Volatility Forecasting with 1-dimensional CNNs

Bernadett Aradi, Gábor Petneházi, *University of Debrecen, Hungary*
aradi.bernadett@inf.unideb.hu

Introduction

Volatility is a natural risk measure in finance as it quantifies the variation of stock prices. A frequently considered problem in mathematical finance is to forecast different estimates of volatility. What makes it promising to use deep learning methods for the prediction of volatility is the fact, that stock price returns satisfy some common properties, referred to as stylized facts. Also, the amount of data used can be high, favoring the application of neural networks.

We used 10 years of daily returns for hundreds of frequently traded stocks, and compared different CNN architectures: some networks use only the considered stock, but we tried out using much more series for data augmentation. It turned out that, as expected, the latter construction worked better. We present the results of this experiment, also comparing our dilated causal CNNs with more classical methods, such as an ARIMA model.

Generality of Stock Price Volatility

When working with stock market returns, we should keep in mind that there are some documented common properties of them, usually referred to as stylized facts [Cont 2001, Engle and Patton 2007]. Especially if we want to use deep learning, these similarities in the behaviour of the different assets' returns can come very handy, and they suggest that the variability of asset prices is forecastable.

What are these stylized facts?

- Volatilities cluster – they display positive, significant autocorrelation if the lag is up to a few weeks (persistence).
- Returns and volatilities are negatively correlated (positive and negative shocks have different impact).
- Trading volume is correlated with volatility.
- Volatility exhibits mean reversion (in the long run, it should converge to a normal level).
- Exogenous variables (e.g., other assets or deterministic events) can have an impact on volatility.

The stylized facts not only suggest that volatilities are predictable, but also that knowledge extracted from the price behaviour of some assets might be useful to describe that of other assets. Furthermore, the price development of different stocks might have common driving forces.

[Sirignano and Cont 2019] used a high frequency database of market quotes and transactions to predict the direction of price moves. They found that a universal model trained on all stocks outperforms asset-specific models. The authors claim it is evidence of a universal price formation mechanism.

Those previous findings encourage us to study if the generalities of stock price formation can help volatility forecasting. That is, we study if the volatility history of multiple stocks can be used in a joint (deep learning) system to predict the future volatility of the individual securities.

Convolutional Neural Networks for Time Series Prediction

Convolutional neural networks (CNNs) are most often used with images. However, they can also be used with other data, since the main feature of these networks is that they are capable of extracting local patterns from the input. We can even let go of the restriction of having two dimensions: 1 or 3 dimensional convolutions can be used in pretty much the same way.

In the past few years several CNN architectures were successfully applied to time series forecasting. [Mittelman 2015] used fully convolutional networks to time series modeling, replacing the usual subsamplings and upsamplings by upsampling the filter of the l^{th} layer by a factor of 2^{l-1}. [Yi et al. 2017] presented structure learning algorithms for CNN models, exploiting the covariance structure of multiple time series. [Borovykh, Bohte, and Oosterlee 2017] applied a CNN inspired by the WaveNet, using dilated convolutions. Dilations allow an exponential expansion of the receptive field, without loss of coverage [Yu and Koltun 2015].

Network Architecture

Motivated by the recent successes of CNNs in time series forecasting, we chose to use a dilated causal 1 dimensional convolutional neural network. The inputs to this network are 64-step sequences, while the outputs are the same sequences shifted by 1, so that we look one step ahead into the future. The causal convolution means that the output at one point in time only depends on inputs up to that point, and the data is padded, such that the input and the output have the same length. Dilated convolution (or convolution with holes) makes the filter larger by dilating it with zeros. The dilation rate of the l^{th} layer is set to 2^{l-1}, which allows an exponential receptive field growth, and enables a relatively shallow network to look into a relatively distant past. We use 6 causal convolutional layers with exponentially increasing dilation rates. Each layer uses 8 filters with a kernel size of 2, and a relu activation function. It is then followed by a final convolutional layer with a kernel size of 1 and a single filter, so that the output shape matches that of the given time series sequence. The networks were trained for 300 epochs, using the adadelta optimizer.

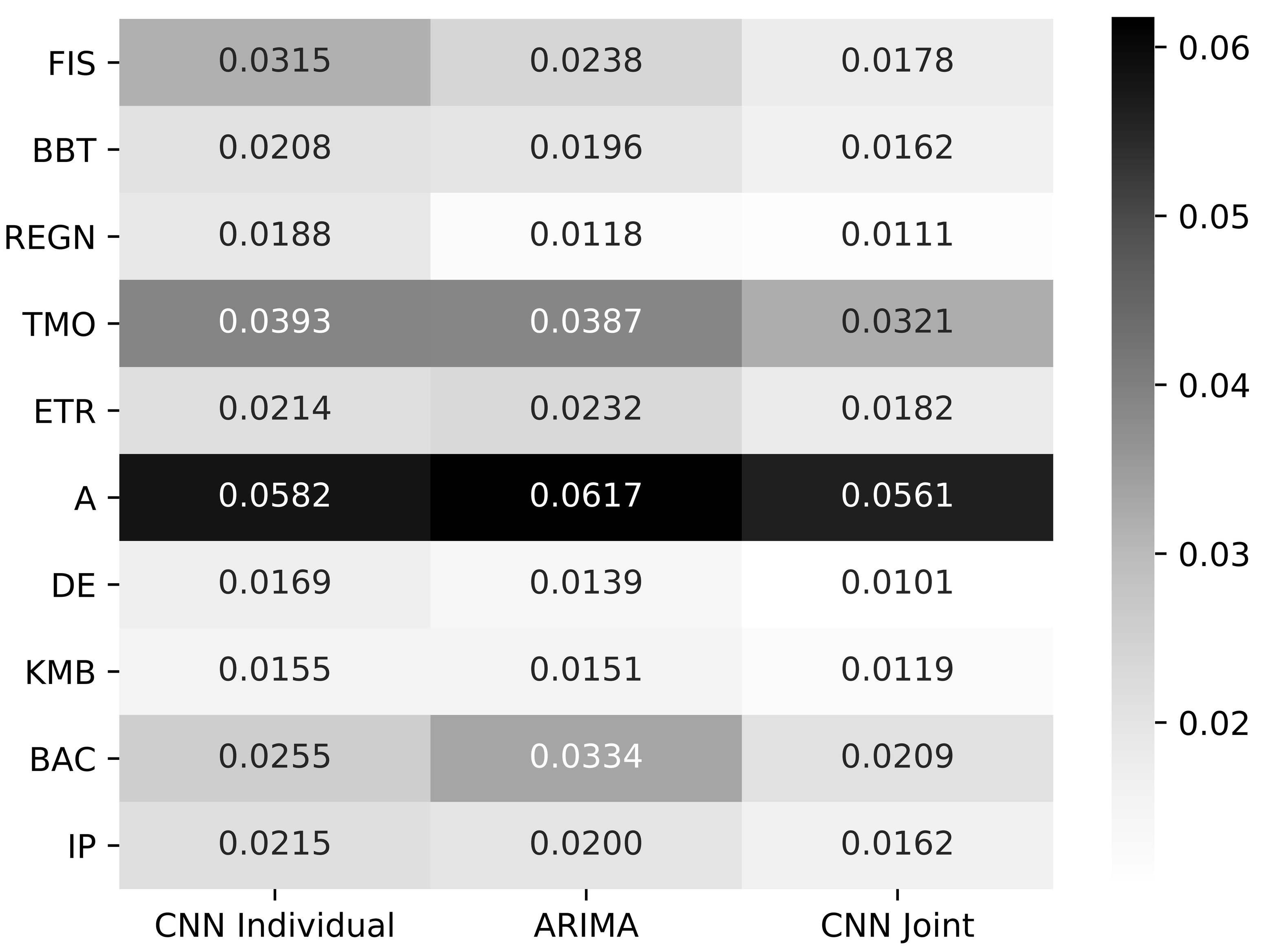

Figure 1: RMSE Scores

Data

We have downloaded 10 years of daily prices for hundreds of frequently traded stocks – constituents of the S&P 500 stock market index. The dataset was obtained through the Python module of Quandl (`https://www.quandl.com/data/WIKI`). Volatilities were estimated as 21-day moving standard deviations of daily logarithmic returns. After removing stocks with more than 10 missing observations, 440 volatility series remained. Each series was split to overlapping 64-day subseries, which were fed to the algorithm to predict the same series shifted by 1 time step. Each value was normalized by subtracting the mean and dividing by the standard deviation of the whole training set.

Experiments

We have randomly chosen 10 stocks, and we used two CNN forecasters for each. The first model learns from the volatility history of the given stock only. The second model learns from all stocks' volatilities, except the chosen 10 stocks. It means that the second model learns from more than 400 times as much data, however it totally disregards the time series that we are forecasting. We also applied an ARIMA model, in order to extend the comparison to a simpler and more classical time series forecasting method. The models were trained and tested on separate time periods: the first 70% of the available nearly 10 years long time period was used for training the models, while the remaining 30% was the evaluation set. We produced one-day-ahead forecasts, and compared the models performance in terms of forecast error and directional accuracy.

Results

	CNN Individual	ARIMA	CNN Joint
Value Forecasts			
RMSE	0.0269	0.0261	0.0211
SMAPE	7.5487	4.9468	4.6977
Direction Forecasts			
Accuracy	0.5387	0.5161	0.5964
F1	0.5210	0.4182	0.7036

The metrics were averaged over the 10 stocks under study. RMSE (root mean squared error) and SMAPE (symmetric mean absolute percentage error) are the reported regression metrics, while directional accuracy and F1 score describe the forecasts ability to get the directions right. The single-stock CNNs poor performance probably stems from the limited data volume. We have also compared our joint convolutional neural network to simple ARIMA models fitted to the individual volatility series. We have chosen the best model based on AIC.

Our CNN trained on multiple stocks outperformed ARIMA forecasts of the individual stock volatilities according to all metrics, even though the ARIMA parameters were chosen using a systematic procedure, while the CNN parameters were chosen rather arbitrarily. The convolutional neural networks performance could probably have been further optimized. The joint CNN model outperformed the single models in terms of forecast error and directional accuracy as well. Figure 1 displays the average distance of forecasted and true values in terms of RMSE. Figure 2 shows directional accuracies.

Conclusions and Future Perspectives

We trained a one dimensional convolutional neural network on multiple stocks volatility rate history, and compared its forecasting performance to benchmark models trained on single series. We found that the deep learning method could take advantage of the multiplied data volume and produce better results. It suggests that the generality of stock prices allows a data expansion that might enable deep learning methods to outperform traditional time series methods in financial forecasting.

These findings open up research opportunities regarding the financial application of deep learning methods. It should be explored if the results apply to different markets and to different forecasting horizons. For example, it would be worth examining if our jointly learned models can help forecasting volatilities of less frequently traded stocks. Or if it works with different data frequencies. We still used very small data – a few hundred stocks with daily price observations. Using intraday stock market data would seem more encouraging.

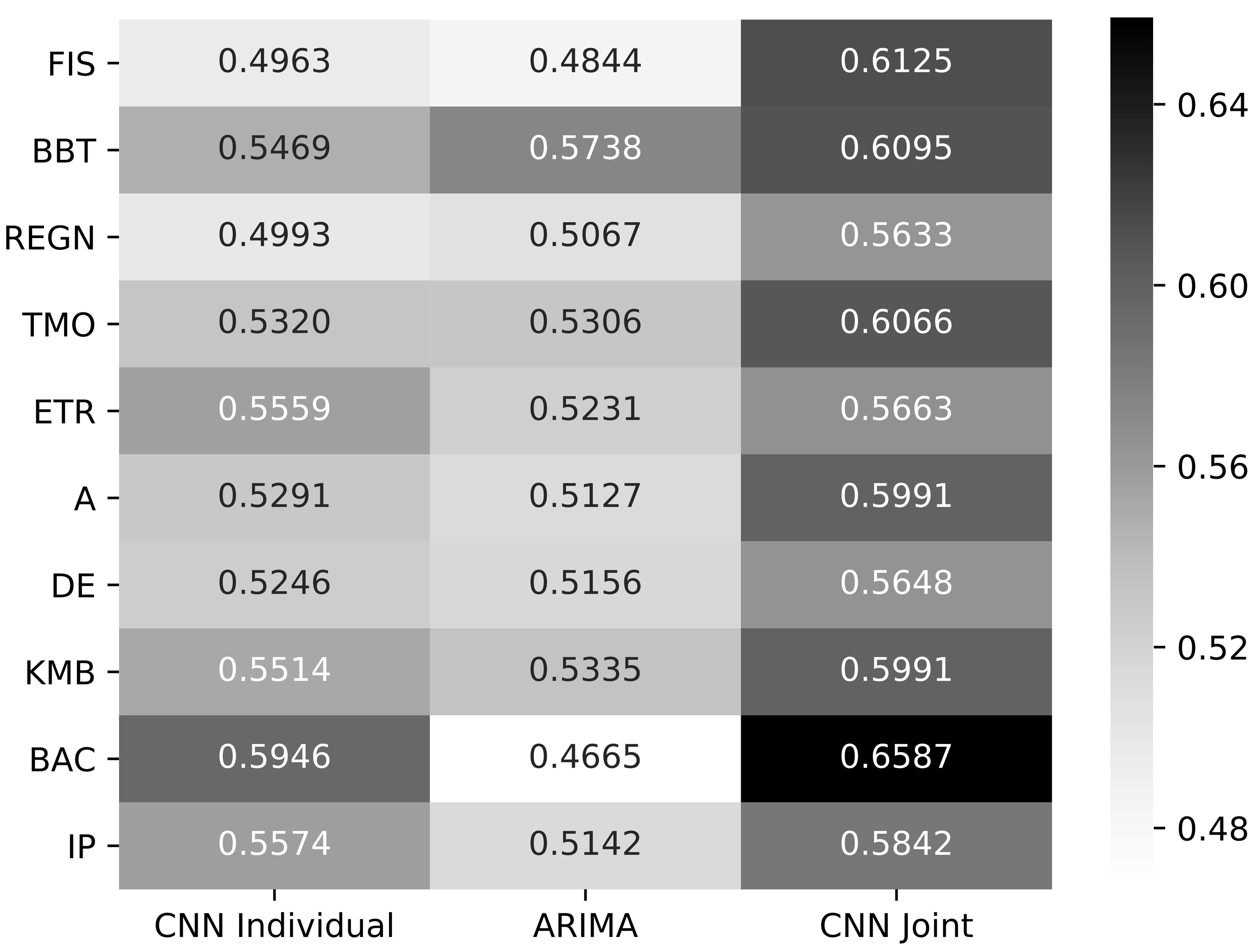

Figure 2: Accuracy Scores

References

A. Borovykh, S. Bohte, and C.W. Oosterlee: *Conditional time series forecasting with convolutional neural networks.* arXiv preprint, arXiv:1703.04691 (2015).

R. Cont: *Empirical properties of asset returns: stylized facts and statistical issues.* Quantitative Finance **1** (2001), 223–236.

R.F. Engle and A.J. Patton: *What good is a volatility model?.* Quantitative Finance **1** (2001), 237–245.

R. Mittelman: *Time-series modeling with undecimated fully convolutional neural networks.* arXiv preprint, arXiv:1508.00317 (2015).

J. Sirignano and R. Cont: *Universal features of price formation in financial markets: perspectives from deep learning.* Quantitative Finance **19** (2019) 1449–1459.

S. Yi et al.: *Grouped convolutional neural networks for multivariate time series.* arXiv preprint, arXiv:1703.09938 (2017).

F. Yu and V. Koltun: *Multi-scale context aggregation by dilated convolutions.* arXiv preprint, arXiv:1511.07122 (2015).