Department of Finance and Risk Engineering

Volatility Forecasting using Temporal Convolutional Networks (TCNs)

Divya Patel, Joaquin Garay, and Roger Li

FRE GY-6123 Financial Risk Management

Prof. James Adams

Fall 2024

1. Abstract

This study explores the use of Temporal Convolutional Networks (TCNs) for forecasting equity market volatility, a key metric in risk management and asset allocation. While traditional models such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and its variants have been widely utilized, TCNs offer distinct advantages, including enhanced parallel processing and the ability to capture short-term dependencies. This research uses return data and intraday volatility measures to compare the predictive performance of a log-linear Realized GARCH(1,2) model and a TCN. The results show that TCNs outperform the Realized GARCH model across 39 out of 40 sample cases based on metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Still, there's no clear outperformance on Mean Square Error (MAE), suggesting that Realized GARCH may perform better on low volatility regimes than TCN. Furthermore, computational performance analysis indicates that TCNs are competitive with GARCH models, demonstrating their feasibility for real-time implementation in financial risk management.

2. Introduction

This project investigates the use of Temporal Convolutional Networks (TCNs) for forecasting equity market volatility, a crucial metric in financial risk management. Volatility forecasting plays a key role in tasks such as portfolio optimization, Value-at-Risk (VaR) estimation, and derivative pricing. Traditionally, models like Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and its variants, including Realized GARCH, have been the dominant tools for modeling and predicting volatility. These models leverage historical return data and, in some cases, intraday volatility measures to capture the dynamic nature of market risk.

Despite their widespread adoption, GARCH models have limitations in capturing complex, nonlinear relationships and adapting quickly to changing market conditions. Neural networks, such as the Long Short-Term Memory (LSTM) model, have demonstrated effectiveness in capturing long-term dependencies in time series data. Building on these advancements, TCNs offer a complementary approach by efficiently modeling short-term dependencies and leveraging parallelism for enhanced computational performance, making them a promising alternative in real-time financial applications.

2.1. Hypothesis

The study aims to test two key hypotheses:

1. **Predictive Performance**: TCNs will outperform the Realized GARCH(1,2) model in one-step-ahead forecasting accuracy, as evaluated through metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error

(RMSE). The Diebold-Mariano test will be used to assess the statistical significance of the performance differences at a 10% significance level.

2. **Computational Efficiency**: While TCN model is expected to perform slower than Realized GARCH during both the training and one-step-ahead prediction steps due to the larger number of parameters, more complex network structure, and reliance on extensive matrix operations, TCNs may achieve competitive computational speed with hardware acceleration due to their parallelizable architecture, which allows greater optimization potential compared to the sequential nature of GARCH. To evaluate the computational cost of the TCN model compared to GARCH, records of the runtime of both models during training and prediction steps will provide a comprehensive comparison across different processing stages.

3. Literature Review

3.1. Framework of Realized Volatility on Forecasting Volatility Models

Historically, volatility modeling has been dominated by parametric models such as the Autoregressive Conditional Heteroskedasticity (ARCH) model introduced by Engle (1982) and its extension, the Generalized ARCH (GARCH), developed by Bollerslev (1986). These models capture the conditional variance of asset returns using past return data. However, their reliance on daily returns often limits their ability to respond to sudden changes in market volatility, particularly during financial crises or periods of heightened market fluctuations.

The introduction of high-frequency data in the early 2000s marked a significant shift in volatility forecasting. Realized volatility measures, derived from intraday high-frequency data, provided a more accurate and timely estimation of volatility over a given period. These measures capture intraday price movements and fluctuations, delivering more detailed information on current volatility levels than the squared return. As a result, realized volatility improves forecasting accuracy, especially in environments with rapidly changing volatility regimes. High-frequency data offer superior predictive power for future volatility when compared to models that rely exclusively on daily return data.

Several models have leveraged the advantages of realized volatility. For instance, the Multiplicative Error Model (MEM) proposed by Engle and Gallo (2006) and Hansen's Realized GARCH model (2012) utilize high-frequency data to achieve better volatility predictions. These models significantly improve performance by incorporating realized measures into the conditional variance framework, allowing them to adapt more quickly to shifts in volatility.

A realized measure simply refers to a statistic that is constructed from high-frequency data. Well-known examples include realized variance, realized kernel, intraday range, number of transactions, and trading volume.

In a more recent study, Petrozziello et al. (2022) applied a similar approach by using high-frequency data and the same dataset (e.g., Dow Jones and NASDAQ indices) to benchmark their neural network model against the Realized GARCH. Their findings confirmed that while neural networks, such as Long Short-Term Memory (LSTM), can outperform traditional models in certain cases, Realized GARCH remains a strong benchmark due to its structured, interpretable framework that performs well under most conditions.

This project draws on the same methodology as Petrozziello's work, highlighting the robustness of the Realized GARCH model as a parametric approach while exploring the potential of more flexible models such as neural networks. The combination of high-frequency data and these advanced models ensures a comprehensive comparison of volatility forecasting methods.

3.2. Deep Learning Advances for Volatility Forecasting

Recent advancements in machine learning, particularly deep learning techniques, have introduced powerful alternatives to traditional econometric models. Long-Short-Term Memory (LSTM) networks, as implemented in recent studies, have demonstrated superior performance in volatility forecasting during periods of high volatility. Temporal Convolutional Networks (TCNs), which are less explored in financial applications, have shown significant promise in capturing temporal patterns, outperforming both GARCH and LSTM models in terms of forecasting accuracy in recent studies. The use of TCNs allows for the modeling of long-range dependencies in time-series data while being computationally efficient, making them well-suited for financial applications like volatility forecasting.

This project builds on the work of Petrozziello et al. (2022) and Zhang et al. (2022), which explore the use of deep learning models for volatility forecasting.

4. Materials and Methodology

4.1. Data

The dataset used will be the same as that in the study by Petrozziello et al. (2022), which includes stock market data from two different indexes and timeframes: the Dow Jones Industrial Average and NASDAQ 100. The first dataset, also used by Hansen et al. (2012), includes 19 assets from the DJIA Index and one exchange-traded fund, SPY. The sample spans from January 2nd, 2002, to August 29th, 2008, with a maximum of 1663 observations per stock, although some have less data. It covers periods of significant market turbulence, such as the Global Financial Crisis (2007-08), making it suitable for testing the performance of volatility models.

The second dataset is related to 20 stocks of the NDX index over the period of December 3rd, 2012, to November 28th, 2017, covering a period of several crises and instability such as the Eurozone Debt Crisis (2012-13), Brexit Referendum (2016), Oil prices collapse (2014-16), among others.

It is worth noticing that the dataset includes already computed values of the realized measures based on a realized kernel using the Parzen kernel function explained in Barndorff-Nielsen et al. (2009).

4.2. Realized GARCH Model

The Realized GARCH model consists of three key components, which are exemplified by the RealGARCH(p,q) specification:

1. **Return Equation**: Models asset returns as a function of the conditional variance.
$$r_t = \mu + \sqrt{h_t} \cdot z_t,$$
where r_t is the asset return, $h_t = var(r_t|F_{t-1})$ the conditional variance (latent volatility), and $z_t \sim i.i.d.(0,1)$ an innovation.
2. **Variance Equation**: Conditional variance depends on both past realized volatility and past conditional variance.
$$h_t = \omega + \sum_{i}^{p} \beta_i \, h_{t-i} + \sum_{j}^{q} \gamma_j x_{t-j},$$
where x_{t-1} represents the realized volatility measure.
3. **Measurement Equation**: Links realized measures of volatility (e.g., realized variance) with latent volatility, integrating high-frequency data for more accurate volatility estimation.
$$x_t = \xi + \psi \, h_t + \tau(z_t) + u_t,$$
where $u_t \sim i.i.d.(0, \sigma_u^2)$ captures the measurement error.

Hansen et al. (2012) presented two different specifications of the Realized GARCH Model, one linear and the other log-linear. The latter is suggested because it induces an ARMA structure for the measurement equation and ensures a positive variance.

Thus, a RealGARCH(1,2) is implemented for every stock model with the following specific structure:

$$r_t = \sqrt{h_t} \cdot z_t \ ,$$

$$\log h_t = \omega + \beta_1 \log h_{t-1} + \gamma_1 \log x_{t-1} + \gamma_2 \log x_{t-2}\ ,$$

$$\log x_t = \xi + \psi \log h_t + \tau_1 z_t + \tau_2 (z_t^2 - 1) + u_t\ ,$$

When assuming Gaussian innovation u_t, the log-likelihood function takes the form of a nice differential function given by

$$l(r, x; \theta) \coloneqq \frac{1}{2} \sum_{t=1}^{n} \log h_t + \frac{r_t^2}{h_t} + \log x_t + \frac{u_t^2}{\sigma_u^2},$$

so the calibration of this type of model is not different than a standard GARCH model, where the aim is to maximize its log-likelihood function.

The one-step-ahead out-of-sample forecast is then obtainable rolling a two-period window as follows:

$$h_t = (h_{t-1})^{\beta_1} \cdot (x_{t-1})^{\gamma_1} \cdot (x_{t-2})^{\gamma_2} \cdot e^{\omega}.$$

4.3. Temporal Convolutional Neural Network Model

The Temporal Convolutional Network (TCN) model is a specialized form of Convolutional Neural Network designed to handle sequential data by capturing long-range dependencies through its causal and convolutional layers. This architecture enables TCNs to effectively model time-series data while preserving orders of events, making them particularly useful for forecasting stock volatility based on historical time-series data.

The architecture of our TCN model comprises a sequence of stacked TCN blocks designed to capture temporal dependencies and patterns across varying time scales in sequential data. Each TCN block in the model includes two causal convolutional layers, followed by layer normalization, rectified linear unit (ReLU) activation, and dropout layers for regularization. The causal convolutions ensure that the model respects temporal causality, allowing only past and present information to influence the current time step, making it well-suited for time-series data.

The dilation rate of each convolutional layer is increased exponentially within the TCN block, enabling each block to capture a larger temporal context without increasing the network depth excessively. By setting the dilation rate in each block to a power of two (1, 2, 4, etc.), the network expands its receptive field exponentially with each additional block. This facilitates efficient capture of long-term dependencies, which are essential in time-series forecasting.

Following each convolutional layer, layer normalization is applied to stabilize the training process and improve generalization by normalizing activations within the network, resulting in faster convergence and better robustness to varying input scales. The ReLU activation function introduces non-linearity into the network, enhancing the model's capability to capture complex temporal patterns and dependencies. Dropout layers with a dropout rate of 0.2 are employed for regularization, which helps prevent overfitting by randomly omitting certain activations during training, thereby promoting robust feature learning.

Each TCN block includes a residual connection, enabling the model to retain information from the original input alongside the transformations performed within the block. The residual connection is adjusted using a 1x1 convolution to match the number of filters in each block when necessary. These residual connections address the problem of vanishing gradients by providing shortcuts for gradient flow, which stabilizes training, particularly in deeper networks. The inclusion of residuals also enhances the model's ability to capture both short-term and long-term dependencies simultaneously.

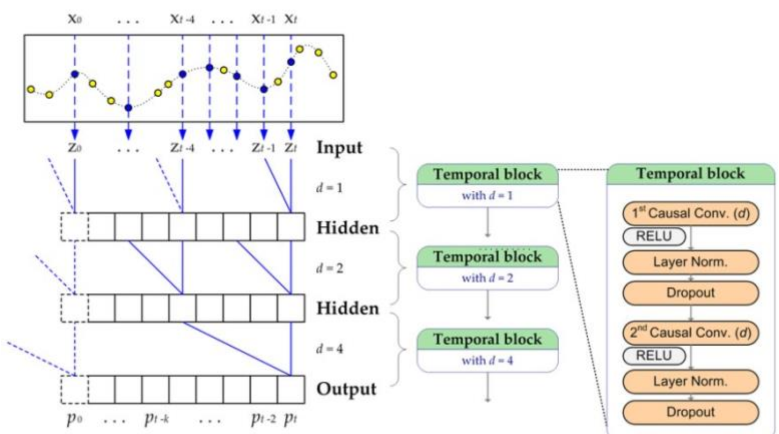

Figure 1: Temporal Convolutional Network Diagram

5. Performance, Analysis, and Expected Results

5.1. Expected Results

The performance of the GARCH models and TCNs is evaluated using MSE, MAE, and RMSE. We expect GARCH models to perform comparably to TCN in terms of MAE and MSE during periods of low volatility, as volatility clustering would be less pronounced, and both models may offer accurate predictions. However, TCN's complexity might come at a higher computational cost, which we will analyze as part of the overall performance evaluation.

During periods of high stress like financial crises, we expect TCN to perform superiorly to GARCH. TCN should capture abrupt volatility spikes more effectively than GARCH models, whose static parameters may lag in response to rapid changes.

By comparing these metrics, we will be able to clearly demonstrate the strengths and weaknesses of each model, providing insights into their suitability for different market conditions and the trade-off between performance and efficiency.

5.2. Performance Data and Analysis

5.2.1. Forecast Data Visualization

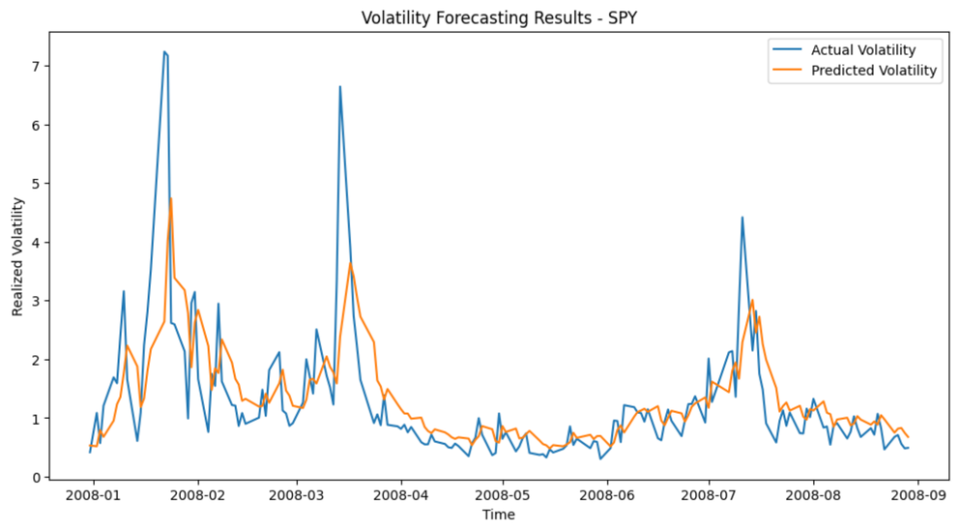

Figure 2: SPY out-of-sample forecast result plot using Realized-GARCH

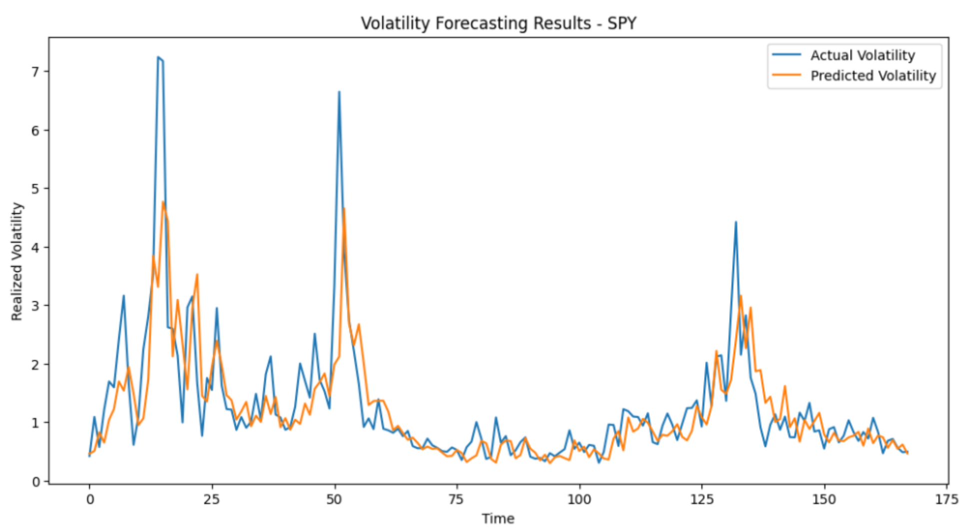

Figure 3: SPY out-of-sample forecast result plot using TCN

5.2.2. Performance Measures

	Realized GARCH			TCN			
Ticker	MSE	MAE	RMSE	MSE	MAE	RMSE	RatioMSE
AAPL	5.85	2.29	2.42	**0.94**	**0.48**	**0.97**	0.16
BIIB	8.76	2.70	2.96	**2.95**	**0.98**	**1.72**	0.34
ADP	1.82	1.12	1.35	**0.26**	**0.35**	**0.51**	0.14
FAST	3.04	1.49	1.74	**1.29**	**0.67**	**1.13**	0.42
EA	3.92	1.62	1.98	**2.10**	**0.66**	**1.45**	0.54
DISCK	3.07	1.46	1.75	**2.17**	**0.92**	**1.47**	0.71
CERN	2.55	1.36	1.60	**1.44**	**0.73**	**1.20**	0.56
BIDU	56.71	6.22	7.53	**3.25**	**1.04**	**1.80**	0.06
VOD	3.15	1.70	1.77	**0.40**	**0.35**	**0.63**	0.13
PAYX	0.84	0.83	0.92	**0.36**	**0.35**	**0.60**	0.43
ATVI	7.50	2.46	2.74	**2.79**	**0.96**	**1.67**	0.37
SBUX	7.42	2.43	2.72	**0.33**	**0.37**	**0.58**	0.04
LBTYK	4.06	1.79	2.01	**1.57**	**0.79**	**1.25**	0.39
ADSK	5.69	2.07	2.39	**2.80**	**0.90**	**1.67**	0.49
DISCA	3.22	1.43	1.79	**2.30**	**0.94**	**1.52**	0.71
DISH	4.02	1.79	2.00	**2.35**	**1.07**	**1.53**	0.58
MYL	11.89	2.73	3.45	**7.63**	**1.48**	**2.76**	0.64
SIRI	2.24	1.37	1.50	**1.06**	**0.63**	**1.03**	0.47
PCLN	2.79	1.47	1.67	**1.07**	**0.50**	**1.03**	0.38
CTSH	5.77	2.23	2.40	**0.68**	**0.43**	**0.82**	0.12

Table 1: Performance Measurements for Realized-GARCH and TCN on NDX Data

	Realized GARCH			TCN			
Ticker	MSE	MAE	RMSE	MSE	MAE	RMSE	RatioMSE
XOM	2.69	0.87	1.64	**1.65**	**0.79**	**1.29**	0.62
INTC	3.96	1.36	1.99	**3.44**	**1.27**	**1.85**	0.87
T	5.13	1.29	2.26	**4.49**	**1.26**	**2.12**	0.87
BAC	58.31	**3.75**	7.64	**52.04**	3.88	**7.21**	0.89
PG	0.63	0.44	0.79	**0.44**	**0.42**	**0.66**	0.69
SPY	0.64	0.49	0.80	**0.53**	**0.42**	**0.73**	0.83
AXP	17.36	**2.51**	4.17	**16.21**	2.58	**4.03**	0.93
MMM	1.25	**0.58**	1.12	**1.00**	0.61	**1.00**	0.80
DD	7.52	1.12	2.74	**4.80**	**1.03**	**2.19**	0.64
IBM	1.48	0.76	1.22	**1.44**	**0.73**	**1.20**	0.97
CAT	3.32	**1.08**	1.82	**3.08**	1.09	**1.75**	0.93
MRK	29.91	**1.70**	5.47	**25.92**	2.08	**5.09**	0.87
GE	3.16	**0.98**	1.78	**2.69**	0.98	**1.64**	0.85

GM	**117.28**	**5.83**	**10.83**	176.55	7.02	13.29	1.51
AA	12.93	2.10	3.60	**9.37**	**2.01**	**3.06**	0.72
JNJ	0.56	**0.32**	0.75	**0.48**	0.36	**0.69**	0.86
DIS	2.35	0.81	1.53	**1.94**	**0.75**	**1.39**	0.83
CVX	3.96	0.99	1.99	**2.30**	**0.92**	**1.52**	0.58
HD	11.85	1.96	3.44	**9.93**	**1.93**	**3.15**	0.84
KO	1.10	**0.54**	1.05	**0.91**	0.55	**0.95**	0.83

Table 2: Performance Measurements for Realized-GARCH and TCN on DJIA Data

5.2.3. Performance Measures Analysis

We randomly selected 20 different assets from both the NDX dataset and the DJIA dataset to train and predict their realized volatility with a forecast horizon of 1 day. In total, we used 40 different assets to measure and compare the performances of both univariates Realized-GARCH models and univariate TCN models by measuring their MSE, MAE, and RMSE. The resulting measures are shown in Table 1 and Table 2, both of which indicate that the univariate TCN models generally perform better than their univariate Realized-GARCH counterparts. On average, the RatioMSE, which is calculated as

$$RatioMSE = \frac{MSE_{TCN}}{MSE_{RealizedGARCH}}$$

is lower than 1 for all assets except one, indicating that TCN resulted in less error compared to Realized-GARCH for most time series used for training and prediction.

As can be seen in a few instances in Table 2, our Realized-GARCH model does perform better in terms of MAE despite having a higher MSE, implying that there are some cases where certain errors are amplified due to squaring. We hypothesize that this phenomenon is caused by the fact that Realized-GARCH will underestimate or overestimate the volatility during and after price shocks because it’s slower to adjust to rapid changes, but at the same time, it provides comparatively more stable and accurate predictions when the volatility is in low regime.

5.2.4. Model Performance Comparison using the Diebold-Mariano Test

NDX Dataset			DJIA Dataset		
ticker	DM	p_value	ticker	DM	p_value
AAPL	-12.77	0.0000	XOM	-0.36	0.3589
BIIB	-11.26	0.0000	INTC	-0.27	0.3940
ADP	-8.01	0.0000	T	-0.14	0.4455
FAST	-9.16	0.0000	BAC	-0.49	0.3109
EA	-8.81	0.0000	PG	-1.82	0.0350

DISCK	-5.27	0.0000	SPY	-0.71	0.2399
CERN	-7.30	0.0000	AXP	-0.06	0.4746
BIDU	-8.64	0.0000	MMM	1.68	0.9520
VOD	-17.78	0.0000	DD	-0.78	0.2180
PAYX	-8.87	0.0000	IBM	0.96	0.8320
ATVI	-9.12	0.0000	CAT	0.43	0.6644
SBUX	-12.61	0.0000	MRK	1.56	0.9400
LBTYK	-9.13	0.0000	GE	-0.62	0.2680
ADSK	-8.29	0.0000	GM	3.06	0.9990
DISCA	-4.41	0.0000	AA	-1.43	0.0780
DISH	-5.69	0.0000	JNJ	1.92	0.9720
MYL	-2.76	0.0031	DIS	-0.03	0.4897
SIRI	-6.29	0.0000	CVX	-0.83	0.2050
PCLN	-10.14	0.0000	HD	-0.52	0.3030
CTSH	-15.56	0.0000	KO	0.09	0.5348

Table 3: Diebold-Mariano statistic for the selected dataset for TCN against R-GARCH. p-values with a significance level of 1%, 5%, and 10% are highlighted in green, light green, and lighter green, respectively

The Diebold-Mariano statistics, used to compare the predictive accuracy of two forecasting models, are presented in Table 3. The test is performed on all 40 assets to compare the performance of the univariate TCN models with their corresponding R-GARCH models. The null hypothesis states that there is no difference in the predictive accuracy between the R-GARCH and TCN models. The statistics are separated based on the dataset. A clear distinction emerges: all assets in the NDX dataset have p-values below 1%, corresponding to confidence levels of at least 99%, strongly rejecting the null hypothesis and indicating that TCN significantly outperforms R-GARCH.

In contrast, only two assets in the DJIA dataset have p-values below 10%, corresponding to confidence levels of at least 90%. Therefore, the null hypothesis cannot be rejected for most assets in this dataset. This distinction can be attributed to the difference in the noise level of the realized measure of the DJIA dataset compared with NDX. In this case, the realized measure exhibits a more stable behavior, suggesting that the GARCH model is not robust in front of noisy data.

Since the realized measure statistic was already computed beforehand and used as input in this study, it is not possible to address the issue, but it is possible to suggest that a different kernel method was implemented in both datasets, leading to a dissimilar result.

5.2.5. Computational Runtime Comparison

	Realized-GARCH	TCN
Average Training Time (s)	10.39	44.05

Average Prediction Time (s)	0.01	1.12
StDev of Training Time (s)	1.29	13.34
StDev of Prediction Time (s)	0.0026	0.2500

Table 4: Runtime for Training and Predicting (Sample Size = 20)

Another comparison metric is the two model's respective computational time performance. As can be seen in the data shown in Table 4, the Realized-GARCH model generally performs faster for both the training and forecasting steps using the same hardware. However, this difference in time used for training and forecasting can be minimized by using dedicated hardware, such as a GPU or TPU, for the TCN model, as Realized-GARCH, unlike TCN, is a largely sequential model that typically does not benefit significantly from parallelization.

6. Conclusions

The findings from our research support our hypothesis that Temporal Convolutional Networks (TCNs) would outperform Realized GARCH models in forecasting financial market volatility. Specifically, our results demonstrate that for 39 out of 40 cases, TCNs outperformed the Realized GARCH model regarding predictive accuracy, as measured by the Mean Squared Error (MSE). This consistent performance advantage validates our initial hypothesis that TCNs, due to their architectural benefits and ability to model long-range dependencies, offer superior predictive capabilities for volatility forecasting.

The robustness of the TCN model was particularly evident across a diverse set of assets, indicating its adaptability and reliability in different market conditions. The lower MSE values observed in TCNs compared to Realized GARCH for most stocks imply that TCNs can capture the complex, nonlinear dynamics inherent in financial time series more effectively than traditional econometric models. Furthermore, TCNs proved especially adept at handling periods of heightened market volatility, which aligns with our hypothesis that TCNs' advanced feature-capturing capabilities would better accommodate abrupt changes in volatility.

According to the outcome of the Diebold-Mariano tests, TCN almost certainly provides better accuracy compared to the Realized-GARCH model when only the NDX dataset is considered. However, when considering the dataset from DJIA, this certainty is weaker and suggests that the difference between the two models might not be as prominent in certain scenarios. This is likely due to the differences in noise levels caused by market crises or different kernel methods used when computing the realized measure time series data. However, for the DJIA dataset, the TCN models still generally outperform the Realized-GARCH models in terms of MSE.

In terms of model accuracy, TCNs generally achieved lower MSE and RMSE values. However, Realized-GARCH showed superior performance on the MAE metric for certain stocks, indicating that GARCH may be more robust in predicting small, consistent deviations in volatility. This mixed performance suggests that while TCNs have strong potential in high-volatility environments, Realized-GARCH may still hold value in scenarios where computational efficiency or lower absolute prediction error is prioritized.

Regarding computational time, TCN performed slower than Realized-GARCH when executed on the same hardware. However, this difference can be minimized with hardware acceleration because of TCN's highly parallelizable architecture.

In conclusion, the consistent success of TCNs in 39 out of 40 cases confirms our hypothesis that TCNs are a superior choice for volatility forecasting in most scenarios. Their effectiveness in handling high volatility and complex temporal dependencies positions them as a valuable tool in financial risk management.

7. References

7.1. Selected Paper

Petrozziello, A., Troiano, L., Serra, A., Jordanov, I., Storti, G., Tagliaferri, R., & La Rocca, M. (2022). Deep learning for volatility forecasting in asset management. Soft Computing, 26(17), 8553–8574. https://doi.org/10.1007/s00500-022-07161-1

7.2.Other References

Barndorff-Nielsen, O. E., Hansen, P. R., Lunde, A., & Shephard, N. (2009). Realized kernels in practice: trades and quotes. The Econometrics Journal, 12(3), C1–C32.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007.

Engle, R. F., & Gallo, G. M. (2006). A multiple indicators model for volatility using intra-daily data. Journal of Econometrics, 131(1–2), 3–27.

Hansen, P. R., Huang, Z., & Shek, H. H. (2012). Realized GARCH: A joint model for returns and realized measures of volatility. Journal of Applied Econometrics, 27(6), 877–906.

Ncume, V., Van Zyl, T. L., & Paskaramoorthy, A. (2022). Volatility forecasting using deep learning and sentiment analysis. arXiv preprint arXiv:2210.12464. https://arxiv.org/abs/2210.12464

Zhang, C.-X., Li, J., Huang, X.-F., Zhang, J.-S., & Huang, H.-C. (2022). Forecasting stock volatility and value-at-risk based on temporal convolutional networks. *Expert Systems with Applications, 207*, 117951. https://doi.org/10.1016/j.eswa.2022.117951