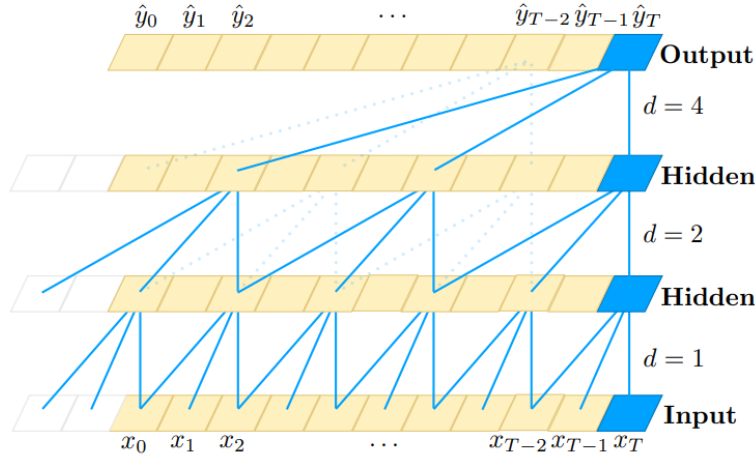


Figure 2: Architectural elements in a TCN. A dilated causal convolution with dilation factors $d = 1, 2, 4$ and filter size $k = 3$. The receptive field is able to cover all values from the input sequence.

TCN 모형은 크게 세가지 부분으로 구성되어있는데, 첫째 causal convolutions과 둘째 dilated convolutions, 그리고 셋째 residual blocks 이다. 인과적 컨볼루션(causal convolution)으로 입력된 자료에서 특성을 추출하고 여러 층에서는 확장된(dilated) 필터 자료를 띄엄띄엄 사용함으로써 더 먼 과거의 자료를 효과적으로 특성을 추출하는데 사용한다. 또한 같은 확장 크기를 지닌 2개의 층을 하나의 residual 블록으로 하여 블록을 쌓는 방식으로 조금 더 안정적인 추정이 가능하게 한다. Figure 2는 Bai 등 (2018) 논문의 그림으로 TCN의 기본 구조 모형을 나타낸 것이다. 각 층마다 커널 $k = 3$ 사이즈를 이용하고, dilation d 를 1, 2, 4로 증가시켜가면서 과거의 자료를 효율적으로 사용하고자 하는 모습을 나타낸 것이다. 여기서 이 그림을 우리의 변동성 지수 데이터와 일치시키면 x_t 는 t 일의 변동성 지수이고, y_t 는 변동성 지수의 1일 후 예측치이다.

TCN으로 시계열을 예측할 때는 인과적(causal) 방법을 적용하여 미래에서부터 과거로의 정보 유출이 없도록 한다. TCN의 input을 $X = (x_0, x_1, \dots, x_{T-1})$ 로 나타내면 output은 $Y = (y_1, y_2, \dots, y_T)$ 로 나타낼 수 있다. y_t 를 예측할 때는 t 시점의 정보는 사용하지 않고, $t - 1$ 시점까지의 정보에만 의존하여 예측한다. RNN과 같이 input으로 준 데이터의 길이와 output 데이터의 길이가 동일하다. 이를 위해 output hidden layer는 input layer와 동일한 길이를 갖는 1D fully-convolutional network (1D FCN)을 사용한다. 이전 layer와 동일한 길이를 유지하기 위해서는 kernel size를 k 라고 할 때 $k - 1$ 개 만큼의 0을 앞에 추가한다(zero padding). d 사이즈의 dilation이 추가된다면, zero padding은 $(k - 1) \times d$ 개를 추가한다.

1D FCN로만 이루어진 기본 구조에서는 장기 기억을 위해서 깊은 network구조나 kernel size k 가 커질 수밖에 없는데, 구조가 깊어지거나 kernel size가 커지면 모형 적합에 많은 연산량이 필요하고 제대로 적합이 되지 않는 문제가 발생할 수 있다. Receptive field는 kernel이 한 번에 보는 영역을 의미하는데 dilated convolutions는 receptive field는 크게 만들어주면서도 연산량의 증가를 가져오지 않는 효과적인 방법이다. d 사이즈의 dilation은 과거의 정보를 효과적으로 이용하기 위해 데이터를 일부 건너 뛰어가며 d 개의 자료마다 한 개씩 자료를 사용하여 같은 개수의 자료이더라도 더 과거의 정보를 사용할 수 있도록 한 방법이다.

TCN은 RNN에 비해 병렬 연산이 가능하기 때문에 학습 속도가 뛰어나고, 기울기 소실(vanishing gradients)문제가 없다. 전통적인 CNN이 특성 추출 능력이 뛰어났지만 분류(classification)데이터에 적합했다면, TCN은 특성 추출 능력도 가지면서 시계열 자료에 적합한 1D CNN의 발전된 형태이다.

Table 2는 TCN과 RNN 계열의 여러 가지 딥러닝 모형에 대한 최적 하이퍼파라미터를 찾기 위해 시행한

Table 2: The hyperparameter settings of grid search

| Hyperparameter | grid |
|---------------------------------------|--------------------|
| Number of hidden nodes | 8, 16, 32, 64, 128 |
| Input length | 22, 264, 528 |
| Window size of scaling for RNN | 264 |
| Layer for RNN | 2, 4 |
| Dropout _{rate} | 0 |
| Learning rate | 0.001 |
| Epoch(early stopping) | 100(patient 10) |
| Number of filters | 16, 32, 64, 128 |
| Convolution kernel size | 2, 3, 5, 7, 8 |
| Input length | 22, 264, 528 |
| Number of stacks | 1,2,3 |
| TCN dilations for each residual block | 1, 2, 4, 8, 16, 32 |

Table 3: The specified hyperparameters of models for data sets

| | | TCN | LSTM | GRU | BiLSTM | CNN-BiLSTM |
|-------------------------|------------------------|---------|------|-----|--------|------------|
| VIX | Number of hidden nodes | - | 16 | 32 | 16 | 32 |
| | Number of layer | - | 2 | 2 | 2 | 2 |
| | Input length | 22 | 22 | 22 | 22 | 22 |
| | Number of filters | 32 | - | - | - | 32 |
| | Kernel size | 2 | - | - | - | 2 |
| | Number of stacks | 1 | - | - | - | - |
| | Dilations | (1,2,4) | - | - | - | - |
| VXD | Number of hidden nodes | - | 16 | 8 | 16 | 32 |
| | Number of layer | - | 2 | 4 | 2 | 2 |
| | Input length | 22 | 22 | 22 | 22 | 22 |
| | Number of filters | 16 | - | - | - | 32 |
| | Kernel size | 2 | - | - | - | 2 |
| | Number of stacks | 1 | - | - | - | - |
| | Dilations | (1,2,4) | - | - | - | - |
| VXN | Number of hidden nodes | - | 16 | 32 | 16 | 32 |
| | Number of layer | - | 2 | 2 | 2 | 2 |
| | Input length | 22 | 22 | 22 | 22 | 22 |
| | Number of filters | 64 | - | - | - | - |
| | Kernel size | 3 | - | - | - | 3 |
| | Number of stacks | 1 | - | - | - | - |
| | Dilations | (1,2,4) | - | - | - | - |
| dropout _{rate} | 0 | | | | | |
| Learning rate | 0.001 | | | | | |
| Optimizer | Adam | | | | | |
| activation | Relu, Tanh, sigmoid | | | | | |
| Loss function | MAE | | | | | |
| Epoch(early stopping) | 100(patient 10) | | | | | |

grid search값들이다. 각 데이터셋 별로 손실 함수(loss function) MAE기준 최적인 하이퍼파라미터로 모형을 적합시켰다. Hidden nodes는 8, 16, 32, 64 등 보편적으로 grid search에 많이 이용하는 값들을 사용하였으며, input length는 주가 변동성 데이터의 특성상 1개월(22일), 1년(264일), 2년(528일)을 선택하였다. Window scaling은 Chou와 Ngom (2016)과 Lee 등 (2017) 을 참고하여 RNN 계열 모형의 예측력 상승을 위하여 사용하였으며, grid search를 통해 찾은 각 모형의 데이터별 최적 하이퍼파라미터는 Table 3에 나타내었다.

Table 3에서 가장 예측력이 우수하였던 최적 하이퍼파라미터는 한 달치 데이터인 22개의 input length, kernel size는 2나 3으로 크지 않은 사이즈, dilations는 1, 2, 4인 층만 쌓아올린 구조, number of stacks은 모든 데이터에서 1로 TCN 구조를 여러겹으로 반복해서 쌓을 필요 없는 간단한 TCN 구조임을 확인 할 수 있다. TCN 기본 구조 자체가 단순하고, 그중에서도 깊지 않고 단순한 구조가 다음 섹션에서 보여주듯이 변동성 지수 예측에서 우수하다는 것은 TCN 모형의 다양한 곳에서의 활용에 대한 가능성을 보여준다. 모형 적합에서 dropout은 시키지 않았으며, 학습률(learning rate)은 0.001로 주고 patient를 10으로 주어서 검증 데이터의 손실 함수에 개선이 없으면 early stop을 시켜 딥러닝 모형의 과적합을 방지하였다. RNN 계열 모형에서는 기본 구조에서 사용되는 tanh, sigmoid 함수를 활성화 함수로 사용하였으며, TCN에서는 ReLU를 활성화 함수로 사용하였다.

위와 같이 함으로써 input $X = (x_0, x_1, \dots, x_{T-1})$ 로부터 x_T 의 1-Step 예측치인 y_T 를 구하는 TCN 모형 $y_T = \text{TCN}(x_0, \dots, x_{T-1})$ 을 구축 할 수 있다. 다단계 예측을 위해서는 2-Step 예측치 y_{T+1} 를 관측된 input으로 간주하여 위에서 구축한 TCN 모형을 적용하여 구한다. 즉, $y_{T+1} = \text{TCN}(x_0, \dots, x_{T-1}, y_T)$ 이다. 이 방법을 반복적으로 적용하여 다단계 예측치 $y_{T+h-1} = \text{TCN}(x_0, \dots, x_{T-1}, y_T, y_{T+h-2})$, $h = 1, 2, 3, \dots$ 을 구할 수 있다.

4. TCN 모형과 RNN 모형들의 예측력 비교

변동성 지수 예측력 비교를 위해 2절에서 기초 분석한 미국 주요 주가 지수의 변동성 지수 세 가지 데이터셋을 사용하였다. 각 데이터의 앞쪽의 80% 데이터를 모형 학습 데이터(training data)로 사용하고 뒤쪽의 20% 데이터를 평가 데이터(test data)로 사용하여 표본 외 h-step 예측(out-of-sample h-step-ahead, $h = 1, 2, 5$)을 하였다. 여기서 $h = 1, 2, 5$ 는 각각 1일, 2일, 1주일에 대응한다. 학습 데이터의 후반 20%는 모형 검증 데이터(validation data)로 사용하였다. 예측력 평가 지표로는 MAE, MAPE와 RMSE의 세 가지 손실 함수를 사용하였으며 다음과 같다.

$$\begin{aligned} \text{MAE} &= (m - h + 1)^{-1} \sum_{t=n-m}^{n-h} |\hat{V}_{t+h|t} - V_{t+h}|, \\ \text{RMSE} &= \sqrt{(m - h + 1)^{-1} \sum_{t=n-m}^{n-h} (\hat{V}_{t+h|t} - V_{t+h})^2}, \\ \text{MAPE} &= 100 \times (m - h + 1)^{-1} \sum_{t=n-m}^{n-h} |\hat{V}_{t+h|t} - V_{t+h}| / V_{t+h}, \quad h = 1, 2, 5. \end{aligned}$$

여기서 n 은 각 계열의 길이이고 Table 1에 나와 있다. 또 $\hat{V}_{t+h|t}$ 는 시점 t 에서 예측한 $t+h$ 시점의 변동성, V_{t+h} 의 값이고, $m = 0.20n$ 이다.

Table 4는 위 세 가지 예측력 평가 지표를 사용하여 VIX, VXD, VXN 데이터에 대해 전통적인 방법인 AR 모형과 기존에 시계열 자료에 적합하다 알려진 RNN 계열의 LSTM, GRU, BiLSTM 과 RNN과 CNN 계열의 하이브리드 모형인 CNN-BiLSTM 모형, 그리고 CNN의 발전된 TCN 모형에 대해 평가한 결과이다. 이때 ARIMA 모형의 시차는 BIC를 최소로 하는 값으로 사용하였다.

Table 4: Multi-Step forecasting results

| | model | VIX | | | VXD | | | VXN | | |
|--------|------------|--------------|--------------|---------------|--------------|--------------|---------------|--------------|--------------|--------------|
| | | RMSE | MAE | MAPE | RMSE | MAE | MAPE | RMSE | MAE | MAPE |
| 1-Step | ARIMA | 2.077 | 1.180 | 5.707 | 2.357 | 1.506 | 6.657 | 2.593 | 1.577 | 4.941 |
| | TCN | 2.044 | 1.146 | 5.559 | 2.272 | 1.447 | 6.605 | 2.579 | 1.531 | 4.874 |
| | LSTM | 2.102 | 1.195 | 5.722 | 2.834 | 1.776 | 7.718 | 2.653 | 1.572 | 5.058 |
| | GRU | 2.138 | 1.211 | 5.780 | 2.826 | 1.732 | 7.506 | 2.653 | 1.568 | 5.007 |
| | BiLSTM | 2.104 | 1.163 | 5.609 | 2.519 | 1.534 | 6.823 | 2.649 | 1.568 | 5.009 |
| | CNN-BiLSTM | 2.095 | 1.166 | 5.600 | 2.478 | 1.674 | 7.572 | 2.768 | 1.599 | 5.089 |
| 2-Step | ARIMA | 2.635 | 1.613 | 7.958 | 3.003 | 1.978 | 8.811 | 3.178 | 2.096 | 6.826 |
| | TCN | 2.629 | 1.574 | 7.707 | 2.914 | 1.892 | 8.327 | 3.233 | 2.068 | 6.627 |
| | LSTM | 2.693 | 1.647 | 8.126 | 4.153 | 2.780 | 11.422 | 3.384 | 2.174 | 7.038 |
| | GRU | 2.807 | 1.727 | 8.423 | 3.955 | 2.570 | 10.585 | 3.403 | 2.170 | 6.940 |
| | BiLSTM | 2.697 | 1.597 | 7.878 | 3.399 | 2.144 | 9.026 | 3.347 | 2.152 | 6.923 |
| | CNN-BiLSTM | 2.692 | 1.608 | 7.911 | 3.345 | 2.376 | 10.385 | 3.624 | 2.257 | 7.145 |
| 5-Step | ARIMA | 4.003 | 2.428 | 11.726 | 4.473 | 2.950 | 12.751 | 4.834 | 3.175 | 10.037 |
| | TCN | 4.000 | 2.342 | 11.089 | 4.375 | 2.766 | 11.716 | 4.947 | 3.077 | 9.498 |
| | LSTM | 4.277 | 2.576 | 12.119 | 6.939 | 5.089 | 20.397 | 5.227 | 3.287 | 10.412 |
| | GRU | 4.583 | 2.852 | 13.146 | 6.456 | 4.570 | 18.298 | 5.293 | 3.289 | 10.207 |
| | BiLSTM | 4.134 | 2.375 | 11.205 | 5.162 | 3.218 | 13.001 | 5.173 | 3.231 | 10.141 |
| | CNN-BiLSTM | 4.184 | 2.464 | 11.645 | 5.705 | 4.171 | 17.993 | 5.621 | 3.414 | 10.493 |

Note: Orders of ARIMA model are based on BIC order. ARIMA(1,1,1) is for VIX, ARIMA(1,1,2) is for VXD, and ARIMA(1,1,1) is for VXN.

Table 4에서 VIX, VXD, VXN 데이터 모두에서 TCN 모형이 모든 기준과 모든 단계의 예측에서 전반적으로 가장 우수한 모형임을 확인할 수 있다. Figure 3은 Table 4의 RMSE, MAE, MAPE를 시각적으로 확인하기 쉽도록 상대효율(relative efficiencies)로 나타낸 그래프이다. 상대효율은 두 모형의 평가 지표값을 비율로 나타낸 것으로 분모에 위치하는 모형의 평가 지표에 비해 분자에 위치하는 모형의 평가 지표의 비율을 의미한다. 예를 들면 TCN 모형의 ARIMA 모형에 대한 RMSE 기준 상대효율은 다음과 같이 나타낼 수 있다.

$$RMSE \text{ 상대효율} = \frac{ARIMA \text{ 예측치의 RMSE}}{TCN \text{ 예측치의 RMSE}}$$

Figure 3에서 수평으로 지나가는 가운데에 위치한 선은 상대효율이 1인 지점으로 이 선의 위쪽에 위치하면 TCN 모형이 비교 대상인 모형에 비해 예측력이 우수하였음을 나타낸다. Figure 3의 9가지 그림에서 VXN의 2, 5 Step 예측의 RMSE 기준의 경우만을 제외하면 모두 상대효율선이 1 수평선 위에 존재함을 시각적으로 확인함으로써 TCN 모형이 기존의 ARIMA 모형이나 다른 딥러닝 방법에 비해 변동성에 대한 예측력이 우수함을 확인할 수 있다.

Table 5은 모형들의 MCS 분석 결과이다. MCS (model confidence set) 분석 (Hansen 등, 2011)은 각 손실 함수 기준에서 각 모형의 순위와 p-value를 제공한다. MCS는 예측에 있어서 신뢰할 수 있는 모형의 집합이고, 순위는 비교 대상 모형들의 예측력 순위, p-value는 해당 모형이 MCS에 속할 확률을 나타낸다. MSE, MAE, MAPE기준에서의 MCS 분석을 시행하였다. TCN 모형은 MCS에 속할 확률이 모두 1이며 여러 가지 기준에서 1순위의 모형으로 선택되는 것을 확인할 수 있으며, 다른 딥러닝 방법에 비해서 기초적인 ARIMA 모형도 상당히 좋은 평가를 받은 것을 확인하였다. 이 결과로부터 TCN 모형이 RNN 계열의 딥러닝 방법인 LSTM, GRU, BiLSTM, CNN-BiLSTM 모형과 전통적인 시계열 통계 모형인 ARIMA 모형보다 예측력이 우수한 모형으로 평가되는 것을 확인하였다.

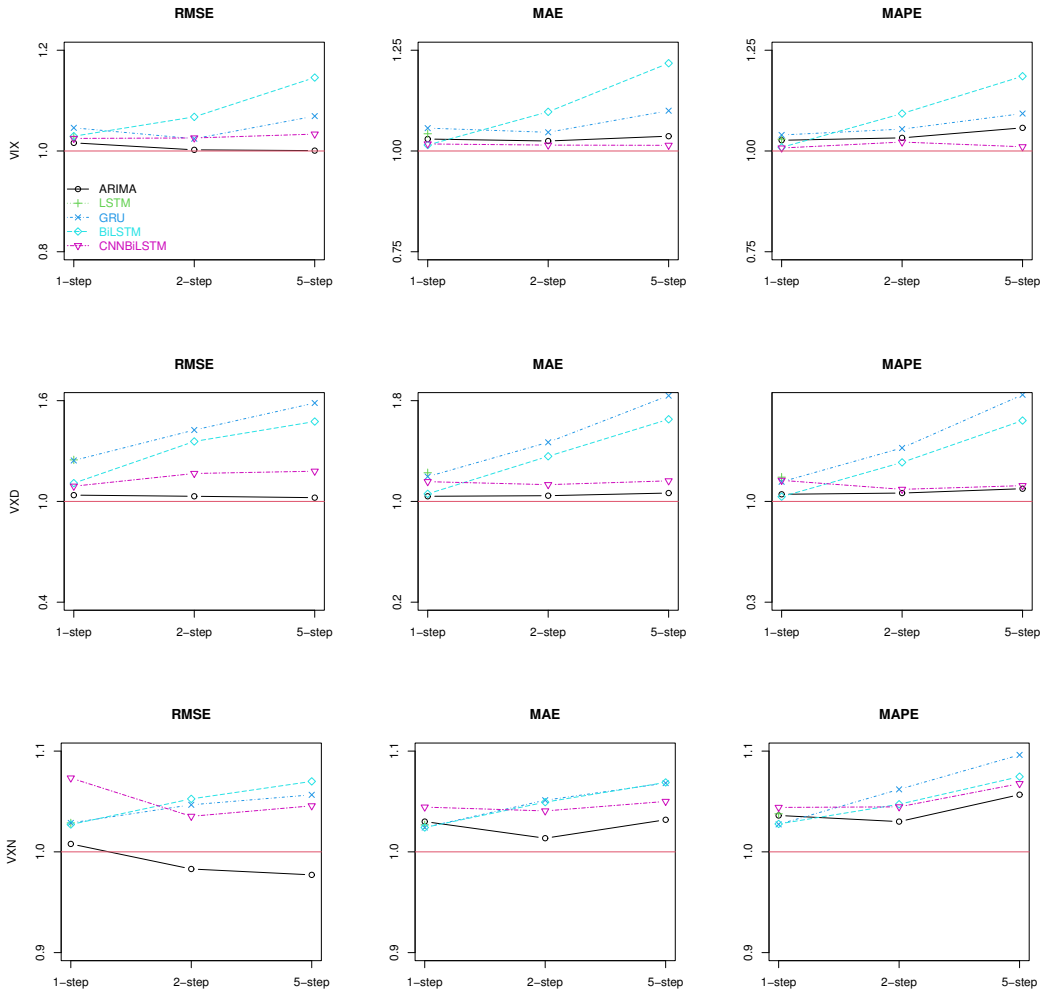


Figure 3: TCN forecasting performance ratio relative to each model.

Table 6은 앞서 다룬 여러 가지 딥러닝 모형들의 running time을 TCN 모형의 running time에 비교하여 상대적인 비율로 나타낸 표이다. Running time 측정은 VIX 데이터를 기준으로 예측력이 최적이었던 하이퍼 파라미터 값들을 기준으로 하여 100회 학습을 하여 모형을 적합 및 예측하여 오차 연산까지 걸리는 총 시간을 측정하였다. Table 6의 파라미터 개수는 각 모형 적합 시에 추정된 weight 들의 수이다.

이 결과로부터 각 모형별 데이터에 최적화된 모형들의 running time면에서 TCN이 가장 효율적임을 확인할 수 있었다.

5. 요약 및 결론

기존에 시계열 자료의 예측에는 Hochreiter와 Schmidhuber (1997)의 LSTM (long short-term memory)과 Chung 등 (2014)의 GRU (gated recurrent unit)이 사용되었는데 RNN 계열의 모형중에서 장기 기억을 위한 모형에 적

Table 5: MCS forecast performance results

| | | VIX | | | VXD | | | VXN | | |
|--------|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| | model | RMSE | MAE | MAPE | RMSE | MAE | MAPE | RMSE | MAE | MAPE |
| 1-Step | ARIMA | 0.53(2) | 0.00 | 0.00 | 0.22(2) | 0.06(2) | 0.38(2) | 0.96(2) | 0.56(2) | 0.35(2) |
| | TCN | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) |
| | LSTM | 0.04(5) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.52(3) | 0.32(3) | 0.10(4) |
| | GRU | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.33(6) | 0.22(4) | 0.09(5) |
| | BiLSTM | 0.53(3) | 0.00 | 0.09(3) | 0.00 | 0.00 | 0.12(3) | 0.44(5) | 0.21(5) | 0.10(3) |
| | CNN-BiLSTM | 0.35(4) | 0.00 | 0.29(2) | 0.00 | 0.00 | 0.00 | 0.44(4) | 0.13(6) | 0.07(6) |
| 2-Step | ARIMA | 1.00(2) | 0.06(5) | 0.00 | 0.61(2) | 0.06(2) | 0.04(2) | 1.00(1) | 0.96(2) | 0.45(2) |
| | TCN | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 0.85(2) | 1.00(1) | 1.00(1) |
| | LSTM | 1.00(4) | 0.08(4) | 0.00 | 0.00 | 0.00 | 0.00 | 0.59(3) | 0.30(4) | 0.18(4) |
| | GRU | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19(6) | 0.13(6) | 0.10(6) |
| | BiLSTM | 1.00(3) | 0.08(2) | 0.06(2) | 0.17(3) | 0.00 | 0.00(3) | 0.55(4) | 0.03(3) | 0.24(3) |
| | CNN-BiLSTM | 1.00(5) | 0.08(3) | 0.06(3) | 0.01(4) | 0.00 | 0.00 | 0.44(5) | 0.24(5) | 0.11(5) |
| 5-Step | ARIMA | 1.00(2) | 0.13(3) | 0.00 | 0.92(2) | 0.12(2) | 0.00(3) | 1.00(1) | 0.96(2) | 0.55(2) |
| | TCN | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(1) | 1.00(2) | 0.95(2) | 1.00(2) | 1.00(1) |
| | LSTM | 0.03(5) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.76(3) | 0.66(4) | 0.44(4) |
| | GRU | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.19(6) | 0.26(6) | 0.17(6) |
| | BiLSTM | 0.56(3) | 0.00 | 0.21(2) | 0.18(3) | 0.00 | 0.00 | 0.68(4) | 0.69(3) | 0.45(3) |
| | CNN-BiLSTM | 0.10(4) | 1.00(2) | 0.00 | 0.02(4) | 0.01(3) | 0.01(2) | 0.58(5) | 0.52(5) | 0.38(5) |

Note: The values in parenthesis are model ranks and the corresponding values are the probability of the model being contained in the model confidence set (MCS).

Table 6: Running time of forecast models

| | Number of parameters | Running time ratio to TCN |
|------------|----------------------|---------------------------|
| TCN | 21287 | 1 |
| LSTM | 3281 | 1.496 |
| GRU | 9729 | 1.664 |
| BiLSTM | 8609 | 2.444 |
| CNN-BiLSTM | 33730 | 2.993 |

합하게 개발된 모형임에도 불구하고 여전히 기울기 소실 문제와 학습 시간과 연산량이 크다는 단점이 있었다. TCN 모형은 입력된 자료에서 특성을 추출하고 층별로 자료를 건너뛰며 사용함으로써 더 먼 과거의 자료를 효과적으로 활용한다. 또한 LSTM에 비해 병렬 연산이 가능한 TCN은 연산 시간이 50% 이상 현저하게 적게 걸리고, 계속 같은 커널을 사용하기 때문에 메모리 소모량도 적다. 무엇보다도 변동성 지표에 대한 예측력이 기존에 시계열 데이터에서 주로 사용하던 RNN 계열의 딥러닝 방법에 비해 우수하였고, 전통적인 시계열 통계 모형인 AR 모형보다도 예측력이 우수함을 확인 하였다. TCN 모형은 변동성 지수 데이터뿐만 아니라 다른 시계열 데이터의 적합에서도 우수한 예측력을 보일 것이 예상되며 향후 여러 분야에서 활용 될 것으로 보인다.

References

Bai S, Kolter JZ, and Koltun V (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, Available from: arXiv preprint arXiv: 1803.01271

- Bi J, Zhang X, Yuan H, Zhang J, and Zhou M (2021). A hybrid prediction method for realistic network traffic with temporal convolutional network and lstm, *IEEE Transactions on Automation Science and Engineering*, **19**, 1869–1879.
- Bucci A (2017). Forecasting realized volatility: A review, *Journal of Advanced Studies in Finance*, **8**, 94–138.
- Chou J and Ngom N (2016). Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns, *Applied Energy*, **177**, 751–770.
- Chung J, Gulcehre C, Cho KH, and Bengio Y (2014). Empirical Evaluation of Gated Recurrent Neural Network-Sequence Modeling. Available from: arXiv: :1412.3555
- Hansen PR, Lunde A, and Nason JM (2011). The model confidence set, *Econometrica*, **79**, 453–497.
- Hewage P, Behera A, Trovati M, Pereira E, Ghahremani M, Palmieri F, and Liu Y (2020). Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station, *Soft Computing*, **24**, 16453–16482.
- Hochreiter S and Schmidhuber J (1997). Long short-term memory, *Neural Computation*, **9**, 1735–1780.
- Lara-Benítez P, Carranza-García M, Luna-Romera JM, and Riquelme JC (2020). Temporal convolutional networks applied to energy-related time series forecasting, *Applied Sciences*, **10**, 2322.
- Lee D, Lee S, Han Y, and Lee K (2017). Ensemble of convolutional neural networks for weakly-supervised soundevent detection using multiple scale input, *Detection and Classification of Acoustic Scenes and Events*, **2017**, 14–18.
- Li W, Wei Y, An D, Jiao Y, and Wei Q (2022). LSTM-TCN: Dissolved oxygen prediction in aquaculture, based on combined model of long short-term memory network and temporal convolutional network, *Environmental Science and Pollution Research*, **29**, 39545–39556.
- Liu Q, Che X, and Bie M (2019). R-STAN: Residual spatial-temporal attention network for action recognition, *IEEE Access*, **7**, 82246–82255.
- McAleer M and Medeiros MC (2008). A multiple regime smooth transition heterogeneous autoregressive model for long memory and asymmetries, *Journal of Econometrics*, **147**, 104–119.
- Poon SH and Granger CWJ (2003). Forecasting volatility in financial markets: A review, *Journal of Economic Literature*, **41**, 478–539.
- Shin JW and Shin DW (2019). Vector error correction heterogeneous autoregressive forecast model of realized volatility and implied volatility, *Communications in Statistics–Simulation and Computation*, **48**, 1503–1515.
- Shin JW and Shin DW (2022). Deep learning forecasting for financial realized volatilities with aid of implied volatilities and internet search volumes, *The Korean Journal of Applied Statistics*, **35**, 93–104.

Received December 14, 2022; Revised December 28, 2022; Accepted December 31, 2022

Causal temporal convolutional neural network를 이용한 변동성 지수 예측

신지원^a, 신동완^{1,b}

^a이화여자대학교 수리과학연구소; ^b이화여자대학교 통계학과

요 약

변동성의 예측은 자산의 리스크에 대비하는 데에 중요한 역할을 하기때문에 필수적이다. 인공지능을 통하여 이러한 복잡한 특성을 지닌 변동성 예측을 시도하였는데 기존 시계열 예측에 적합하다 알려진 LSTM (1997)과 GRU (2014)은 기울기 소실로 인한 문제, 방대한 연산량의 문제, 그로 인한 메모리량의 문제 등이 존재하였다. 변동성 데이터는 비정상성(non-stationarity)과 정상성(stationarity)을 모두 가지고 있는 특성이 있으며, 자산 가격 하방 쇼크에 더 큰 폭으로 상승하는 비대칭성과 상당한 장기 기억성, 시장에 큰 사건이 발생할 때 기존의 값들에 비해 이상치라 할 수 있을 정도의 예측할 수 없는 큰 값이 발생하는 특성들이 존재한다. 이렇게 여러 가지 복잡한 특성들은 하나의 모형으로 구조화되기 어려워서 전통적인 방식의 모형으로는 변동성에 대한 예측력을 높이기 어려운 면이 있다. 이러한 문제를 해결하기 위해 1D CNN의 발전된 형태인 causal TCN (causal temporal convolutional network) 모형을 변동성 예측에 적용하고, 예측력을 최대화 할 수 있는 TCN 구조를 설계하고자 하였다. S&P 500, DJIA, Nasdaq 지수에 해당하는 변동성 지수 VIX, VXD, and VXN, 에 대하여 예측력 비교를 하였으며, TCN 모형이 RNN 계열의 모형보다도 전반적으로 예측력이 높음을 확인하였다.

주요용어: 딥러닝, TCN, 변동성 예측, 변동성 지수, 시계열

이 성과는 한국 연구재단의 지원을 받아 수행된 연구임 (No. 2019R1A6A1A11051177).

¹교신저자: (03760) 서울시 서대문구 이화여대길 52, 이화여자대학교 통계학과. E-mail: shindw@ewha.ac.kr