VESTIGIUM

CAPITAL

The Big Wall(Street)

1. Introduction

Artificial intelligence is already changing our lives and it´ll revolutionize every single industry soon, if it hasn´t done it yet. The main reason behind this is that computers can handle more information, process it faster and do it cheaper than any human.

From an economic perspective, this will allow customers to enjoy better, more personalized, and cheaper products and services. But from a social perspective, the impact can be even greater, allowing mayor breakthroughs in science, medicine, climate change mitigation, reducing inequality and much more.

This is the promise of artificial intelligence as a technology. But obviously there are some “walls” or challenges that must be solved first, including the need for new regulation, ethical as well as data privacy and technical issues.

At Vestigium Capital we have been developing AI systems that can design personalized and actively managed stock portfolios. In other words, we use AI to make financial markets accessible to everyone.

During the development of our systems, we faced different challenges but perhaps the biggest one was that stock markets are unpredictable. They respond to what in statistics is called a random walk, being randomness the key feature.

Thanks to AI, and assuming that realistic synthetic simulations can be generated, algorithms can be trained in many different scenarios in order to maximize the expected return of your investment strategies.

The problem is that current available public techniques for simulating the movement of a stock, or any other financial asset, are highly unrealistic. Leaving us with only two options for training any algorithm:

1. Rely on historical data: causing a clear problem of **overfitting** and **data scarcity.**
2. Rely on unrealistic simulation: causing a problem of **data bias**, algorithms are only as good as the data that they are trained on.

After facing this situation, we did what we know best: create an AI model to generate realistic and synthetic market data. In the following pages we will describe how it was made and how it works. Furthermore, we will make the code public hoping that it can help other developers and financial professionals.

What is the most important problem for applying AI to wealth management?

"It's difficult to say which problem is the most important as each of the challenges for applying AI to wealth management has its own significance. However, data quality is a critical issue for AI implementation in wealth management. High-quality data is essential for AI algorithms to make accurate predictions and recommendations. Inaccurate or incomplete data can lead to incorrect recommendations, which can result in significant financial losses for clients. Additionally, poor data quality can lead to compliance risks and legal issues. Therefore, addressing data quality is crucial for the successful implementation of AI in wealth management."

Chat GPT-4

2. The State of the Art

Before explaining our model, it is important to analyze and demonstrate why the current available models for generating synthetic stock prices are not good enough for training an AI system.

We are going to focus on the best two models available right now: **Geometric Brownian Motion** and **TIMEGAN.** Even though both may work relatively well and bring some advantages, these models lack what we consider important issues that will be detailed later (Kinlay, 2022).

2.1. The Experiment

We will take an empirical approach and both models under a real-world experiment. We will generate synthetic daily prices for the company Alphabet Inc (ticker: GOOGL) in a Open/High/Low/Close format (O/H/L/C).

In order to conclude if the generated data can be used for training AI algorithms, we are going to closely look at the main characteristics defining O/H/L/C prices:

- **Internal Consistency:**

$$H_T \geq Max(\, O_t, L_{t,}C_t)$$

$$L_T \leq Min(\, O_t, H_{t,}C_t)$$

- **Probability Density Function of Daily Fluctuations:** reflects daily fluctuation (Df) changes during the trading session.

$$Df = \frac{C_t}{O_t} - 1$$

- **Probability Density Function of Intra-day Fluctuations:** reflects intra-day fluctuation (IntraDf) spread between the higher and lower prices during trading session.

$$IntraDf = \frac{(H_t - L_t)}{O_t} - 1$$

- **Probability Density Function of Inter-day Fluctuations:** reflects inter-day fluctuations (InterDF) changes during the after-hour trading session (post/pre-market).

$$InterDf = \frac{O_t}{C_{t-1}} - 1$$

- **Clustering Volatility:** we expect our simulations to have a non-constant volatility as it is observed in real market data.

2.2. Geometric Brownian Motion (GBM)

Currently, this is the most used model for simulating market data. It is a stochastic approach based on the Brownian movement of particles in a fluid observed by Robert Brown in 1827. The model assumptions are determined by the characteristics that define movements and basic laws of fluid mechanics, being the most important:

- **Continuous Variable:** when you move from point A to B you pass through all the points between A and B.
- Movements fluctuations follow a **normal distribution.**
- **Constant volatility**

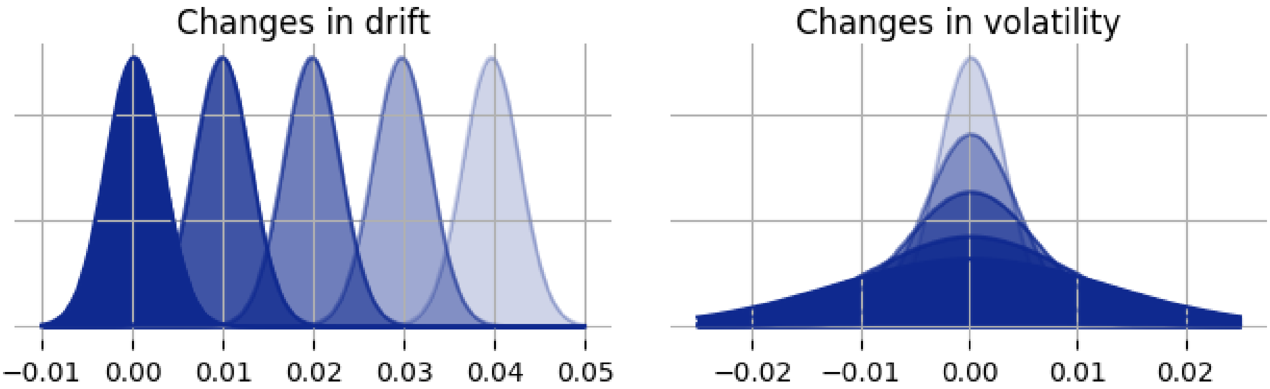
2.2.1. GBM Parametrization

The GBM model parametrization is relatively simple as we only need to adjust 2 parameters:

- **The drift** or "μ"
- **The volatility** or "σ"

The drift parameter allows us to determine the average long-term trend that we want for our simulations. While the volatility parameter allows us to determine the standard deviation of the increments between each time-step.

From a more technical perspective, we can adjust the position ("μ") and the kurtosis ("σ") of the Probability Density Function for the increments in each time-step as shown in these figures:



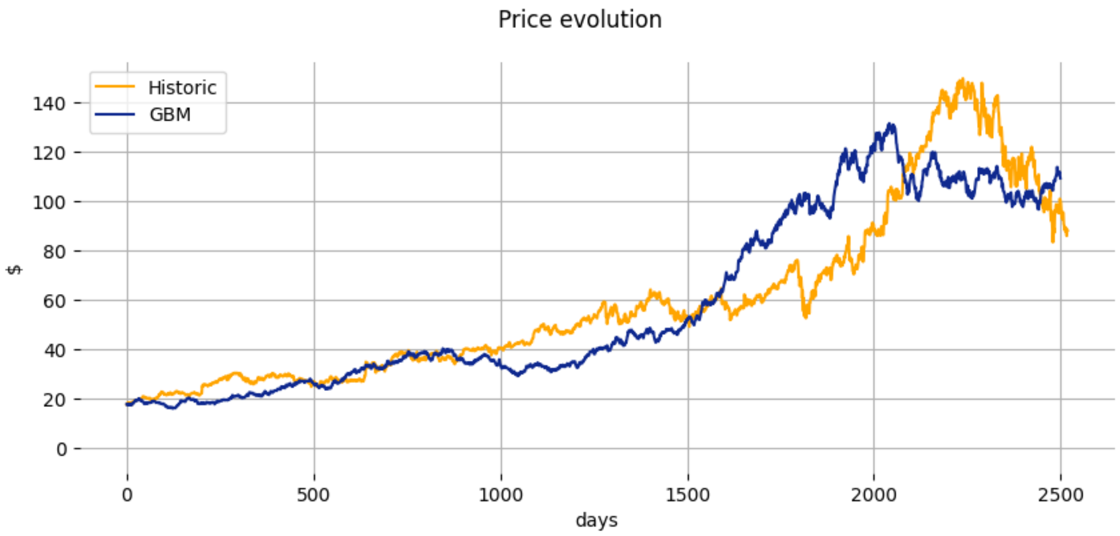
To generate daily synthetic prices in O/H/L/C format, we simulated in a higher granularity (hourly) and then calculated the High and Low values for each day. This guarantees a **100% consistency in the simulated data.**

The parameters used for this experiment were:

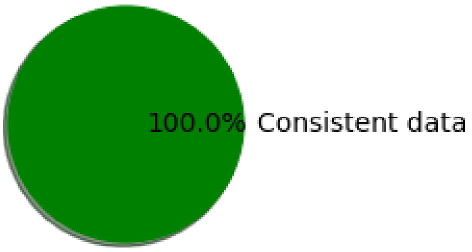
- $\mu = 0.2$
- $\sigma = 0.18$
- $Time - steps = 252 * 7 * 10$ *(252 = days, 7 = hours, 10 = years)*

2.2.2.GBM Results

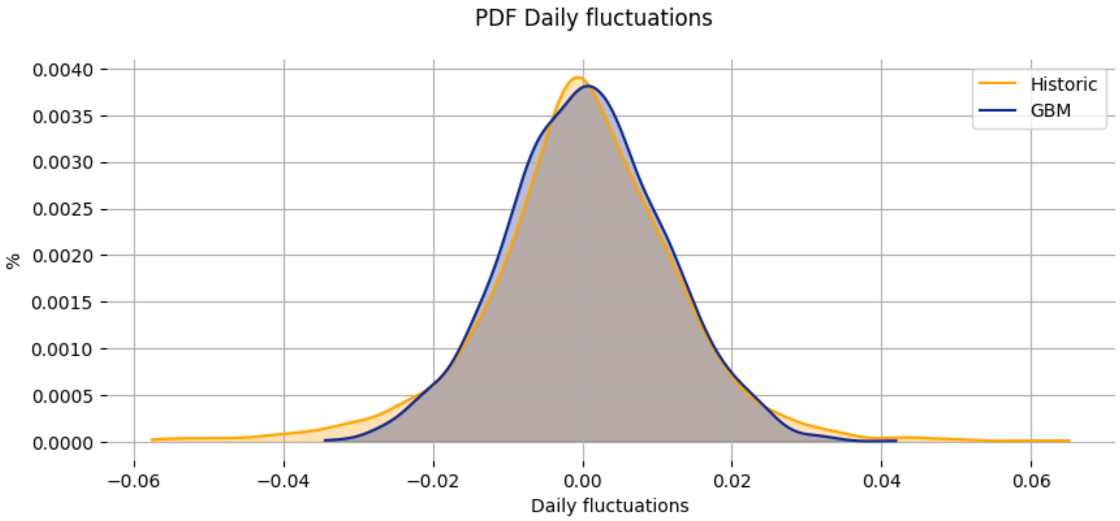
- Price Evolution



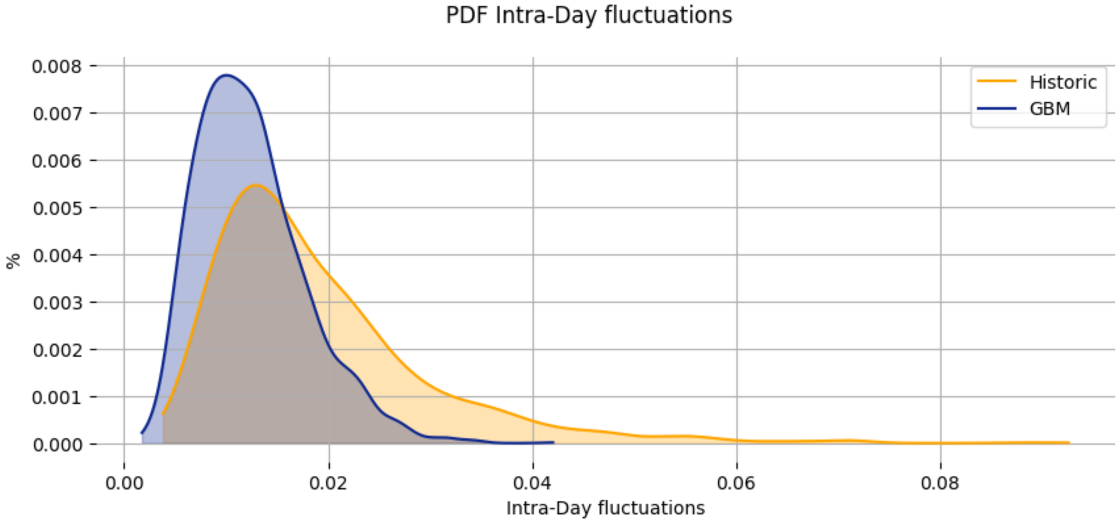
- Internal Consistency:



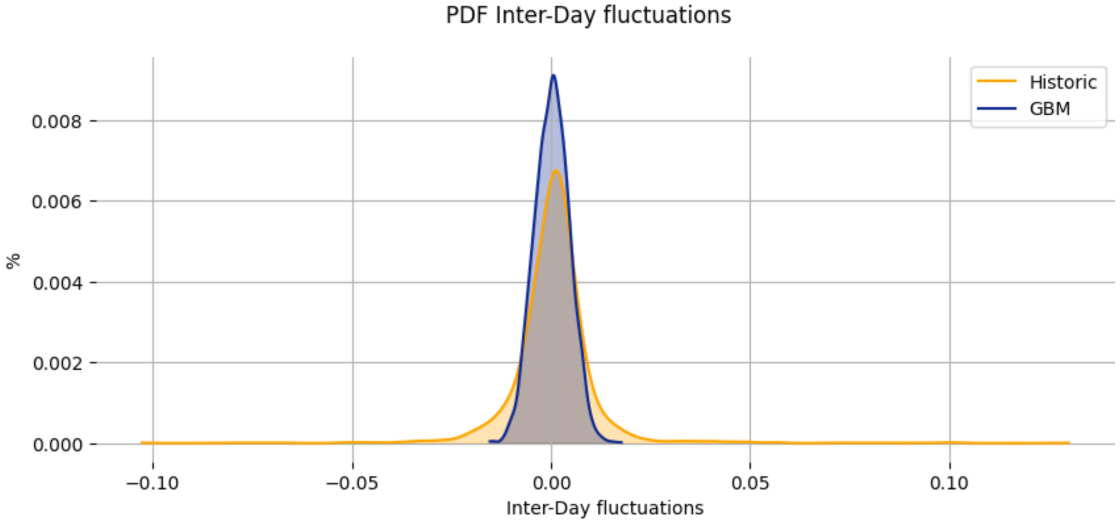
- Probability Density Function of Daily Fluctuations:



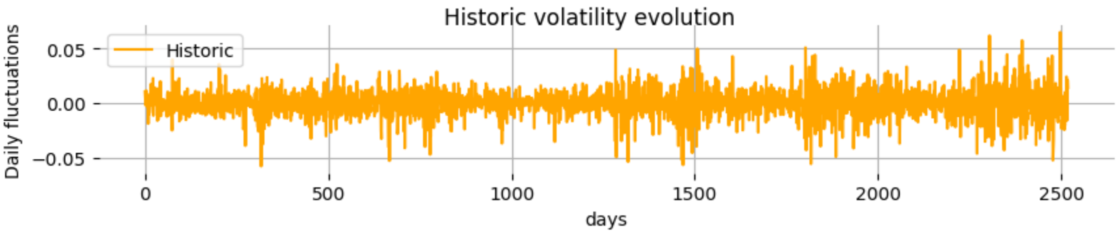
- Probability Density Function of Intra-day Fluctuations:

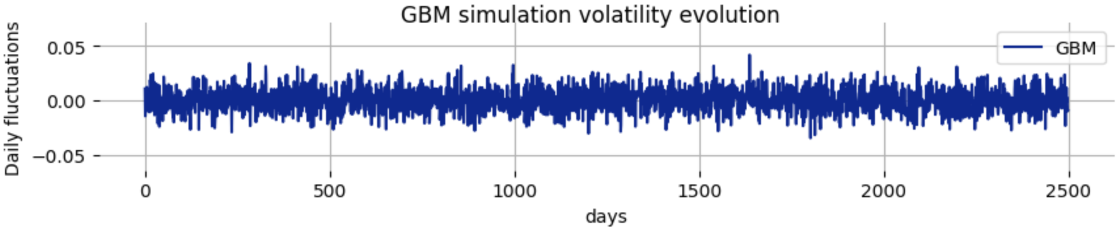

PDF Intra-Day fluctuations

- Probability Density Function of Inter-day Fluctuations:


PDF Inter-Day fluctuations

- Volatility Evolution:


Historic volatility evolution


GBM simulation volatility evolution

2.2.3. GBM Conclusions

Geometric Brownian Motion model is an easy to implement method that needs low computational resources and can generate synthetic prices in a fast way. However, results show that the simulations obtained are highly unrealistic and fail to capture the complexity of the financial markets. The most important problems are:

- **Normally distributed returns:** even though this characteristic is intrinsic to GBM model, we applied a Shapiro-Wilk normality test, and we obtained a P-value of 0.63 for simulation returns and $1.47\,e^{-21}$ for the historic returns. Proving that even when we parametrized the model to obtain the most similar PDF for the daily returns, there still are huge differences between them.
- **All time steps are considered equal:** Inter-day fluctuations are completely different to daily fluctuations, and the model cannot be adjusted to incorporate this characteristic in its simulations.
- **Constant volatility:** GBM model assumes constant volatility which is far from being true in financial markets. This problem can be solved by using an adaptation of GBM called *"Heston Model"* that defines the volatility as a random process, but parametrization process gets more complicated.

Another important consideration is that the difference between simulations and reality relies on the tails of the distributions. In other words, simulations are not incorporating the extreme events both on the negative or positive side.

2.3. TIMEGAN (Generative Adversarial Networks)

In recent years Generative Adversarial Network or GANs have had a huge impact in the generative field. This new approach improved the generation of synthetic data for many different formats (images, video, tabular data, etc.), but it could not be applied to time-series.

It was only after 2019, when Yoon *et al;* 2019 published a paper called: "Time-series Generative Adversarial Networks", where they proposed a framework for applying GANs to time-series.

2.3.1. TIMEGAN Parametrization

TIMEGAN models need to be trained before using them, and for this particular use case we will have to train a different model for every asset that we want to simulate. This is a big problem considering that training time using CPU´s configuration and for 5,000 epochs was of **9,5 hours**.

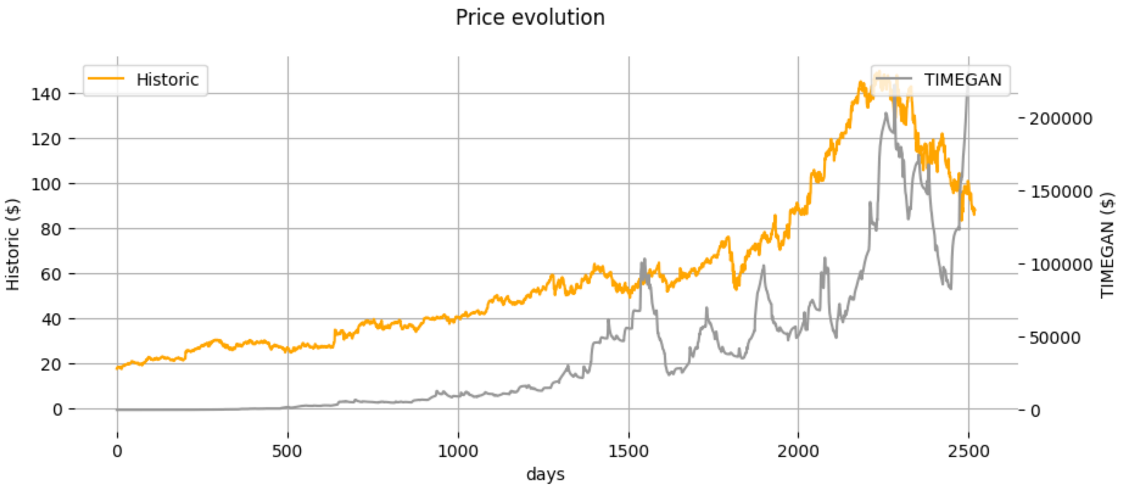
Another serious problem that we observed is that, because of the neural network characteristic, this model generates fixed dimension simulations. This dimension is determined by the parameter "*sequential length*" or "*seq_len*" that can be modified, but any changes imply a new training process.

For this experiment we used the configuration proposed by Jinsung Yoon and Daniel Jarrett, except for the epochs. Some parameters could still be readjusted. For example, the sequential length of 24 makes sense if you have hourly data but doesn't seem to have a logical explanation for simulating daily stock prices.

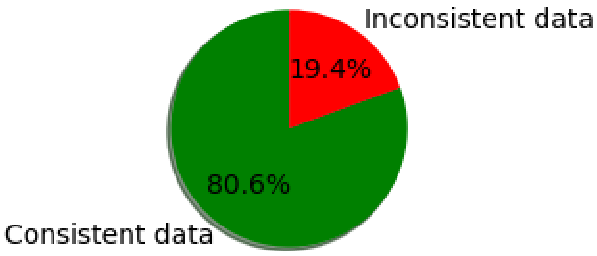
- $\mathrm{seq\_len} = 24$
- $n\_seq = 6$
- $learning\_rate = 5e - 4$
- $\log\ \_step = 100$
- $noise\_dim = 32$
- $dim = 128$
- $batch\_size = 128$

2.3.2. TIMEGAN Results

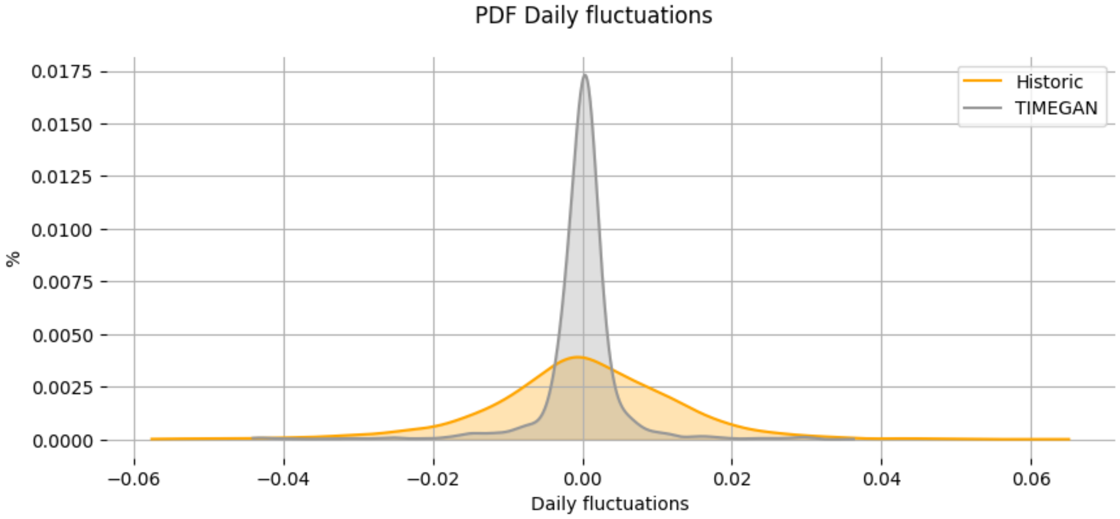
- Price Evolution:



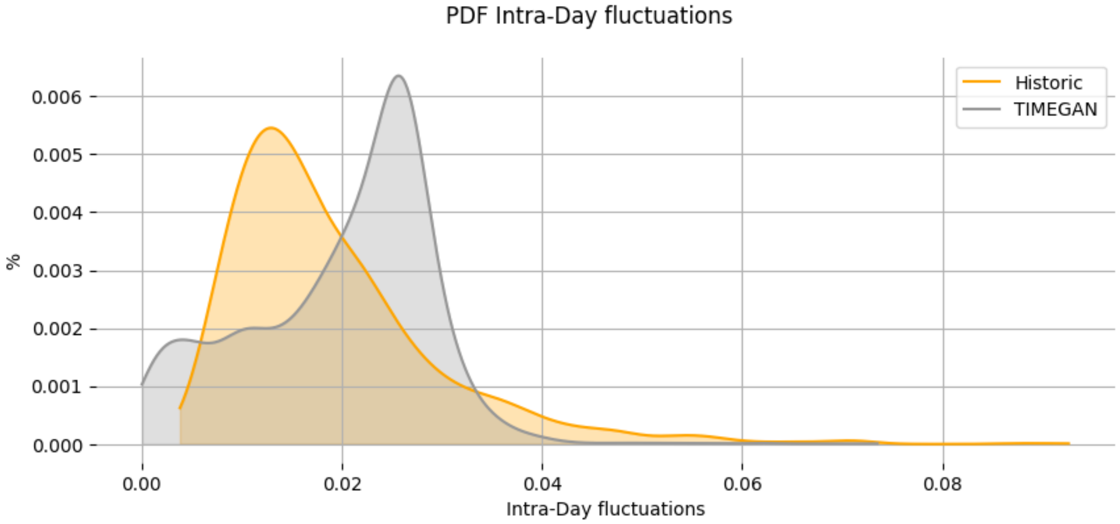
- Internal Consistency:



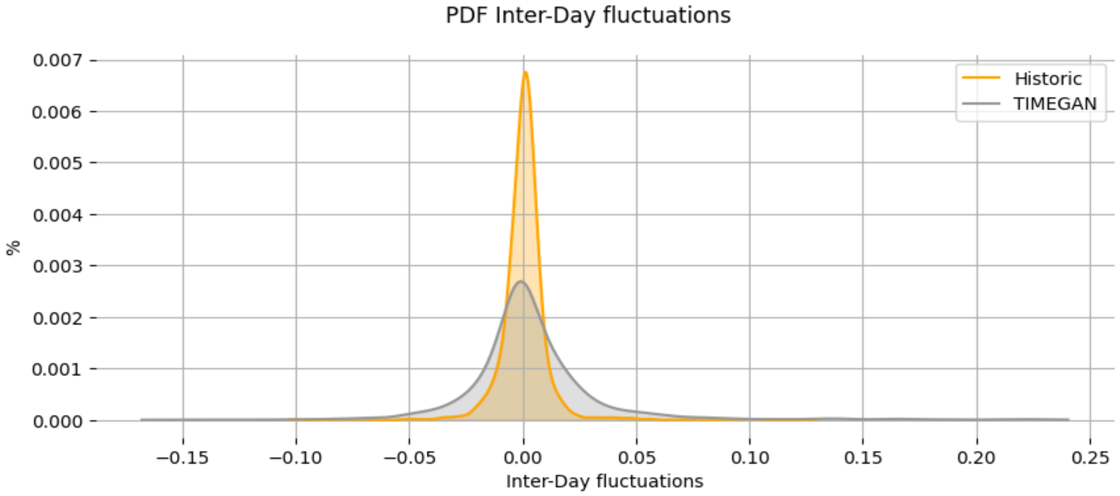
- Probability Density Function of Daily Fluctuations:



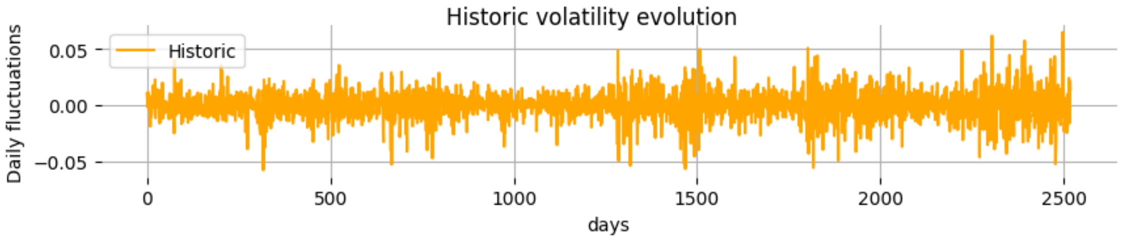
- Probability Density Function of Intra-day Fluctuations:

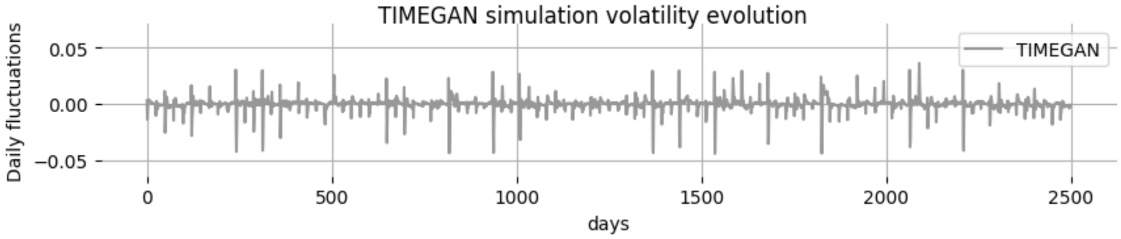


- Probability Density Function of Inter-day Fluctuations



- Volatility Evolution





2.3.3. TIMEGAN Conclusions

The results obtained using the TIMEGAN framework for generating synthetic prices are far from being realistic. But, even in the hypothetical scenario that could improve the results by reducing training times or by changes in parametrization, other important problems inherent to the model would still remain:

- **Inefficiency:** we would have to train one model for each financial asset.
- **Fixed length simulations:** TIMAGAN can only generate simulations of the same length as the sequential length parameter used in the training process.
- **Black-box model:** we cannot adapt the output of the model without retraining the model with different data.
- Need for **high computational resources.**

3. A Novel Approach for Generating Synthetic Market Data: The VC Method

After considering the pros and cons of available models, we started the design process of a new framework for generating synthetic market data (O/H/L/C) with three main goals:

- **Efficient implementation process:** reducing parametrization and training process.
- **Improved realism of the generated data** trying to capture all the complexity of the financial markets.
- **Adjustable:** incorporating the possibility of influencing the output of the model.

The first step when it comes to generate synthetic data is to thoroughly study and understand the original data itself. In the case of stock prices in O/H/L/C format we are dealing with one of the most complex cases of data, a time series with the following characteristics:

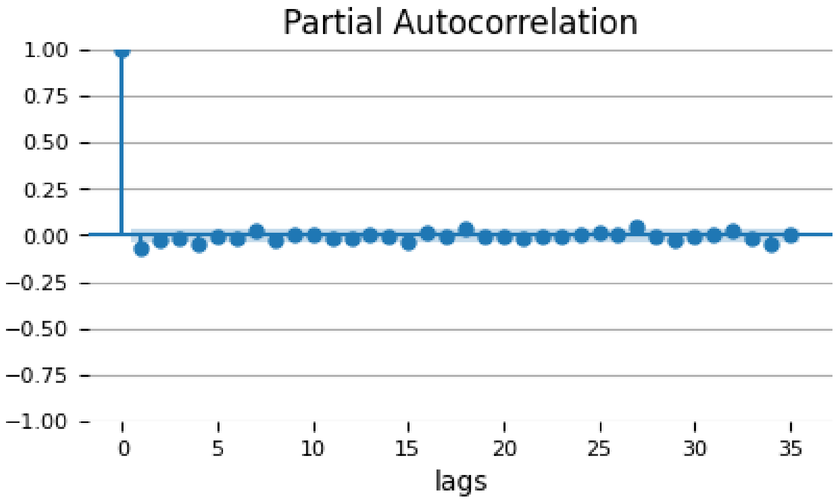
- **Differentiated time-steps:** movement of prices in after-hour or pre-market time is completely different from trading hours.
- **Non-gaussian:** fluctuations of prices **do not follow a normal distribution.**
- **Clustering volatility**
- **Non-stationary**
- **Drift**

Besides these characteristics there is one key feature that has been strongly debated among financial experts without reaching an agreement. Are price fluctuations random, or can they be predicted?

3.1. Random Walk vs Deterministic Walk

In order to answer this question, we are going to use two different techniques on the historical data for the company Alphabet Inc (12-31-2012 12-31-2022):

- **Partial autocorrelation (PACF):** measures the correlations between a time series and its own lagged values. And as we can see in the following graph, there is no significant relationship between them. As we cannot reject the null hypothesis that the Autocorrelation is equal to zero for any lagged value.



- **Randtest (python library):** a more sophisticated method that uses an exponentially decaying moment prediction to determine the net deviation between the predicted and actual elements of a sequence. Trying to find any hidden pattern in the given data. The obtained result was "True" meaning that the daily fluctuations are random with a confidence level of 96.82 %

The conclusion we reached after testing with 2 different methods is that there is no relationship between values of our time-series. In other words, the stock price fluctuations follow the characteristics of a **Random Walk**. This conclusion allows us to rule out all the autoregressive models of our analysis and treat each time step as an independent event.

3.2. Philosophy

A Random Walk is any movement process defined by a random event over a series of time steps in a limited geometrical space. In our use case we have a 2-dimensional space defined between [0 , ∞), and the number of time steps is the length of the desired simulation. So, we focused on understanding the two features that define a random event:

- **Sample space:** all the possible outcomes that can occur in each time step.
- **Probabilities:** the probability associated with each possible outcome.

The relationship between this two variables defines the **Probability Density Function or PDF** of the event. If we can estimate this function is very easy to replicate a random event and therefore generate synthetic and realistic random walks.

So, our objective was to generate simulations by decomposing O/H/L/C quotations into independent random events and calculating their probability density functions based on historical information.

3.3. Methodology

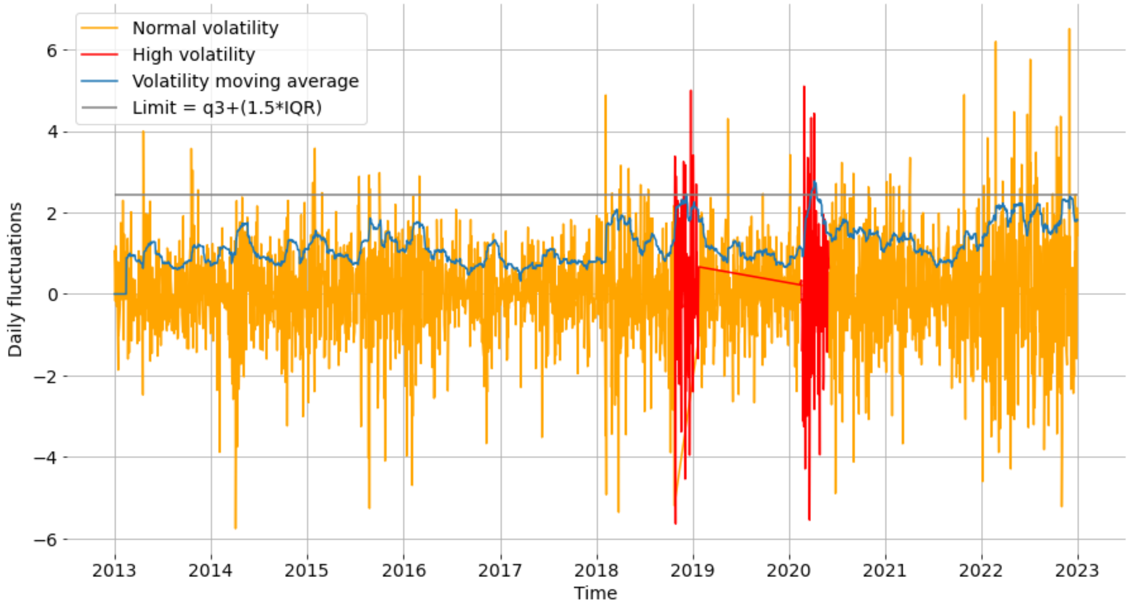
In this section we are going to describe, step by step, how this novel framework works. As in previous examples we are going to work with daily historical O/H/L/C data of Alphabet Inc.

3.3.1. Step 1: Split Sample

First, depending on the volatility levels, we divide the original sample into two different sub-set:

- **Normal volatility periods**
- **High volatility periods**

Normal and high volatility level periods are determined according to the daily volatility moving average. To identify the beginning of a high volatility period, we apply the interquertile range detection method. Once a high volatility period begins, it will remain in effect for as long as the volatility moving average does not stay below the limit for 30 straight days.



For the case of Alphabet Inc. shown in the graph above, it can be observed that, according to our standards, there were two high volatility periods.

The reason why we include step 1 as part of our methodology is the need for our simulations to take into consideration the existence of "black swan events" which significantly modify the fluctuation of prices of any asset. This is also known as clustering volatility.

3.3.2. Step 2: Decompose O/H/L/C Daily Fluctuations

The second step is to decompose the daily fluctuations into an independent stochastic process. This will incorporate into our model the difference between price fluctuations during the pre-market, trading hours an post-market.

- **Daily Fluctuations:** reflect changes during the trading session.

$$Df = \frac{C_t}{O_t} - 1$$

- **Intra-day Fluctuations:** reflect spread between the highest and lowest prices during trading session.

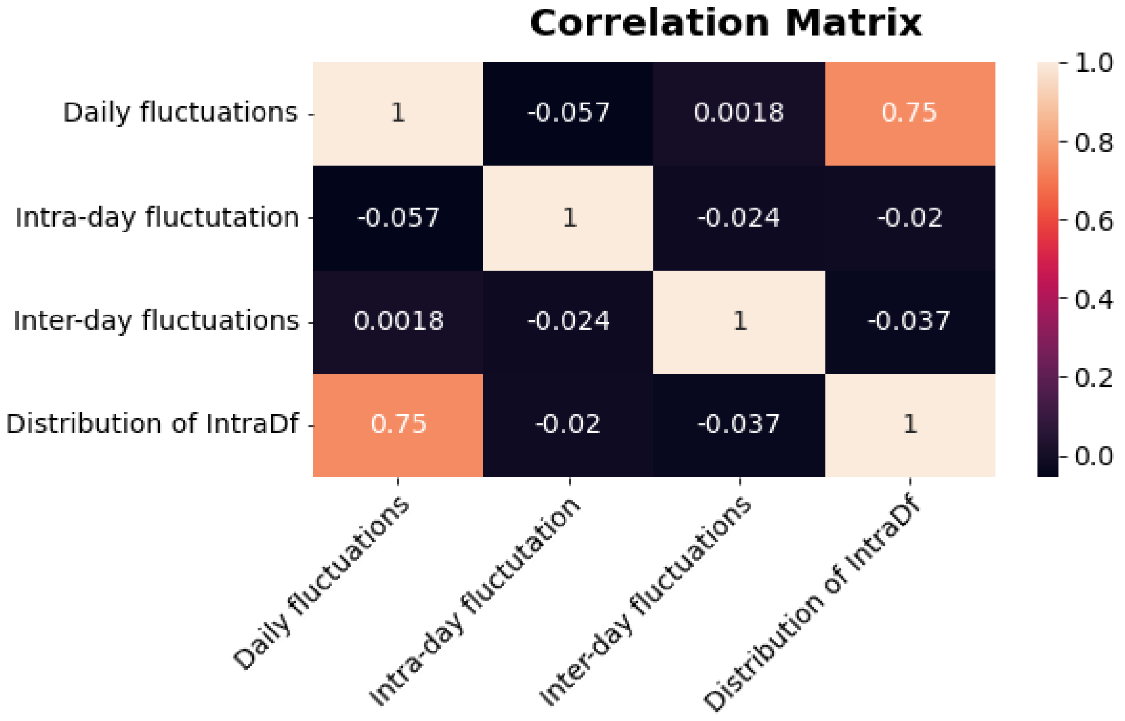
$$IntraDf = \frac{(H_t - L_t)}{O_t} - 1$$

- **Inter-day Fluctuations:** reflects changes during the after-hour trading session (post/pre-market).

$$InterDf = \frac{O_t}{C_{t-1}} - 1$$

- **Distribution of Intra-day Fluctuations:** this variable reflects what percentage of the total Intra-day fluctuations is associated with the High price.

$$Distribution\ of\ IntraDf = \frac{(H_t - O_t)}{(H_t - L_t)}$$



As we can see in the correlation matrix, we were able to decompose daily price fluctuations into three completely independent variables. But there is a **strong relationship between Daily Fluctuations and the Distribution of Intra-day**

Fluctuations. This relationship is caused by the fact that when the price of a stock rises, the intra-day variation tends to move towards the High.

To incorporate this relationship in our simulations, we decided to generate Intra-day Fluctuations using a **linear regression model** fitted with historical info. We use synthetic daily fluctuation as dependent variable, and added some white noise until a similar level of correlation was reached.

There is another important relationship within the variables that determines the consistency of the data. The **Intra-day fluctuation can never be lower than the daily fluctuation**. This condition assures that the High is always the highest and the Low is always the lowest value for any given day.

So, to incorporate this relationship and generate consistent synthetic data, we decomposed the Intra-day fluctuations in two: **Daily fluctuation** and a **Differential of Intra-day fluctuation**.

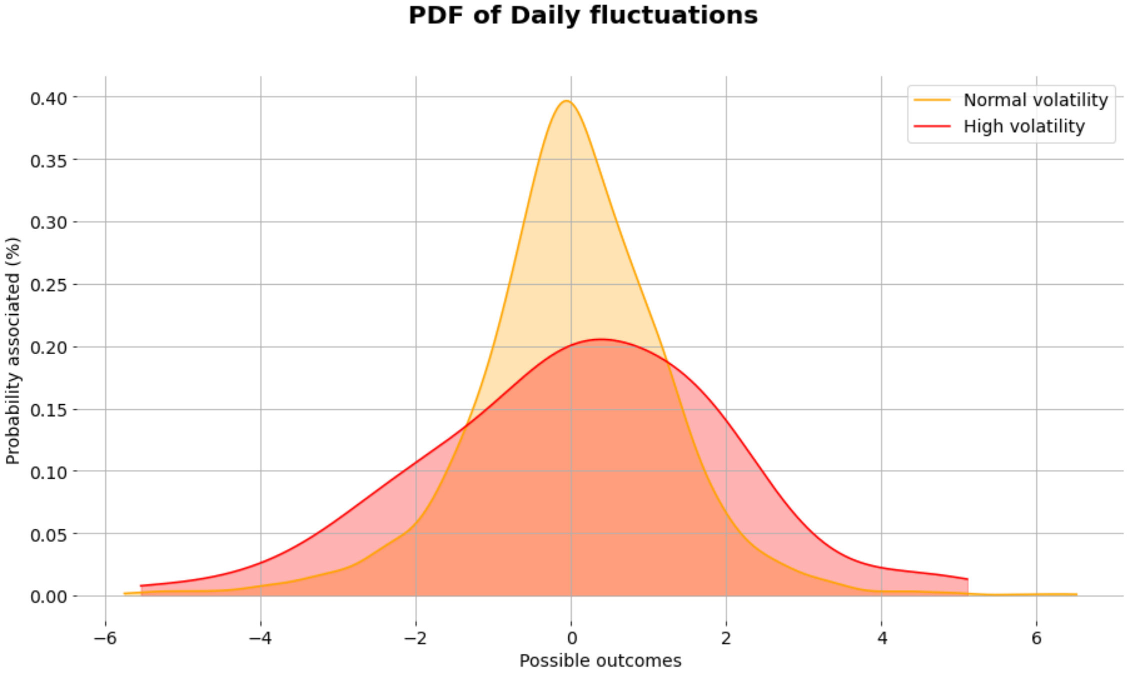
$$IntraDf_t = Df_t + Differential\ of\ IntraDf_T$$

$$Differential\ of\ IntraDf_T = IntraDf_t - Df_t$$

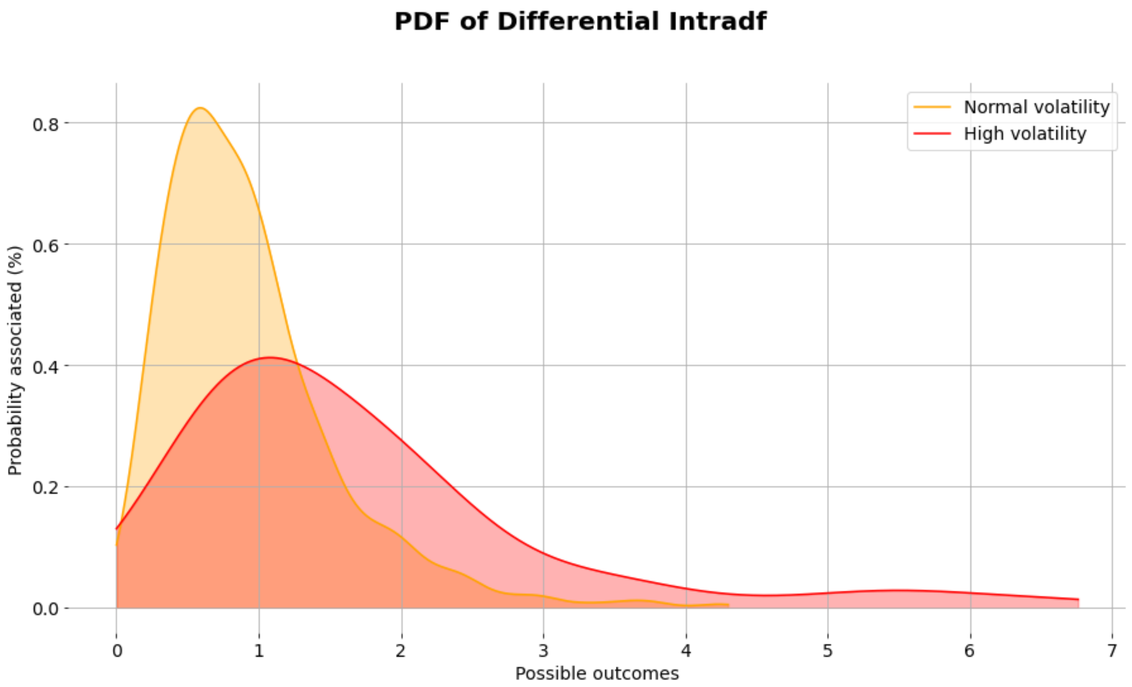
3.3.3. Step 3: Calculating Probability Density Functions (PDF)

Once we have identified the three independent variables described above, we now need to obtain their PDF. t´s important to remember that we need to calculate each PDF differentiating between normal volatility periods and high volatility periods.

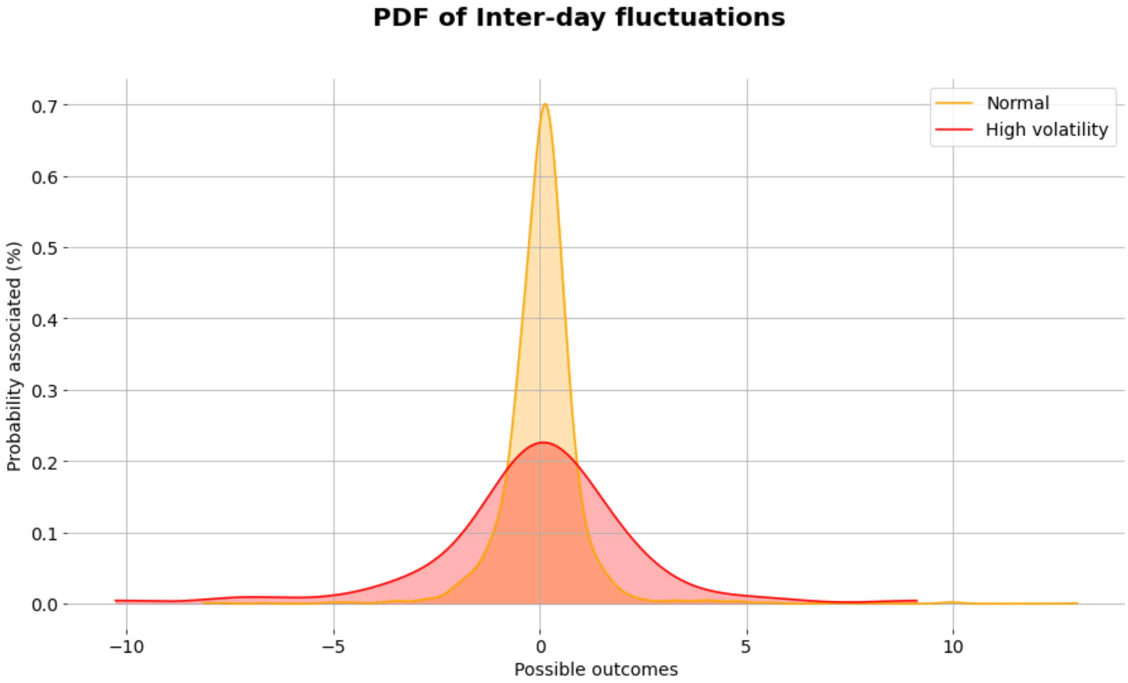
- Daily Fluctuations:



- Differential of Intra-day Fluctuations:



- Inter-day Fluctuations:



One of the biggest benefits of working with PDF´s is that we are capable of transforming our discrete historical data into a continuous variable. This means that our model learns from historical information but is able to incorporate values that have never happened in the past. We have called this **augmented bootstrapping**.

3.3.4. Generating Synthetic Data

At this point, we have all the necessary information and we are just two parameters away from generating synthetic market data:

- Initial Price
- Length of the simulation

Once we have set these parameters, the model starts a sequential process which is basically the repetition of some random and deterministic events:

1. **Determine if it´s the beginning of a high volatility period:** this is a binomial random event with two possible outcomes: true or false. If true, then a second random event takes place to determine the duration of this period based on the PDF of historical length of high volatility periods (augmented bootstrapping).
2. **Generate the daily fluctuation**: random event using the historical PDF of the normal or high volatility period according to the outcome of first event.
3. **Generate the differential intra-day fluctuation**: random event using the historical PDF of the normal or high volatility period according to the outcome of first event.
4. **Generate the inter-day fluctuation:** random event using the historical PDF of the normal or high volatility period according to the outcome of first event.
5. **Calculate the distribution of intra-day fluctuation:** deterministic event, where a linear regression model calculates the intra-day fluctuation for which the previously generated daily fluctuation is used as the dependent variable.

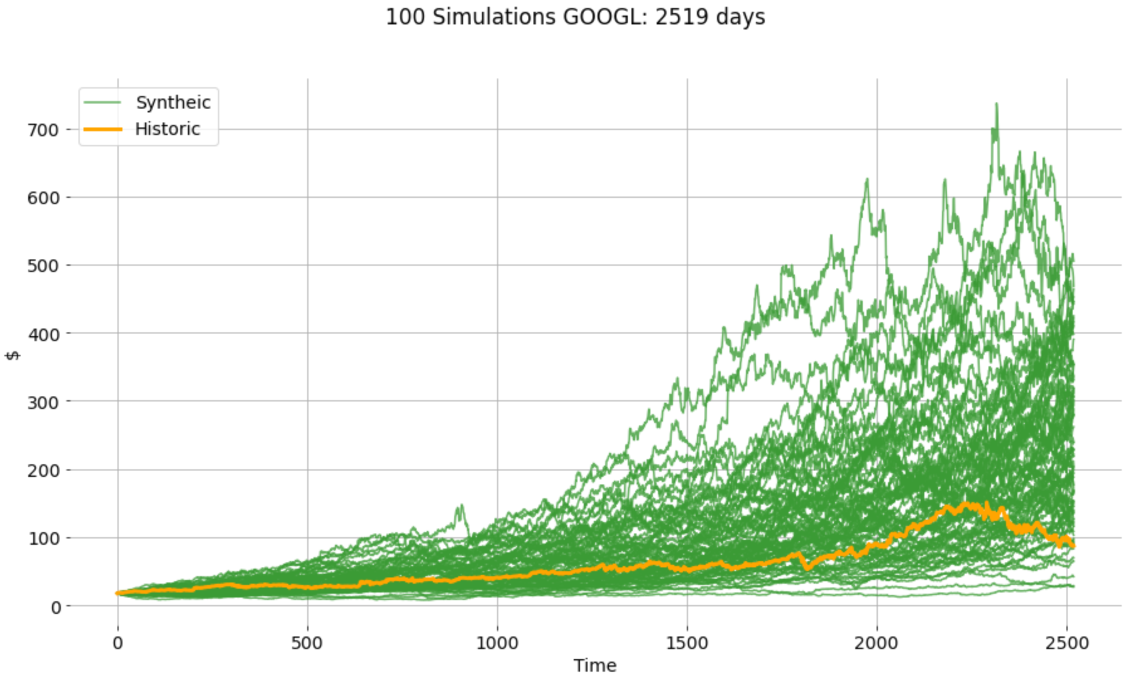
3.4. Results

Now that we understand how this new framework works, it is time to analyze the results obtained, focusing on our three main goals:

3.4.1. Efficient Implementation Process

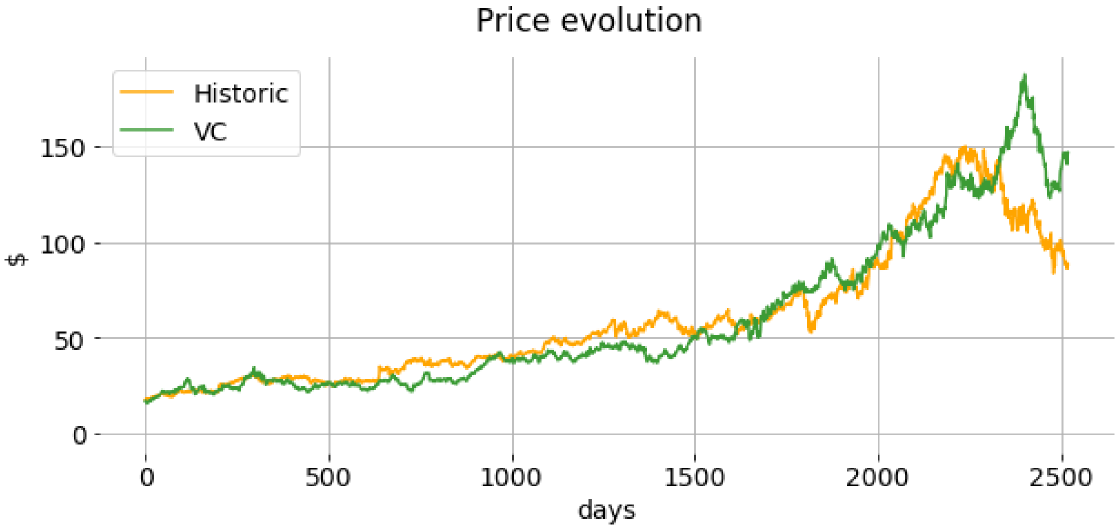
One of our main goals was to design a **fast and easy to implement** framework. As we have mentioned above the user only needs to determine two parameters: **length of the simulation** and **initial price**. Allowing anyone to use it with no need to have any financial or A.I. related knowledge.

For testing the speed of our model, we generated 100 simulations of 2519 days, making a total of 1.000 years of synthetic data. The total processing time was 70.38 seconds or **0.07038 seconds per year.**

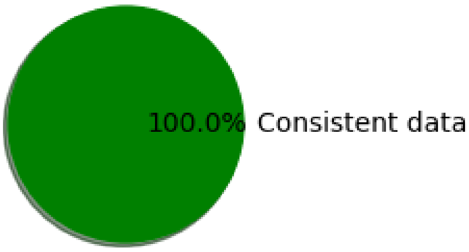


3.4.2. Improved Realism of the Generated Data

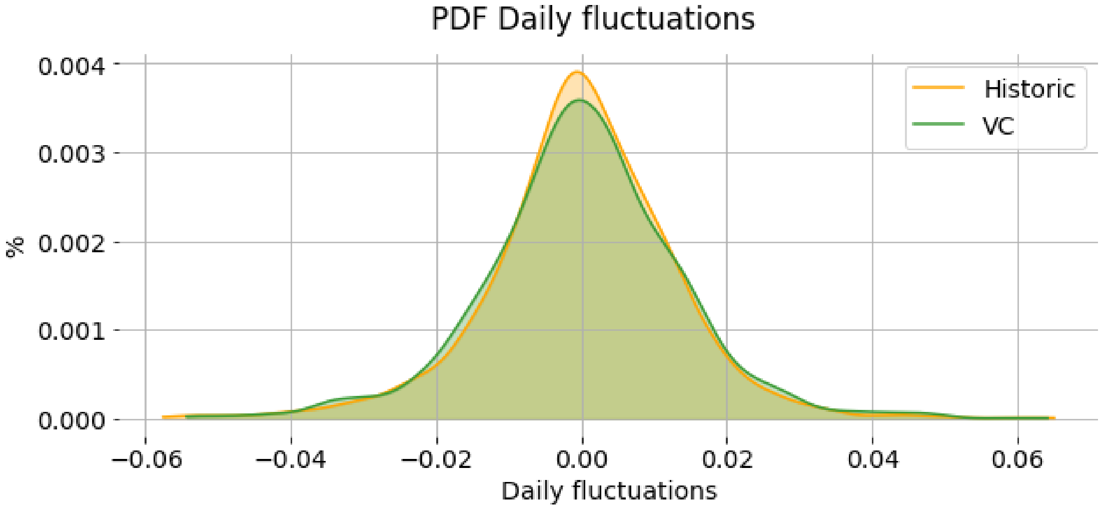
- Price Evolution:



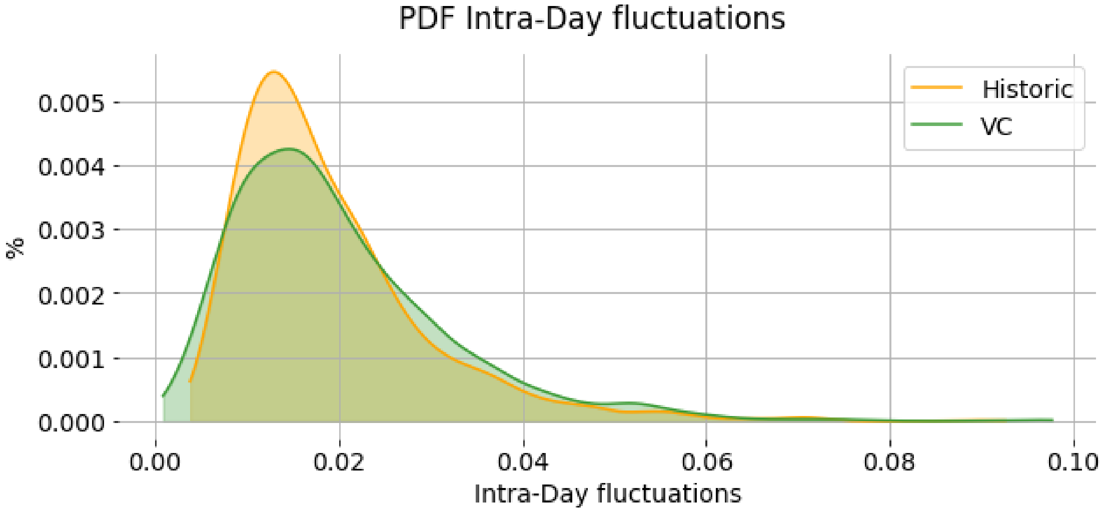
- Internal Consistency:



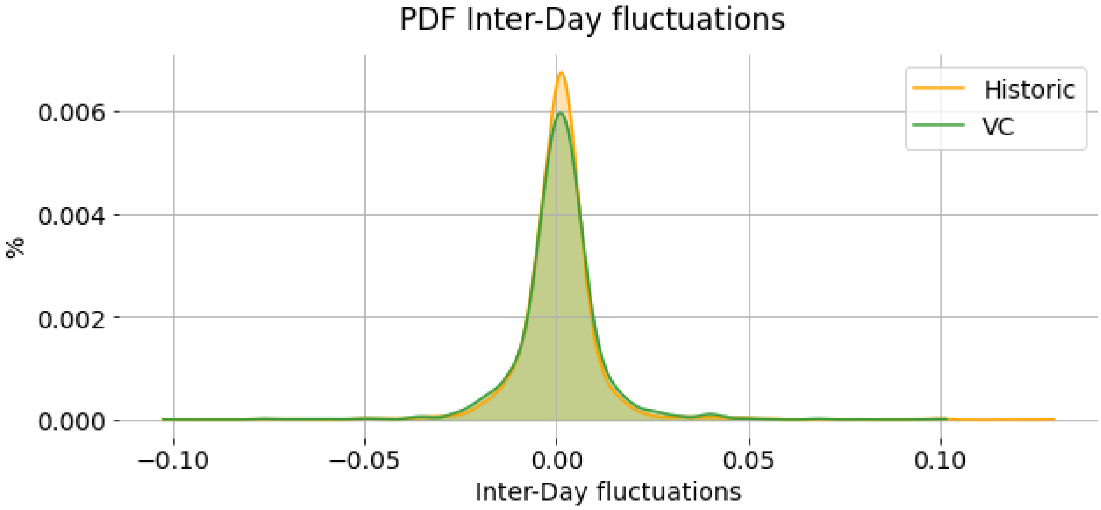
- Probability Density Function of Daily Fluctuations:



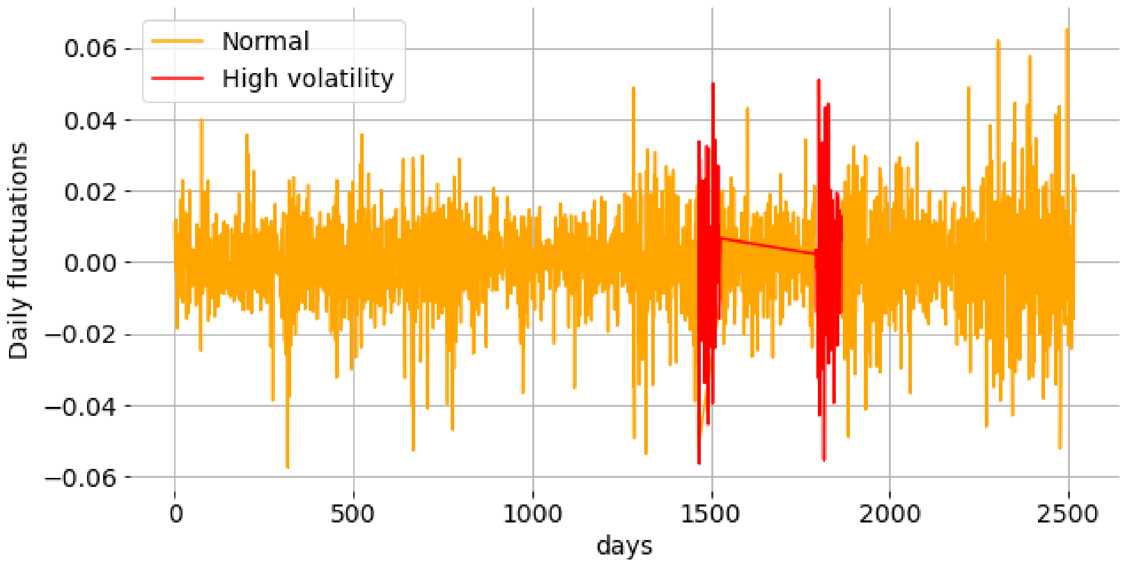
- Probability Density Function of Intra-day Fluctuations:

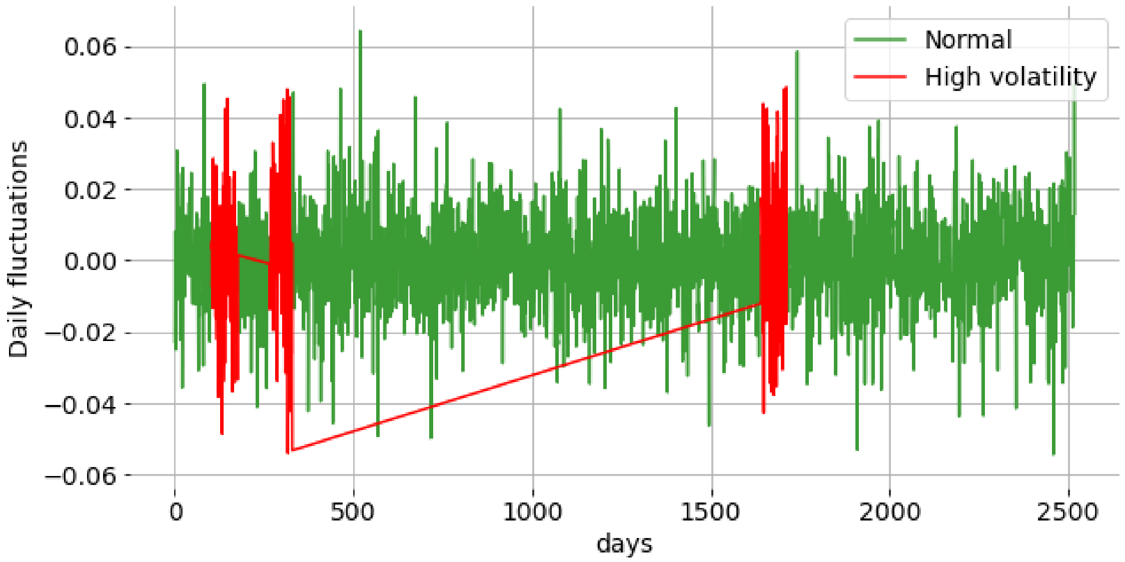


- Probability Density Function of Inter-day Fluctuations:



- Volatility Evolution

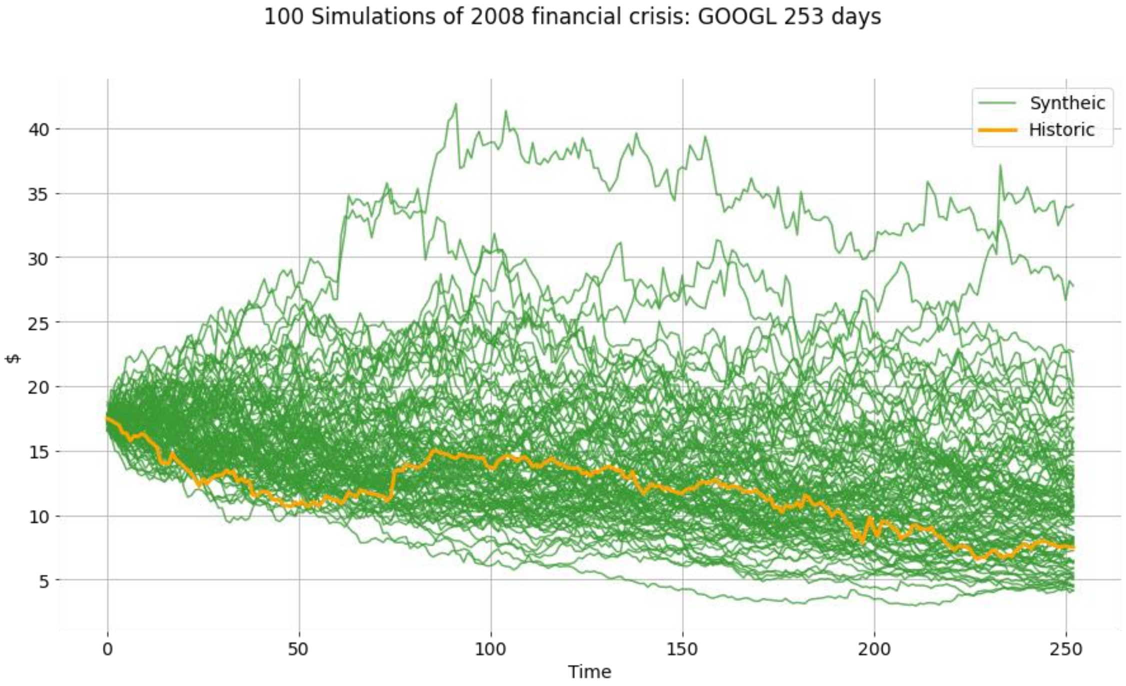

Historic volatility evolution


VC volatility evolution

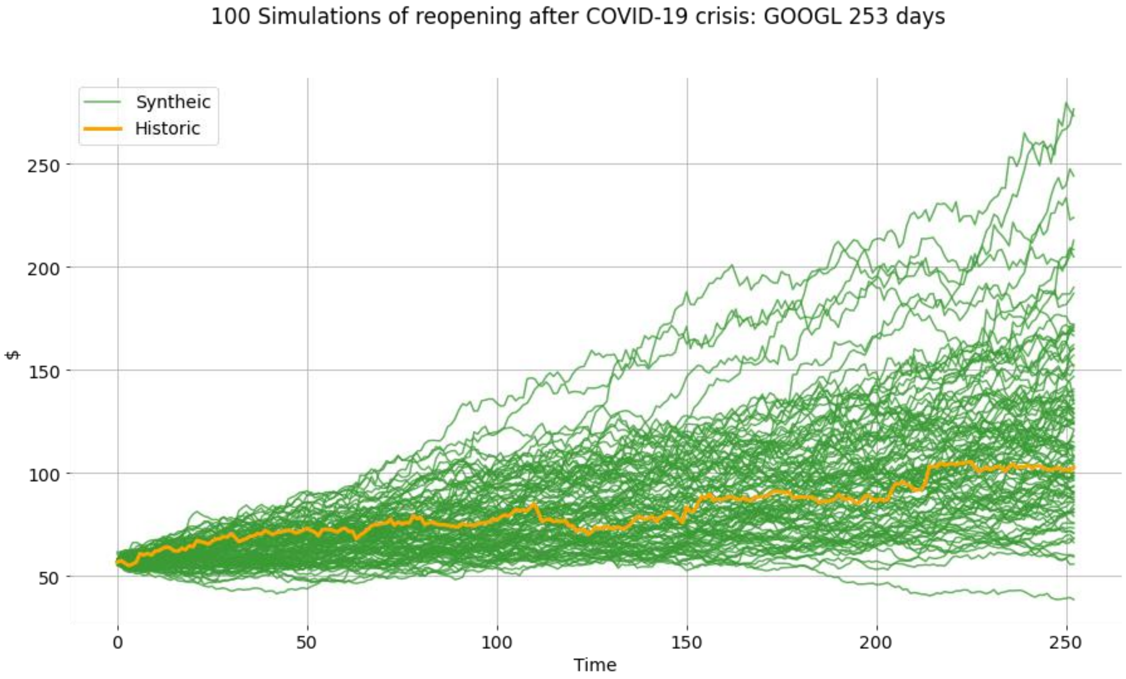
3.4.3. Adjustable

Our framework relies on the stochastic replication of random events based on historical information. So, by changing the historical period we can very easily adapt the generated data.

For example, we can generate 100 different paths of the stock Alphabet Inc., using the historical information of the 2008 financial crisis. This is a very useful feature for professionals working on areas such as risk management or stress testing.



In the following example, we used the historical information from the reopening period after the COVID-19 crisis, 2020-03-31 to 2021-03-31.



3.5. Conclusions

The biggest problem for applying machine learning and A.I. techniques to generate investment strategies is **overfitting.** This produces a lack of robustness that generates huge performance differences from training to real environments.

If we use synthetic data for the training process, we can mitigate this problem. But the available frameworks for generating synthetic data are highly unrealistic and/or inefficient.

At **Vestigium Capital** we have created **The VC Method**, a novel stochastic framework that allows us to generate infinite and realistic synthetic O/H/L/C data. This is a major breakthrough which can be applied in many different areas such as pricing derivatives, risk management or creating and training algorithms. Opening new horizons for the implementation of AI techniques in the financial industry.

- ## References:

Kinlay, J. (July 25, 2022). A New Approach to Generating Synthetic Market Data. *QUANTITATIVE RESEARCH AND TRADING.* https://jonathankinlay.com/

Yoon, J., Jarrett, D., & Van der Schaar, M. (2019). Time-series generative adversarial networks. Advances in neural information processing systems, 32.