Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título "Generation of synthetic option markets using Generative Adversarial Networks" en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2019-2020 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Ricardo Moreno Alonso Fecha: 16/ 07/ 2020

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Fernando de Meer Pardo Fecha: 17/7/2020

Generation of synthetic option markets using Generative Adversarial Networks

Comillas Pontifical University. Master in Smart Industry Final Project

Ricardo Moreno Alonso
Supervised by Fernando de Meer Pardo

Madrid, August 2020

Abstract

Options are financial securities of great importance in risk management and are widely used instruments in hedging strategies. The use of reinforcement learning in hedging is an active field of research. However, the historical data on option prices might not be sufficient to train complex machine learning models without overfitting. This motivates the creation of option market simulators, which can augment the existing data. This project consists in the creation of an option market simulator based on Generative Adversarial Networks (GANs). This work uses a similar methodology to that used in the paper *"Deep Hedging: Learning to simulate equity option markets"* written by Magnus Wiese *et al*. However, a different generating mechanism to the one used in such paper is explored. Here, option market scenarios are generated as images. The realism of the generated markets is evaluated by comparing distributional and time series statistics on the real and synthetic markets. The generated markets are reasonably realistic and guaranteed to be arbitrage-free by construction. Nevertheless, the recursive mechanism used in the aforementioned paper produces significantly better results. See the extended version of this article for a more detailed description of the project.

1 Introduction

Options are financial securities of great importance to investors and risk managers in the financial industry and elsewhere. A company that produces biscuits, for example, may use options (or futures) on wheat to protect themselves financially against an increase in its price and therefore be able to sell their biscuits at a stable price [1].

In the financial industry, options are of particular interest for hedging strategies. A hedge is an investment which is intended to reduce the risk of adverse price movements in an asset. Normally, a hedge consists in taking an offsetting position in a related security [2]. As an example, an investor holding a stock can buy a put option to reduce their exposure to a fall in the price of the stock.

In a an ideal frictionless market like that of the Black-Scholes model [3], mathematical finance can provide for a perfect hedging strategy for a given portfolio of derivatives. However, hedging in real life is much more complicated and involves transaction costs, among other difficulties. The use of reinforcement learning to tackle this problem is an active field of research [4, 5]. However, deep machine learning models require huge amounts of data to avoid overfitting and the historical data might not be enough for many applications in finance. This motivates the creation of option market simulators, which can augment the existing data.

The goal of this work is to use Generative Adversarial Networks (GANs) to generate realistic synthetic option markets using a similar approach to that used in [6] but exploring a different generating mechanism.

2 State of the art

Time series of asset prices exhibit some common statistical features across a wide variety of markets and instruments. These features are known as *stylized facts* [7]. Generating financial time series consists in producing synthetic time series that properly imitate this behaviour or that of some instrument or market in particular.

The most basic way to generate financial time series is the use of stochastic processes. This approach consists in designing a stochastic process that can model the behaviour of such time series realistically and then fine-tuning its parameters on historical data. The time series are then generated as realisations of such stochastic processes using the calibrated parameters. Brownian motion [8] and geometric brownian motion [9] are the simplest stochastic processes used to model financial time series. Some more sophisticated attempts at this task come in the form of the Heston model [10] and the ARCH and GARCH models [11, 12]. The main advantage of generating financial time series through stochastic processes is the ease of implementation. The greatest shortcoming of these models is their lack of realism, since most of them fail to reproduce all *stylized facts* at once as pointed out in [7].

Generative modelling aims to solve the problem of generating data samples imitating those in a training set. Creating synthetic financial time series is a generative modelling problem, which motivates the application of sophisticated generative modelling techniques to the problem in hand. Generative Adversarial Networks (GANs) are an example of modern generative models. Some of the existing literature on the use of GANs to generate financial time series is explored here.

A GAN is a machine learning framework invented by Ian Goodfellow and his colleagues in 2014 [13]. In a GAN setup, two neural networks (a generator and a discriminator) play a game. The discriminator is a classifier whose goal is to distinguish fake data samples from real ones. The generator network's goal is to create fake data samples that are as realistic as possible so as to fool the discriminator into classifying them as real. Once the networks are trained, the generator can be used to produce synthetic data samples. GANs have been particularly succesful in generating synthetic images [14, 15, 16]. The GAN framework has also been applied to create synthetic time series [17, 18].

GANs have also been used succesfully in financial time series simulation in particular. In [19], a conditional GAN setting is used to generate financial time series to fine-tune and aggregate trading strategies. In [20], temporal convolutional networks (TCNs) are used for both the generator and discriminator to generate financial time series using GANs. Finally, in [21, 22] GANs are used to predict the stock market.

One of the most relevant studies to this project is [23]. In this paper, the author is able to generate synthetic returns series for some stocks belonging to the S&P500 index and for the Chicago Board of Options Exchange (CBOE) Volatility Index known as the VIX using GANs.

The literature on this subject also includes some articles regarding the simulation of option markets in particular. In [24] and [25], the authors model option markets using stochastic processes. The most relevant paper to this work is [6]. In this article, the authors create a set of generative models based on neural networks to simulate the dynamics of a surface of option prices. The generator is a deep neural network which maps noise and the current and past states of the market to the new state. The present project uses a similar methodology to that proposed in [6] based on the use of discrete local volatilities (DLVs) but explores a different generating mechanism.

3 Problem formulation

3.1 Option pricing

Options are financial securities which give their holder the right (but not the obligation) to buy or sell a certain asset on a certain time period at a predetermined price. The asset is called the underlying and the predetermined price is called the strike price [26]. A European option is that which can be exercised only at a specified expiry date and not before that. In this project we focus solely on European call options.

In 1973, Fisher Black and Myron Scholes published *"The Pricing of Options and Corporate Liabilities"* [27]. In this article, they derived a valuation formula for options from the idea that option prices should not allow for arbitrage opportunities. Under some assumptions, the no-arbitrage price $C(t,S)$ for a European call option at time t on an underlying asset with price S at t is given by:

$$C(t,S) = SN(d_1) - Ke^{-r(T-t)}N(d_2) \tag{1}$$

Where K is the strike price and T is the expiry date of the call option, r is the risk-free rate in the market, $N(\cdot)$ is the cumulative distribution function of a standard normal random variable and d_1 and d_2 are given by:

$$d_1 = \frac{\ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} \tag{2}$$

$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{3}$$

Where σ is the volatility assumed for the underlying asset price process between t and T.

3.2 Implied volatility

The Black-Scholes formula allows to calculate the price of a European option from the following five variables: the underlying asset current price S, the strike price K, the time remaining to the expiry date $T-t$, the risk-free rate in the market r and the volatility of the underlying asset price σ.

Note that the first 4 variables in the option price formula are market observables. The LIBOR index or the US national treasury bonds yield are typical values which are used as the theoretical concept of a riks-free rate. The only non-observable in the formula is the volatility of the underlying. This volatility is not observable firstly because it is the volatility of the underlying in the future (between time t today and the option expiry T). Each agent in the market will value an option using their best estimate of σ. Once a value for this volatility is chosen, the price of an option is fully determined.

In a typical order-driven matched bargain market, option prices are determined like those of any other financial security by an offer and demand mechanism using bid and ask prices. The quoted price of a security is the price of the last transaction [28]. In the same way as fixing a volatility for the underlying asset price σ determines an option price, fixing an option price also determines a volatility σ. The implied volatility of an option price is defined as the value of σ one must introduce in the Black-Scholes formula to obtain such option price. The implied volatility of an option price represents the market's expectation of the volatility of the underlying asset price up to the expiry date.

3.3 Project objective

On a given day and a given underlying asset, option prices and their respective implied volatilities for different combinations of strike and maturities define a surface on the strike-maturity plane.

In this project, the goal is to simulate the dynamics of the component of option prices that depends on investor's fear, which is given by implied volatility. Relative option prices (to the underlying price) are also a good measure of investors sentiment, as the short-term dynamics of relative option prices are primarily driven by changes in implied volatility. The dynamics of investor's fear can thus be represented by the time-evolution of relative option prices and implied volatility surfaces. In order to enable a fair comparison of such surfaces across time, these can be defined on a fixed grid of relative strike prices (to the current underlying price) and maturity horizons.

The no arbitrage principle in finance says there is no "free lunch" in the market. Informally, an arbitrage opportunity is the possibility to make a profit in a financial market without risk and without net investment of capital [29]. In the real world markets, arbitrage opportunities are small, rare and fleeting. It is therefore desirable that the generated option markets do not exhibit arbitrage opportunities. A surface of call prices needs to satisfy some conditions in order to be free of static arbitrage opportunities. The most simple of such conditions is that, for a fixed maturity, call prices must be decreasing in strike price. See the extended version of this article for a deeper explanation of such conditions [30].

The goal of this project is to generate synthetic multivariate time series that imitate the dynamics of relative call prices and their respective implied volatilities on a fixed grid of relative strikes and maturity horizons. The generated markets should be **realistic** and the generated surfaces should be **arbitrage-free**. The realism of the markets will be evaluated by comparing distributional and time series statistics on the historical and synthetic markets.

4 Methodology

The methodology is divided into three parts: processing of the options data, model training and evaluation of the synthetic markets.

4.1 Transformation of the data

The underlying index for the generated markets in this project is the S&P500. The generative models of this project are trained on processed data of closing pirces of European options on this index for the period between the 1st of January 2004 and the 30th of September 2019. The source of data is the Chicago Board of Options Exchange (CBOE) data shop [31].

The used grid on which the option markets are generated is defined as the set $K \times M$, where K is the set of relative strikes and M is the set of maturity horizons in business days, given by:

$$K = \{0.85,\ 0.9,\ 0.95,\ 1,\ 1.05,\ 1.1,\ 1.15\}$$
$$M = \{30,\ 50,\ 100,\ 150,\ 200\}$$

Training a generative model to produce option markets on such grid requires data of the historical evolution of option prices on such grid. However, financial markets do not offer all possible options on any combination of strikes and maturities, so there might not be observable data for option prices on the grid $K \times M$ for many days in the CBOE dataset. This requires that the prices on the grid $K \times M$ on each day are estimated by interpolation from the observable prices on such day.

In order to generate relative call price surfaces that are free of arbitrage one needs to impose such conditions on the generative model. The conditions on call prices are too complicated and those on implied volatilities are even more awkward [6]. Because there is no easy known way to impose such conditions on a GAN generator, it is convenient that the GAN generative model is trained to produce surfaces of discrete local volatilties (DLVs), as proposed in [6].

Discrete local volatilities (DLVs) provide a "codebook" for arbitrage-free call price surfaces because the set of such surfaces is in one-to-one correspondence with the set of positive DLVs surfaces. By transforming relative call price surfaces into DLVs and then into log-DLVs and training the GAN model on the log-DLVs historical process one does not need to worry about restraining the generator output, because any real-valued log-DLV surface will necessarily translate into a postive DLV surface and an arbitrage-free surface of relative call prices. The historical data of interpolated relative call prices on the grid $K \times M$ is translated into the form of log-DLVs. A correction procedure is applied before this transformation on those surfaces that exhibit arbitrage. The procedure is similar to that proposed in [32] and consists in finding the closest arbitrage-free surface of relative call prices to the input surface. The historical implied volatilities process is also computed, using the FVX index as risk-free rate.

Finally, because of the strong correlations across time between the different components of the log-DLV historical process, a dimensionality reduction procedure using Principal Component Analysis (PCA) is used. The log-DLV process is thus compressed from 35 dimensions to 10 principal components, while conserving 85.82% of the variance.

Figure 1 illustrates how the interpolated and corrected surfaces of relative call prices are processed:

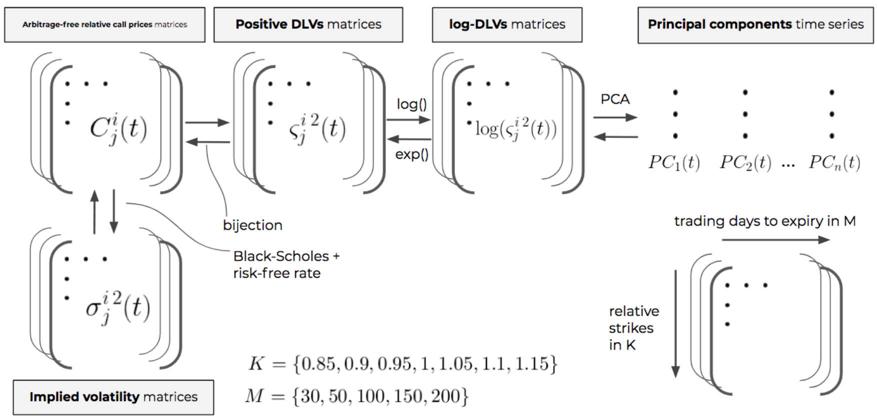

Figure 1: *Summary of the processing of the data from interpolated and corrected surfaces of relative call prices to the principal components of the log-DLV process.*

4.2 Models

The next stage of the project is to train a GAN to imitate the dynamics of each of the principal components historical series. A market can then be generated by sampling a time series from each of the 10 GANs independently and then carrying out the inverse processing of the data from principal components to relative call prices and implied volatilities accordingly. The principal components of the log-DLVs process are uncorrelated in the historical data by construction through the PCA algorithm, which is why this methodology is used. Generating a synthetic process of log-DLVs by applying the inverse PCA transformation on independently generated samples of the principal components should result in a correlation structure of the generated log-DLVs process close to that found in the historical process. When transforming relative call prices into implied volatilities for a generated market, the risk-free rate is assumed constant and takes a random value from the historical FVX series.

The particular GAN setting used in this project is the Wasserstein GAN with gradient penalty (WGAN-GP) [33]. In all models, the generator takes a random noise signal of 50 values as input, where the values are a sequence of independent and identically distributed standard normal random variables. The output of the generator is a sequence of 1000 values corresponding to a generated scenario for its corresponding principal component. The discriminator takes sequences of 1000 values for real and fake scenarios of its corresponding principal component and classifies them as real or fake.

In order to create a training set of 1000 values-long scenarios for a given principal component, a sliding window of length 1000 is rolled over the whole historical series. A picture of the series is taken every interval of 10 values, so each picture intersects with the previous one in 990 values.

Both the generator and discriminator are based on 1-dimensional convolutional layers, combined with batch normalization (in the generator only) and the use of Leaky ReLu as activation function. The full architecture of the discriminator and generator networks shall remain undisclosed due to confidentiality issues with the company at which this project is developed. Please contact ETS Asset Management Factory for further inquiries.

4.3 Evaluation of the generated markets

Evaluating the quality of the samples produced by a generative model is a complex task. In the particular case of synthetic image generation with GANs, one may think that fooling a person in distinguishing real and fake images should be the ultimate performance measure. However, this measure may favour models that concentrate on a limited section of the data (memorizing and mode dropping) and do not properly recover the whole data distribution. Quantitative measures, while being less subjective, may not correspond to how people perceive the images. These, along with other hardships make evaluating GAN performance notoriously difficult and there is no clear consensus as to which measure should be used [34]. Refer to [34] and [35] for further discussion of the evaluation of generative models.

In this article, the generated markets are evaluated by comparing a series of distributional and time series statistics on the real and synthetic data. This project is a variation of [6], which motivates the use of (some of) the same metrics as this paper to enable its use as a benchmark. The main difference between this project and [6] is that here time series are generated as one-dimensional images whereas in [6] a recursive generating mechanism is used. Comparing the scores obtained here against this benchmark may enable to draw some conclusions on the two methods.

Scores for one-dimensional time series:

Let $\boldsymbol{x}_h$ be the historical time series for a given variable. By taking sliding windows on this series as in the building of the training set one can build a set of one-dimensional images $\boldsymbol{x}^{(i)}$ of scenarios for such time series, denoted $\mathcal{D}_h$:

$$\mathcal{D}_h = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}\right\} \tag{4}$$

Let $\mathcal{D}_g$ be a set of synthetic scenarios for such time series:

$$\mathcal{D}_g = \left\{\widetilde{\boldsymbol{x}}^{(1)}, \widetilde{\boldsymbol{x}}^{(2)}, ..., \widetilde{\boldsymbol{x}}^{(M)}\right\} \tag{5}$$

In order to quantify the aforementioned measures of similarity, the following scores are defined:

- **Autocorrelation score**: This is simply the euclidean distance between the autocorrelation function (up to 32 lags) of the historical series and the average autocorrelation function in the synthetic samples:

$$\left\| ACF_{32}(\boldsymbol{x}_h) - \frac{1}{M}\sum_{i=1}^{M} ACF_{32}(\widetilde{\boldsymbol{x}}^{(i)}) \right\|_2 \tag{6}$$

- **Skewness score**: This is the euclidean norm of the difference between the skewness of the historical series and that of the whole generated set:

$$\left\| \text{skew}[\boldsymbol{x}_h] - \text{skew}[\left\{\widetilde{\boldsymbol{x}}^{(1)}, \widetilde{\boldsymbol{x}}^{(2)}, ..., \widetilde{\boldsymbol{x}}^{(M)}\right\}] \right\|_2 \tag{7}$$

- **Kurtosis score**: Likewise, the kurtosis score is the euclidean norm of the difference between the kurtosis of the historical series and that of the whole generated set:

$$\left\| \text{kurt}[\boldsymbol{x}_h] - \text{kurt}[\left\{\widetilde{\boldsymbol{x}}^{(1)}, \widetilde{\boldsymbol{x}}^{(2)}, ..., \widetilde{\boldsymbol{x}}^{(M)}\right\}] \right\|_2 \tag{8}$$

Cross-correlation scores for multi-dimensional time series:

In order to evaluate the degree of realism of multi-dimensional time series, it is not sufficient to evaluate the previous scores on each dimension individually. This would ignore the existing correlations between the different series. The correlation matrices of the generated and historical multi-dimensional processes are calculated and compared as a way of measuring how well the generator imitates the correlation structure of the real process.

Let $\boldsymbol{X}_h$ be the historical multi-dimensional time series and $\mathcal{D}_g = \left\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, ..., \boldsymbol{X}^{(M)}\right\}$ be a set of generated multi-dimensional time series. The correlation matrix of the historical process $\boldsymbol{X}_h$ is denoted Σ_h. The average correlation matrix of the generated samples in $\mathcal{D}_g$ is denoted Σ_g. The cross-correlation score is simply the euclidean distance between the two matrices:

$$\|\Sigma_h - \Sigma_g\|_2 \tag{9}$$

Evaluation procedure:

As shown in Figure 1, arbitrage-free option markets allow for several equivalent representations so the quality of the generated markets could be evaluated in any of these. Here, following what is done in [6], the markets are evaluated in implied volatilities.

Given a time series $\boldsymbol{x} = \{x_t\}_{t=1}^{T}$, the corresponding absolute returns series $\boldsymbol{r} = \{r_t\}_{t=2}^{T}$ is calculated as $r_t = x_t - x_{t-1}$. The individual metrics are calculated on each of the 35 series of the used grid $K \times M$ in implied volatilities and their absolute returns. These individual scores are then averaged to obtain a single score for the whole process. Additionally, the cross-correlation scores are calculated on log-DLVs as well as on implied volatilities, for comparison with the benchmark scores.

5 Results

The historical process of implied volatilities is shown in Figure 2:

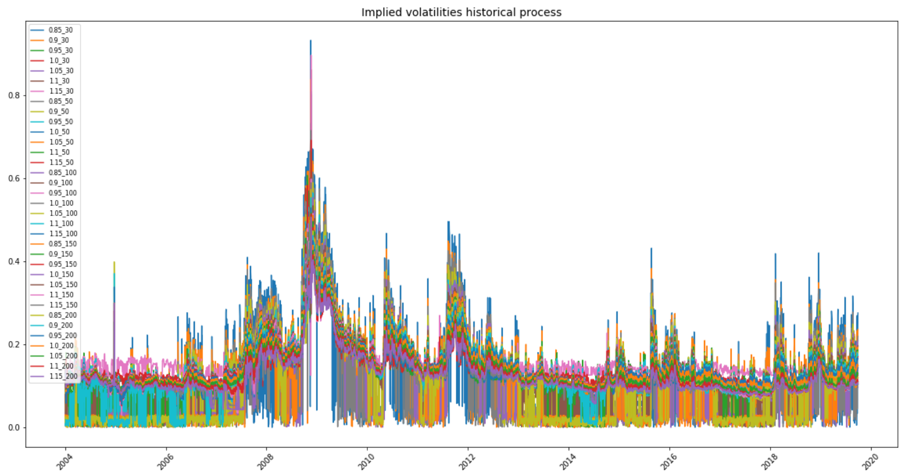

Figure 2: *Historical implied volatilities process.*

As a way of showing that the implied volatility markets have been generated correctly and look reasonable, two samples of the generated markets are presented:

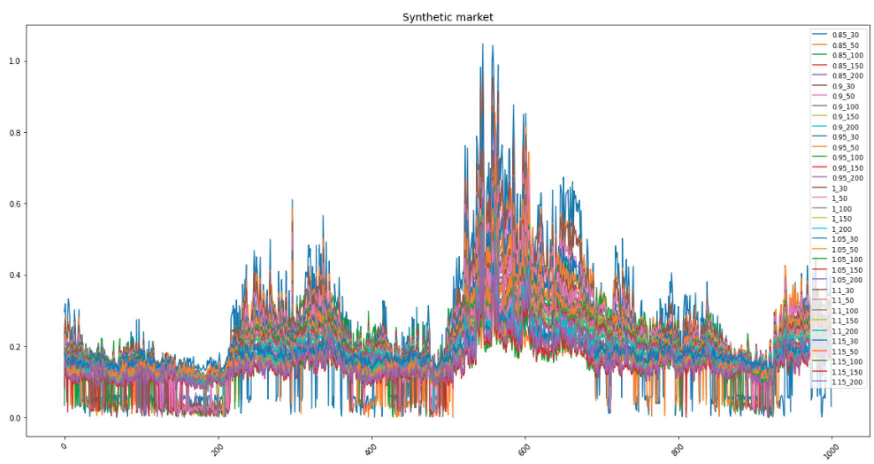

Figure 3: *Synthetic sample 1.*

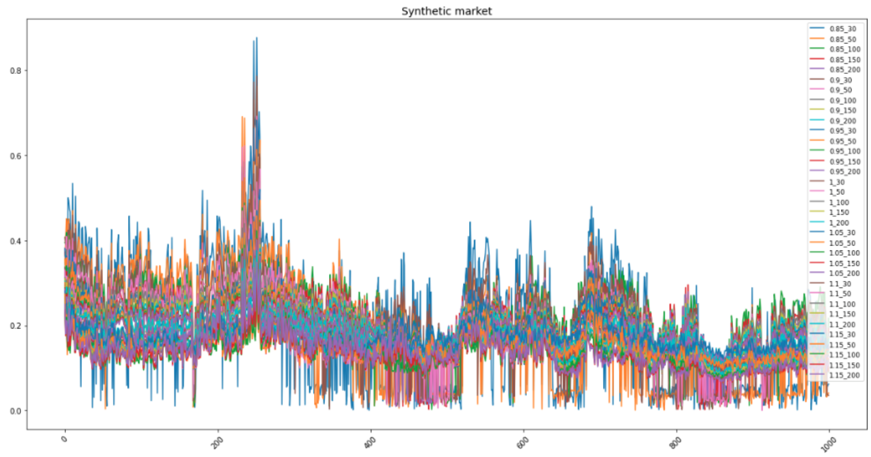

Figure 4: *Synthetic sample 2.*

At a first glance, the previous figures show how the generated samples correctly imitate the behaviour of the historical implied volatility process, with alternating periods of calmness and high volatility.

Table 1 shows the computed scores on the historical market and a sample of 200 synthetic markets. See appendix A for a series of plots comparing the real and synthetic distributions of markets.

	ACF	Skewness	Kurtosis	Cross-correlation
Values	1.435	1.022	4.719	4.850
Absolute Returns	0.289	1.024	61.821	5.756

Table 1: *Scores of the synthetic implied volatility markets.*

As explained, the scores for the best model in [6] are used as a benchmark. Table 2 shows a comparison between the results of this project and the benchmark on a few scores used in their work:

	Values ACF	Values skewness	Values kurtosis	log-DLVs cross-correlations	log-DLVs absolute returns cross-correlations
Benchmark	0.186	0.063	0.097	0.137	0.771
This project	1.435	1.022	4.719	6.811	15.518

Table 2: *Comparison with benchmark scores.*

The ACF, skewness and kurtosis scores in the previous table are calculated on implied volatilities whereas the cross-correlation scores are calculated on log-DLVs and their absolute returns.

6 Conclusion

The option markets generated in this project can be said to be decently realistic and are guaranteed to be arbitrage-free by construction. However, the scores obtained here are much worse than the benchmark scores for the best model in [6]. Although it is not rigorous to generalise across all network architectures, GAN settings and other variations of the methodology, generating option markets through the recursive mechanism of [6] seems to produce a distribution of synthetic option markets that approximates the real distribution more closely than the method used in this project.

A possible explanation is that the generator in this work aims to recover a distribution on a 1000-dimension space, whereas in [6] such distribution lies in a lower dimensional space (32, which is the number of points on the used grid, times the number of dimensions of the conditioning state). The task of learning a probability distribution is more difficult the higher the dimensionality of the underlying space.

Secondly, the sliding windows method used here to produce a training set of scenarios yields roughly 300 images per principal component from almost 15 years of option data. Using the methodology of [6] would result in a training set for the GAN of almost 4000 examples from the same 15 years of data.

Finally, in [6] the used models are trained on option prices on the EUROSTOXX 50 index between 2011 and 2019, which do not include the crisis of 2008 as in this project. This implies that the real data distribution of the present project could have a stranger shape, with a certain amount of mass at the anomalous events of 2008. This is likely a harder distribution to learn for a GAN generator, which may have deteriorated the results of this project with respect to not including the crisis of 2008 in the training data.

A Plots

The following figures show a series of plots comparing the empirical distributions found in the real and synthetic implied volatilities processes. The synthetic distributions are calculated on a sample of 200 synthetic markets. The plots are presented in a matrix form, where rows correspond to relative strikes and columns correspond to maturity horizons. All plots are calculated both on implied volatilities and their absolute returns. For each component of the processes, let $\boldsymbol{x}_h$, $\mathcal{D}_h$ and $\mathcal{D}_g$ be defined as in 4.3.

Figures 5 and 6 show the empirical distribution of values and absolute returns in the historical series against the distribution found in the generated sample:

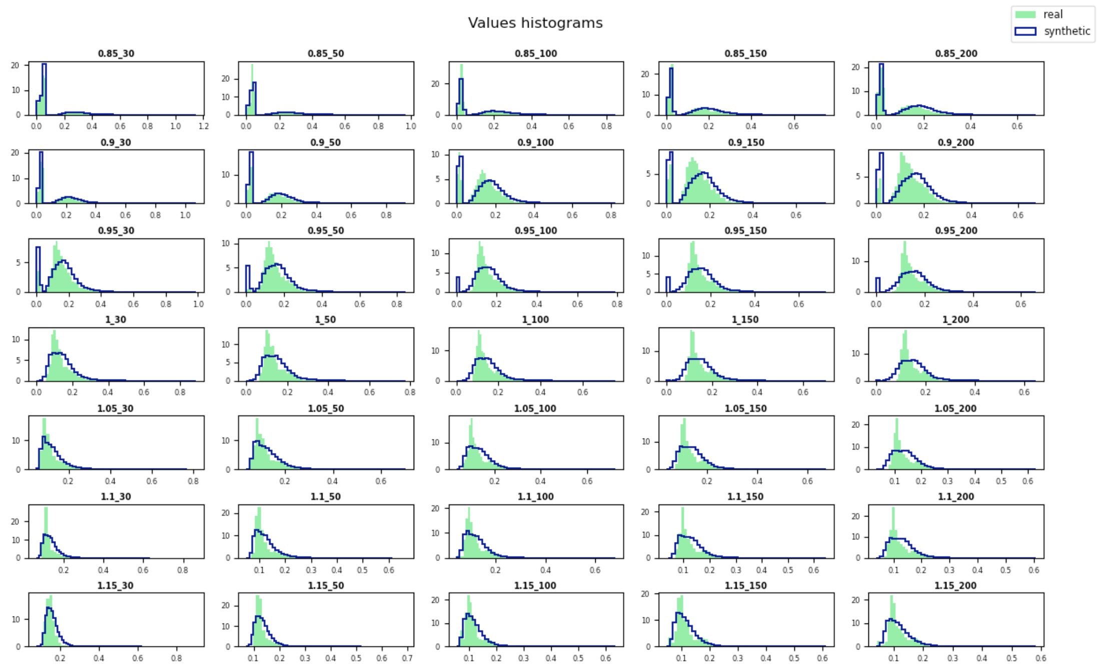

Figure 5: *Values histograms.*

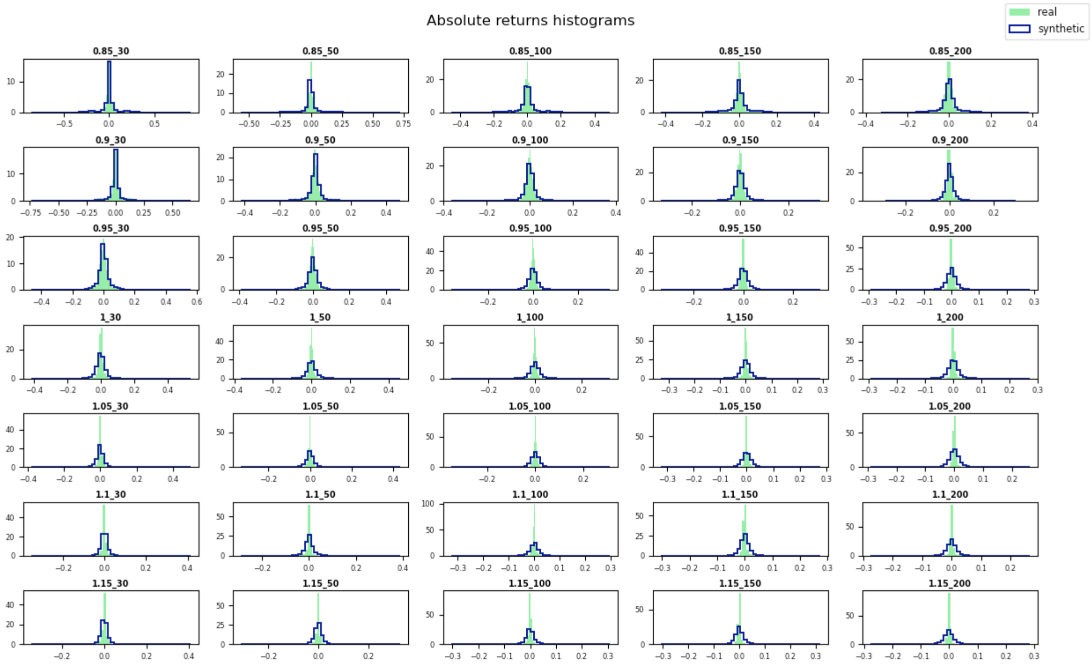

Figure 6: *Absolute returns histograms.*

Figures 7 and 8 show the autocorrelation functions of the historical series against the corresponding average autocorrelation function in the generated sample:

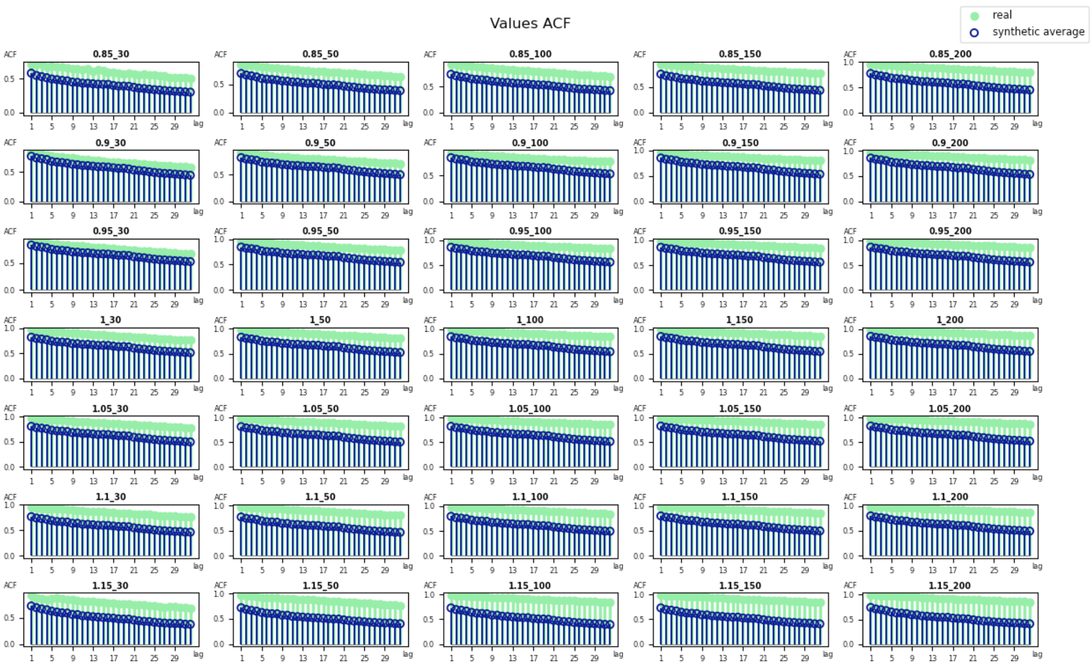

Figure 7: *Values ACFs.*

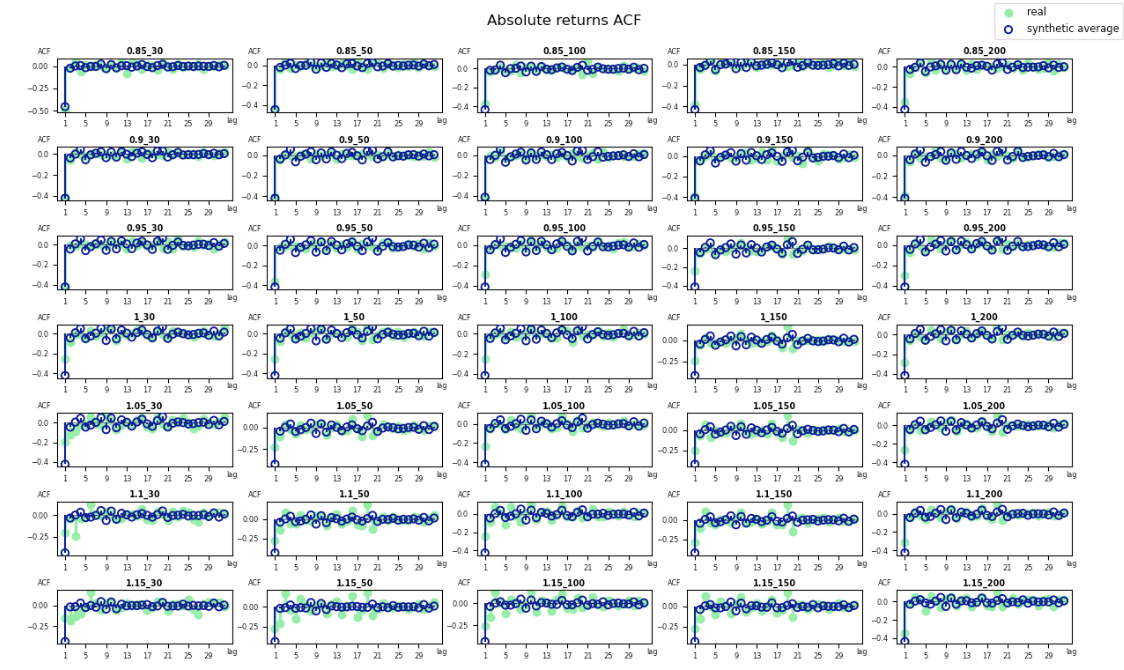

Figure 8: *Absolute returns ACFs.*

In order to compare the degree of asymmetry around the mean in the real and synthetic distributions, histograms of the skewness in the real and generated series are plotted. Figures 9 and 10 show, for each component, a histogram of skew$[\boldsymbol{x}^{(i)}]$ for $\boldsymbol{x}^{(i)}$ in $\mathcal{D}_h$ against a histogram of skew$[\widetilde{\boldsymbol{x}}^{(i)}]$ for $\widetilde{\boldsymbol{x}}^{(i)}$ in $\mathcal{D}_g$:

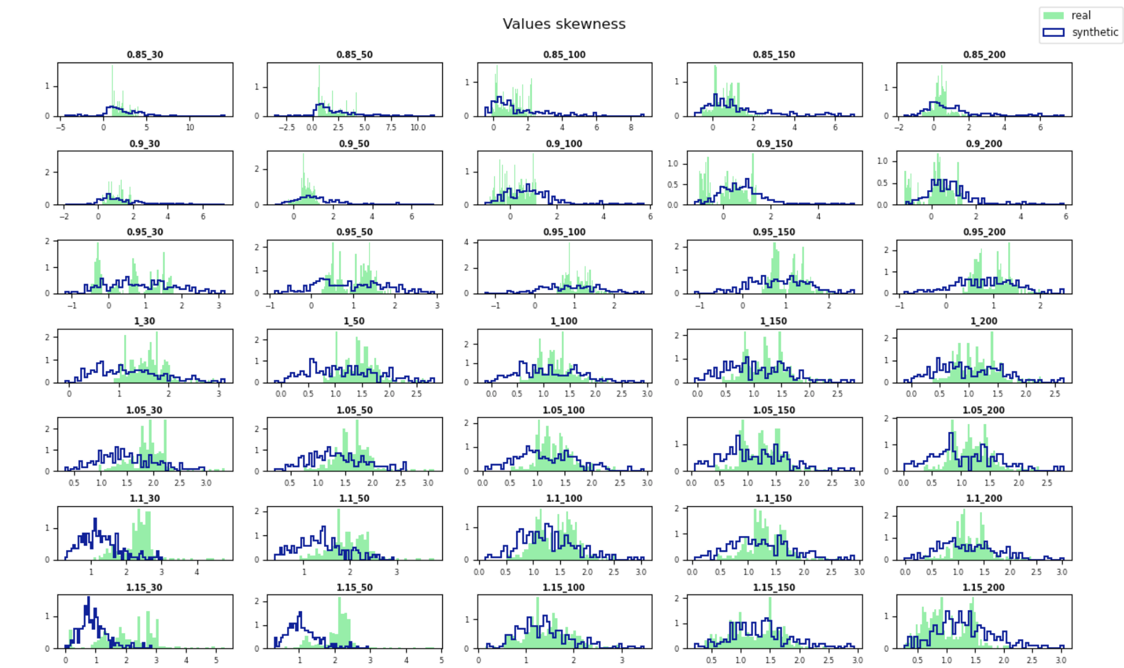

Figure 9: *Values skewness.*

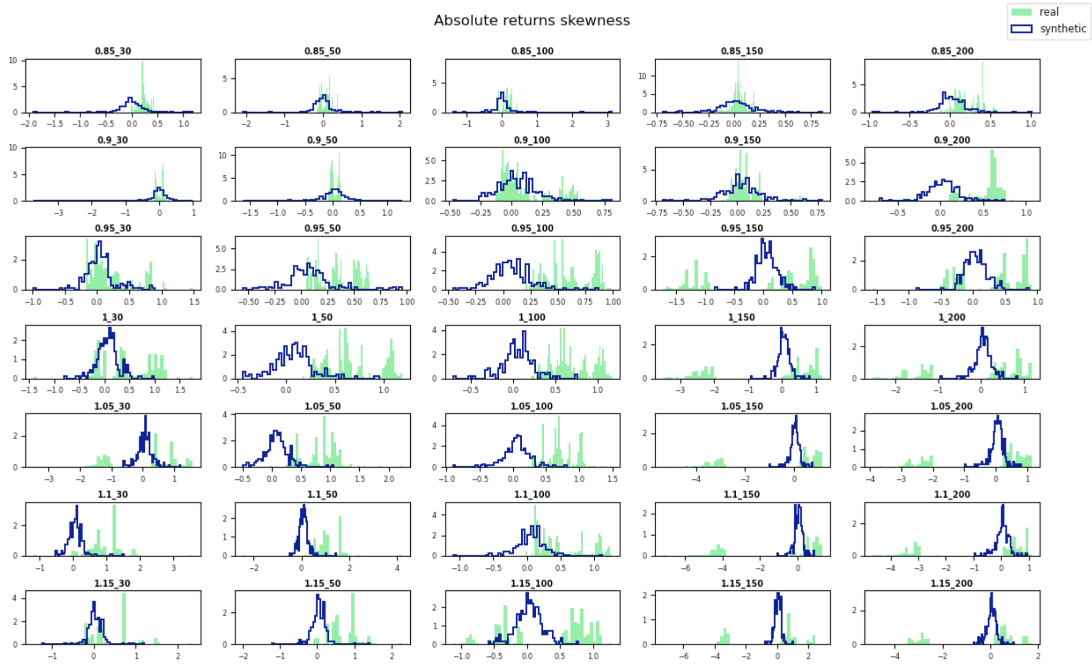

Figure 10: *Absolute returns skewness.*

Likewise, histograms of the kurtosis in the real and generated series are plotted as a way of comparing the probability of occurrence of extreme values. Figures 11 and 12 show, for each component, a histogram of $\text{kurt}[\boldsymbol{x}^{(i)}]$ for $\boldsymbol{x}^{(i)}$ in $\mathcal{D}_h$ against a histogram of $\text{kurt}[\widetilde{\boldsymbol{x}}^{(i)}]$ for $\widetilde{\boldsymbol{x}}^{(i)}$ in $\mathcal{D}_g$:

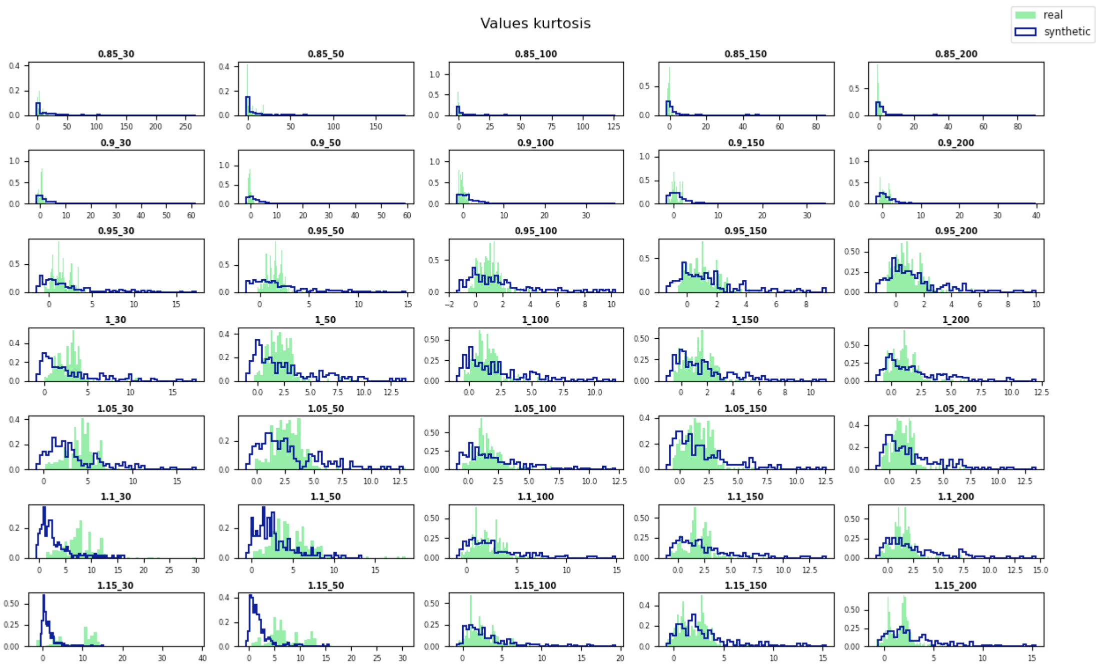

Figure 11: *Values kurtosis.*

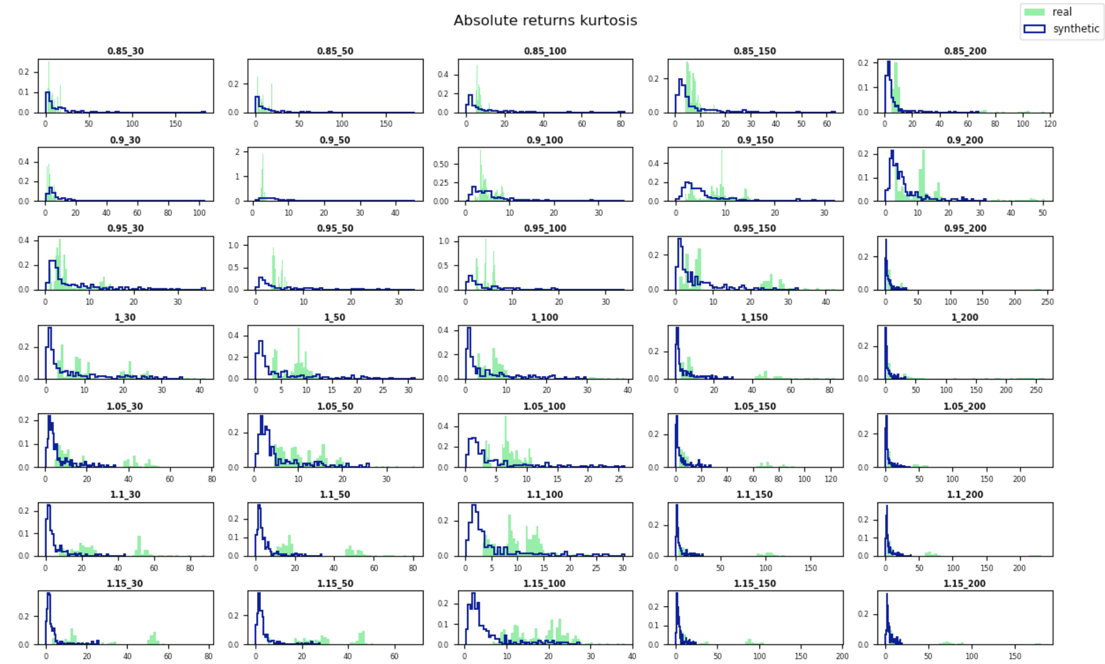

Figure 12: *Absolute returns kurtosis.*

Finally, Figures 13 and 14 show the correlation matrix of the historical market and the average correlation matrix on the synthetic markets:

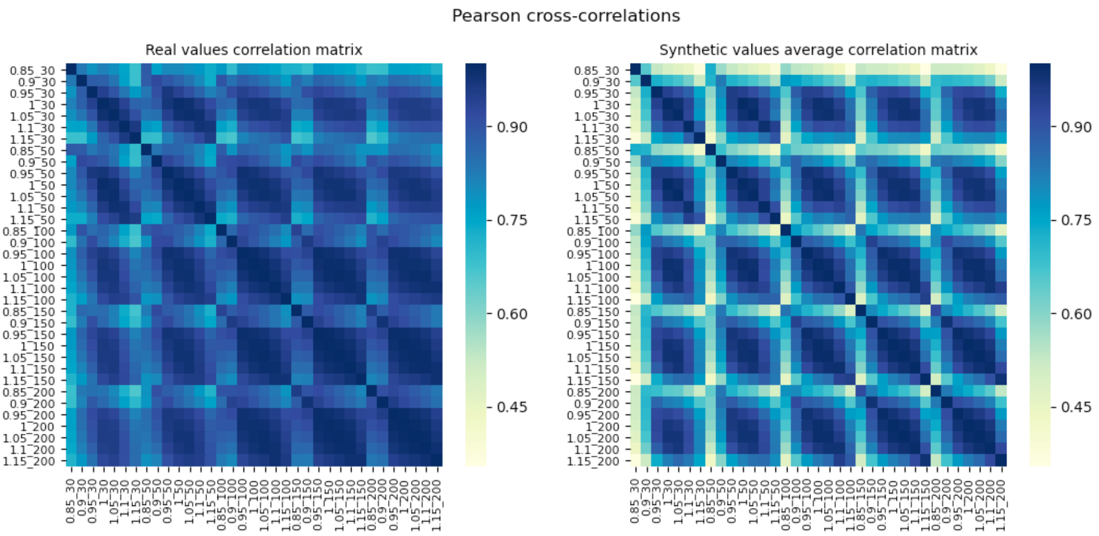

Figure 13: *Values cross-correlation matrices.*

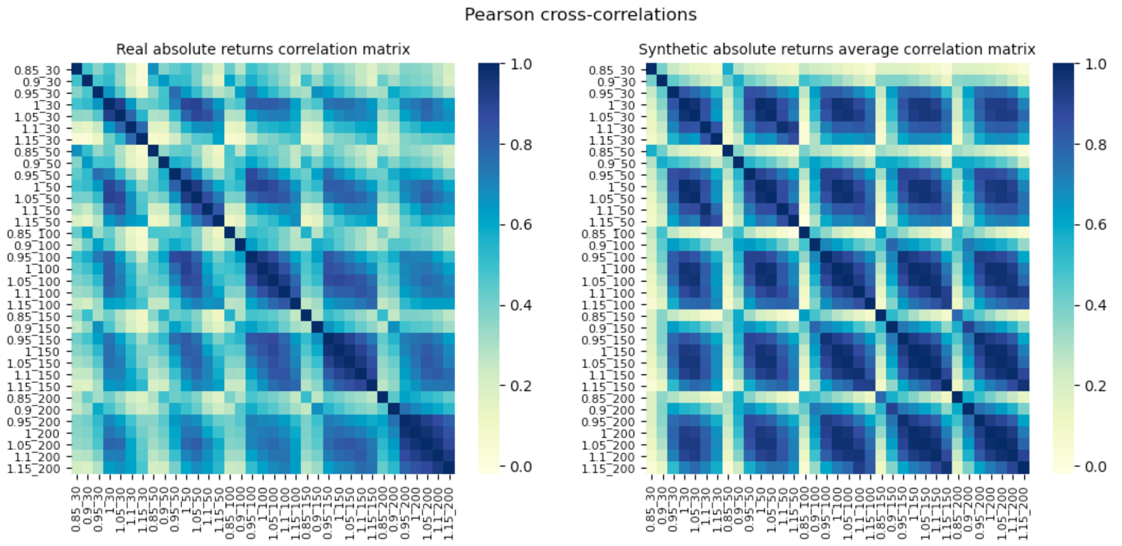

Figure 14: *Absolute returns cross-correlation matrices.*

References

[1] Alexandre Hubert. "Using Deep Learning for Better Option Pricing". In: (2020), p. 5.

[2] Nathan Reiff. *Hedge*. 2020. URL: https://www.investopedia.com/terms/h/hedge.asp.

[3] Wikipedia contributors. *Black–Scholes model — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-May-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?%09%09%09%20title=Black%E2%80%93Scholes_model&oldid=954778667.

[4] Petter N Kolm and Gordon Ritter. "Dynamic replication and hedging: A reinforcement learning approach". In: *The Journal of Financial Data Science* 1.1 (2019), pp. 159–171.

[5] Hans Buehler et al. "Deep hedging". In: *Quantitative Finance* 19.8 (2019), pp. 1271–1291.

[6] Magnus Wiese et al. "Deep hedging: learning to simulate equity option markets". In: *Available at SSRN 3470756* (2019).

[7] Rama Cont. "Empirical properties of asset returns: stylized facts and statistical issues". In: (2001).

[8] Wikipedia contributors. *Brownian motion — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-May-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?title=Brownian_motion&oldid=957390935.

[9] Wikipedia contributors. *Geometric Brownian motion — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-May-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?title=Geometric_Brownian_motion&oldid=947842861.

[10] Ricardo Crisóstomo. "An Analyisis of the Heston Stochastic Volatility Model: Implementation and Calibration Using Matlab". In: (2014).

[11] Wikipedia contributors. *Autoregressive conditional heteroskedasticity — Wikipedia, The Free Encyclopedia*. [Online; accessed 21-May-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?title=Autoregressive_conditional_heteroskedasticity&oldid=949619786.

[12] AK Panorska, S Mittnik, and ST Rachev. "Stable GARCH models for financial time series". In: *Applied Mathematics Letters* 8.5 (1995), pp. 33–37.

[13] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[14] Wikipedia contributors. *StyleGAN — Wikipedia, The Free Encyclopedia*. [Online; accessed 21-May-2020]. 2020. URL: https://en.wikipedia.org/w/index.php?title=StyleGAN&oldid=956027927.

[15] Kevin Schawinski et al. "Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit". In: *Monthly Notices of the Royal Astronomical Society: Letters* 467.1 (2017), pp. L110–L114.

[16] Xintao Wang et al. "Esrgan: Enhanced super-resolution generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. –.

[17] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. "Real-valued (medical) time series generation with recurrent conditional gans". In: *arXiv preprint arXiv:1706.02633* (2017).

[18] Luca Simonetto. "Generating spiking time series with Generative Adversarial Networks: an application on banking transactions". In: (2018).

[19] Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. "Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination". In: *arXiv preprint arXiv:1901.01751* (2019).

[20] Magnus Wiese et al. "Quant GANs: deep generation of financial time series". In: *Quantitative Finance* (2020), pp. 1–22.

[21] Kang Zhang et al. "Stock market prediction based on generative adversarial network". In: *Procedia computer science* 147 (2019), pp. 400–406.

[22] Xingyu Zhou et al. "Stock market prediction on high-frequency data using generative adversarial nets". In: *Mathematical Problems in Engineering* 2018 (2018).

[23] Fernando De Meer Pardo. "Enriching Financial Datasets with Generative Adversarial Networks". PhD thesis. Master's thesis, Delft University of Technology, the Netherlands, 2019.

[24] Sid Browne and Arthur Maghakian. *Method and system for simulating implied volatility surfaces for basket option pricing*. US Patent 7,440,916. Oct. 2008.

[25] Johannes Wissel. "Arbitrage-free market models for option prices". In: *Preprint, ETH Zurich* (2007).

[26] James Chen. *Options*. 2020. URL: `https://www.investopedia.com/terms/o/option.asp`.

[27] Fischer Black and Myron Scholes. "The pricing of options and corporate liabilities". In: *Journal of political economy* 81.3 (1973), pp. 637–654.

[28] Chris. W Rea. 2010. URL: `https://money.stackexchange.com/questions/1063/can-someone-explain-a-stocks-bid-vs-ask-price-relative-to-current-price`.

[29] Walter Schachermayer. "The notion of arbitrage and free lunch in mathematical finance". In: *Aspects of Mathematical Finance*. Springer, 2008, pp. 15–22.

[30] Ricardo Moreno Alonso. "Generation of synthetic option markets using generative adversarial networks (extended version)". In: (2020).

[31] *Cboe DataShop/Historical Data/End-of-Day Option Quotes Data*. URL: `https://datashop.cboe.com/option-quotes`.

[32] Hans Buehler and Evgeny Ryskin. "Discrete Local Volatility for Large Time Steps (Extended Version)". In: *Available at SSRN 2642630* (2017).

[33] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems*. 2017, pp. 5767–5777.

[34] Ali Borji. "Pros and cons of gan evaluation measures". In: *Computer Vision and Image Understanding* 179 (2019), pp. 41–65.

[35] Lucas Theis, Aäron van den Oord, and Matthias Bethge. “A note on the evaluation of generative models”. In: *arXiv preprint arXiv:1511.01844* (2015).