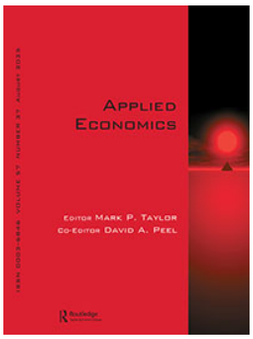

Routledge
Taylor & Francis Group

Applied Economics

ISSN: 0003-6846 (Print) 1466-4283 (Online) Journal homepage: www.tandfonline.com/journals/raec20

GANs and synthetic financial data: calculating VaR*

David E. Allen, Leonard Mushunje & Shelton Peiris

To cite this article: David E. Allen, Leonard Mushunje & Shelton Peiris (2025) GANs and synthetic financial data: calculating VaR*, Applied Economics, 57:37, 5680-5695, DOI: 10.1080/00036846.2024.2365456

To link to this article: https://doi.org/10.1080/00036846.2024.2365456

Published online: 18 Jun 2024.

APPLIED ECONOMICS
2025, VOL. 57, NO. 37, 5680–5695
https://doi.org/10.1080/00036846.2024.2365456

Routledge
Taylor & Francis Group

GANs and synthetic financial data: calculating VaR*

David E. Allen[a,b,c*], Leonard Mushunje[d] and Shelton Peiris[a]

[a]School of Mathematics and Statistics, The University of Sydney, Sydney, NSW, Australia; [b]School of Business and Law, Edith Cowan University, WA, Australia; [c]Department of Finance, Asia University, Wufeng, Taiwan; [d]Department of Statistics, Commercial Bank of Zimbabwe and Columbia University, New York, USA

ABSTRACT

Generative Adversarial Neural nets (GANs) are a new branch of machine learning techniques. A GAN learns to generate new data from the training data set. We examine the characteristics of the fake financial data using GANs trained on samples of daily S&P 500 and FTSE 100 index values. GANs feature two competing neural networks in a game theoretic context. The Generator net generates pseudo data that is presented to the discriminator net which then attempts to distinguish between the real and the fake data. This facilitates unsupervised learning on the dataset. The generative network generates data sets, while the discriminative network evaluates them. Equilibrium is reached when the generator can fool the discriminator half the time. Potential convergence difficulties led to the development of Wasserstein GANs, which we use in the analysis. We examine the characteristics of the generated fake S&P500 and FTSE 100 data sets. A key issue is how closely does the fake series mimic the real series? We explore this using a variety of metrics including regression analysis, applications of moments and characteristic functions, plus Random Forest analysis. We provide a practical application using the fake data to calculate Value at Risk (VaR).

KEYWORDS

Generative Adversarial Neural nets (GANs); synthetic data; VaR; moments; random forest

JEL CLASSIFICATION

G12; G14; G17; C70; C80

I. Introduction

Generative Adversarial Neural Networks (GANs), as developed by Goodfellow et al. (2014) can be viewed as a modified version of the Turing test, first proposed by Alan Turing, the father of modern computing, in 1950. Turing suggested a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human. Turing proposed that a human evaluator would judge natural language conversations between a human and a machine designed to generate human-like responses, via a keyboard, from a machine in the next room. If the participant could not distinguish human responses from those of a machine, the machine would have passed the test.

GANs consist of two different neural networks, a generator G and a discriminator D. The human in one of the rooms in the Turing example is replaced by another neural network. The generator G is responsible for the generation of data, and the discriminator D functions to judge the quality of the generated data and provide feedback to the generator G. These neural networks are optimized under game-theoretic conditions: the generator G is optimized to generate data that deceive the discriminator D and the discriminator D is optimized to distinguish the source of the input, namely the generator G or realistic dataset.

In the next section, we provide brief descriptions of some of the main forms of financial time series GANS models. Unlike other time series models, GANs are maximum likelihood free. We do not necessarily implement such techniques when training GANs. GANs are examples of generative models, where the term can be used to refer to any model that takes a training set, consisting of samples drawn from a distribution *pdata*, and learns to represent an estimate of that distribution somehow. The result is a probability distribution pmodel. This might be estimated explicitly, or samples drawn from its distribution might be generated. This might seem redundant, but it can inform and improve our ability to represent and manipulate high-dimensional probability distributions.

CONTACT David E. Allen profallen2007@gmail.com School of Mathematics and Statistics, The University of Sydney, Carslaw Building, Eastern Avenue, Camperdown, Sydney, NSW 2050, Australia

*The analysis in the paper was undertaken with Python, R, and GRETL. We are grateful to the reviewers for their constructive comments.

Generative models of time series data can be used to simulate possible future scenarios. GANs can be used to improve image resolution, create art and be used for image translation.

In the context of this paper, we use a GAN to generate fake financial data. This could be useful in circumstances where adequate financial data is lacking, or in which circumstances are rapidly changing. We use a variety of methods including regression analysis, the development of higher moments and cumulants, plus Random Forest machine learning techniques to evaluate the difference between the real and synthetic financial series. In the example that follows, we use GAN-derived financial data to estimate Value-at-Risk (VaR) for one of the major stock market indices, namely the S&P 500 index, having used GANs to generate synthetic data for both the S&P500 Index and the FTSE 100 index.

The paper is organized into four sections, subsequent to this introduction, the sample and method are discussed in section II, the results in section III and the paper concludes in section IV.

II. Sample and method

GAN architecture

Figure 1 shows all the states and output including the connections between all the networks that make up a full GAN network. Firstly, the generator network takes in the random input, before it is passed on to the discriminator network. On the other hand, the real input is passed directly to the discriminator function. The discriminator then classifies the output as either real or fake. This comes with loss functions for each classification, as we shall further discuss.

GANs have the advantage of using latent code, no Markov chains are required, and they are often regarded as producing the best samples, Goodfellow (2016).

$$Generator\ Network\ x = G(z; \theta^{(G)}). \tag{1}$$

Where the function must be differentiable, have no invertibility requirement, be trainable for any size of z, x can be made conditionally Gaussian, given z, but there is no requirement to do this.

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} logD(x) - \frac{1}{2}\mathbb{E}log(1 - D)G(z))). \tag{2}$$

$$J^{(G)} = -J^{(D)}.$$

Goodfellow (2016) points out that the equilibrium is a saddle point of the discriminator loss, the process resembles the Jensen-Shannon divergence, and the generator minimizes the log probability of the discriminator being correct. The discriminator's position is displayed in Figures 2 and 3.

What is the solution to $D(x)$ in terms of *pdata* and *pgenerator*? Assume that both densities are non-zero everywhere and solve for where the functional derivatives are zero.

$$\frac{\delta}{\delta(D)(x)} = 0.$$

Goodfellow (2016) explains further that in a non-saturating game:

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim P_{data}} logD(x) - \frac{1}{2}\mathbb{E}_{Z} log(1 - D(G(z))) \tag{3}$$

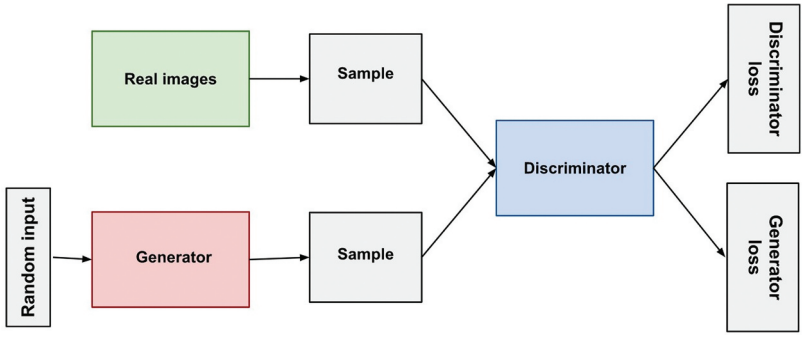

Figure 1. GAN Architecture. Source: Goodfellow (2016).

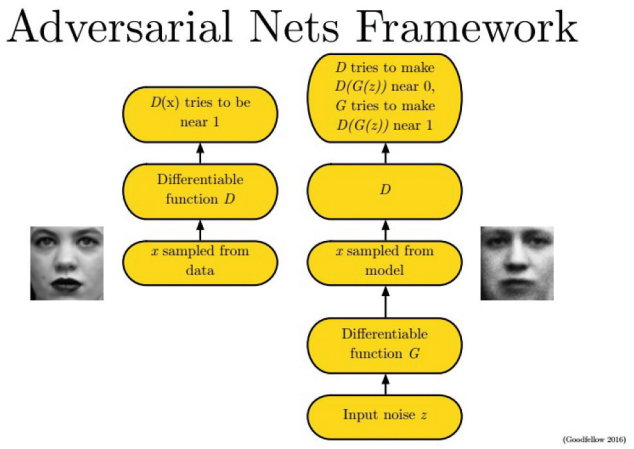

Figure 2. Discriminator Strategy. Source: Goodfellow (2016).

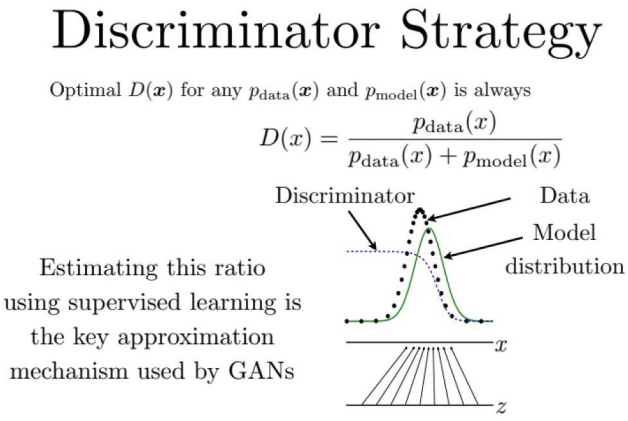

Figure 3. Discriminator Strategy. Source: Goodfellow (2016).

$$J^{(G)} = \frac{-1}{2}\mathbb{E}_z log D(G(z)).$$

In the above case, equilibrium is no longer describable with a single loss, and the generator maximizes the log probability of the discriminator being mistaken. However, in a practical sense, the generator can still learn even when faced with situations in which the discriminator rejects all the samples presented to it by the generator.

An essential core idea of a GAN is the use of 'indirect' training through the discriminator, another neural network that can tell how 'realistic' the input seems, which itself is also being updated dynamically. A key aspect of this is that the generator is not trained to minimize the distance to a specific image, in the case of image recognition, but rather on ways to fool the discriminator. This enables the model to learn in an unsupervised manner.

In a more mathematically precise manner, Wiese et al. (2020) have suggested the following mathematical illustration of the core mechanics of GAN architecture.

In more formal terms, let $N_Z, N_X \in \mathbb{N}$ and let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Furthermore, assume that X and Z are $\mathbb{R}^{N_X}$ and $\mathbb{R}^{N_Z}$ valued random variables, respectively. The distribution of a random variable X will be denoted by $\mathbb{P}_X$.

For the purposes of GANs $(\mathbb{R}^{N_Z}; \mathcal{B}(\mathbb{R}^{N_Z})$ and $(\mathbb{R}^{N_X}; \mathcal{B}(\mathbb{R}^{N_X})$ are called the latent and the data measure space, respectively. The random variable Z represents the noise prior and X the targeted (or data) random variable. The goal of GANs is to train a network $g : \mathbb{R}^{N_Z} \times \Theta^{(g)} \to \mathbb{R}^{N_X}$ such that the induced random variable $g\theta(Z)g\theta \circ Z$ for some parameter $\theta \in \Theta^{(g)}$ and the targeted random variable X have the same distribution, i.e. $g\theta(Z) \overset{d}{=} X$. Goodfellow et al. (2014) suggested the Generalized Adversarial Neural Net (GAN) modelling framework for NNs and introduced the generator and the discriminator nets as follows:

The generator net can be defined as $g : \mathbb{R}^{N_Z} \times \Theta^{(g)} \to \mathbb{R}^{N_X}$be a network with parameter space $\Theta^{(g)}$. A random variable $\tilde{X}$, as defined by:

$$\tilde{X} : \Omega \times \Theta^{(g)} \to \mathbb{R}^{N_X}$$

$$(\omega\theta \mapsto g_\theta(Z(\omega)),$$

is called the generated random variable. Furthermore, the network g is called a generator and $\tilde{X}_\theta$ the generated random variable with parameter θ.

The discriminator net can be defined as follows: $\tilde{d} : \mathbb{R}^{N_X} \times \Theta^{(d)} \to \mathbb{R}$ be a network with parameters $\eta \in \Theta^{(d)}$ and $\sigma : \mathbb{R} \to [0,1] : x \mapsto \frac{1}{1+e^{-x}}$ be the sigmoid function. A function $d : \mathbb{R}^{N_X} \times \Theta^{(d)} \to [0,1]$ which is defined by $d : (x, \eta \mapsto \sigma \circ \tilde{d}_\eta(x))$ is called a discriminator.

The discriminator and generator are applied to a sample that comprises a collection of $\{Y_i\}\, i \overset{M}{=} 1$ of M independent copies of some random variable Y is called an M size sample of Y. The notation $\{y_i\}\, i \overset{M}{=} 1$ refers to a realization $\{Y_i(\omega)\}\, i \overset{M}{=} 1$ for some $\omega \in \Omega$.

The adversarial GAN framework involves the discriminator and generator net competing

against each other in a game-theoretic zero sum game. The generator tries to create samples $\{\tilde{x}_\theta, i\}\, i \overset{M}{=} 1$ such that the discriminator cannot distinguish whether the samples were drawn from the target or the generated distribution.

Therefore, the discriminator $\tilde{d}_\eta : \mathbb{R}^{N_X} \to [0,1]$ behaves as the detective net or classifier and assigns a probability $x \in \mathbb{R}^{N_X}$ a probability that each sample drawing is a realization of the target distribution.

This means that the optimization of GANs can be considered as involving two steps. In the first, the discriminator's parameters $\eta \in \Theta^{(d)}$ are chosen to maximize the function $\mathcal{L}(\theta\cdot), \theta \in \Theta^{(g)}$, given by:

$$\mathcal{L}(\theta,\eta) := \mathbb{E}[log(d_n(X))] + \mathbb{E}[log(1 - d_n(g_\theta(Z)))]$$

$$= E[log(d_n(X))] + E[log(1 - d_\eta(\tilde{X}_\theta))].$$

In this manner, the discriminator net learns to distinguish between real and generated data.

In the second, the generator's parameters $\theta \in \Theta^{(g)}$ are trained to minimize the probability that the generator's samples are distinguished as not being from the data sample but from a generated mimic.

Thus, the GAN objective becomes a min-max game.

$$\overset{min}{\theta \in \Theta^{(g)}}\ \overset{max}{\eta \in \Theta^{(d)}}\ \mathcal{L}(\theta,\eta).$$

The last decade has seen an explosion in the variety of methods used both in implementations and applications of GANs. There is now a virtual 'zoo' of different types of GANs. These include Vanilla GANs, conditional Gan (CGAN)s, deep convolutional GAN (DCGAN)s, CycleGANs, Generative Adversarial Text to Image Synthesis, Style GAN, Super Resolution GAN (SRGAN), to name but a few.

Further complications are involved in the choice of the discriminator function. The point of the discriminator is mainly to act as a critic to provide feedback for the generator network about 'how far it is from perfection', where 'far' is defined as Jensen – Shannon divergence. However, there are a number of different criteria that can be used to measure distance. There are many possible divergences to choose from, such as the f-divergence family, which would give the f-GAN, see Nowozin et al. (2016). The Wasserstein GAN is obtained by using the Wasserstein metric, which satisfies a 'dual representation theorem' that renders it highly efficient to compute, see Arjovsky, Chintala, and Bottou (2017). We apply this method in the current paper.

Prior work applying GANs in finance include Zhou et al. (2018), who integrated the adversarial learning framework to stock price prediction. Koshiyama et al. (2019), employed GANs to develop a financial trading strategy, whilst Fiore et al. (2019) applied GANs to credit card fraud detection. Takahashia et al. (2019) explored the modelling of financial time series using GANs. Eckerli and Osterrieder (2021) provide a survey of GAN applications in finance. Cont et al., (2022) have developed a model employing GANs to explore tail risk, whilst Wiese et al., (2020) have constructed a 'Quant Gan' approach which focuses on the development of stochastic volatility neural network processes that model known facts about financial time series.

Wiese et al (2020) also provide a significant contribution in their rigorous mathematical definition of Temporal Convolution Neural Nets (TCNs), for the first time in the literature, and demonstrate their application to the generation of financial return data. The key ingredient of TCNs is dilated causal convolutions. These causal convolutions are convolutions, where the output only depends on past sequence elements. They construct a neural net architecture that is consistent with stochastic volatility models, and describe the generator function of their Quant GANs: as the stochastic volatility neural network (SVNN).

Cont et al. (2022) focus on using GANs to model tail risk and exploit the joint elicitability property of Value-at-Risk (VaR) and Expected Shortfall (ES). Their approach is capable of learning to simulate price scenarios that preserve tail risk features for benchmark trading strategies, including consistent statistics such as VaR and ES. They examine various trading strategies and explicitly focus on modelling tail risk.

Thus, their discriminator function $\bar{D}$ takes strategy *PnL* distributions as inputs, and outputs two values for each of the K strategies, aiming to provide the correct $(VaR_\alpha; ES_\alpha)$.

$$\bar{D}^* \in arg\min_{\bar{D}} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{p\sim\mathbb{P}_\tau}\left[s_\alpha \left(\begin{array}{c} \overbrace{\bar{D}(\underbrace{\Pi^K \neq P_\tau}_{strategy\ \ pnl\ \ distribution}; p)}^{VaR\ and\ ES\ prediction\ from\ \bar{D}} \end{array} \right) \right].$$

Moments and characteristic functions

Cornish and Fisher (1938) set out the properties of moments, characteristic functions and cumulants in a celebrated paper. Mendel (1991) summarized a number of their uses and novel applications in the signal processing literature. Cumulants have only recently been adopted in the finance literature. For example, Martin (2013) incorporated them into the consumption-based asset-pricing model by suggesting that information about the higher moments – equivalently, cumulants, of consumption growth is encoded in the cumulant-generating function, and recently Kyle and Kyle and Todorov (2023) have included them in a more general, risk premium, asset pricing framework.

Cornish and Fisher (1938) suggested that the distribution of a variable quantity x can be specified by means of a frequency distribution f, which is frequently termed the probability integral, which specifies the total frequency in the population for which the variable is less than an assigned value x. If the distribution f is continuous and differentiable, then, $\delta f/\delta x$ represents the frequency density in element of the range x, or the ordinate of the frequency curve at this point.

Cornish and Fisher (1938) proceed to define the characteristic function in the following manner by defining a function of the real variable t in the form:

$$M(t) = \int_{-\infty}^{\infty} e^{itx} df, \tag{4}$$

which is called the characteristic function of the distribution. If, in the neighbourhood of $t = 0$, M can be expanded in a series of powers of t, the series will be:

$$\sum_{r=0}^{\infty} \frac{(it)^r}{r!} \int_{-\infty}^{\infty} x^r df,$$

or,

$$\sum_{r=0}^{\infty} \frac{(it)^r}{r!} \mu_r^{'}$$

Where $\mu_r^{'}$ is the rth moment of the distribution of x about the origin. The characteristic function may be described as the moment generating function. $\mu_r^{'}$ is the average value of x^r. If $\mu_1^{'}$ is the mean, the factor e^{itx} may be resolved into the product,

$$e^{1it,\mu^{'}}.e^{it}(x-\mu_1^{'}) \tag{5}$$

of which the first factor is a constant, whilst the second is the mean of the distribution, and thus this expression can be expanded to generate the moments about the mean. The relationship between moments about zero and moments about the mean can be captured in the following expression:

$$1 + \mu_2 \frac{t^2}{2!} + \mu_3 \frac{t^3}{3!} + \mu_4 \frac{t^4}{4!} + \ldots.$$

$$= e^{-\mu_1^{'}t}(1 + \mu_1^{'}t + \mu_2^{'}\frac{t^2}{2!} + \mu_3^{'}\frac{t^3}{3!} + \mu_4^{'}\frac{t^4}{4!} + \ldots.),$$

producing a series of relations:

$$\mu_2 = \mu_2^{'} - \mu_1^{'2}$$

$$\mu_3 = \mu_3^{'} - 3\mu_3^{'}\mu_1^{'} + 2\mu_1^{'3}$$

$$\mu_4 = \mu_4^{'} - 4\mu_3^{'}\mu_1^{'} + 6\mu_2^{'}\mu_1^{'2} - 3\mu_1^{'4}$$

In this manner, the moments about the mean can be obtained from those about any other origin. Cornish and Fisher (1938) also mention that Laplace (1774) introduced a function called the cumulative function which is the logarithm of the

characteristic function. If x is distributed in a distribution that is specified by the frequency element and y is independently distributed in a distribution specified by the element, the frequency of the simultaneous occurrence of any pair of values x and y will be, and the characteristic function of the sum, will be:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{it(x+y)}\,df_1\,df_2 \tag{6}$$

This is the product of the characteristic function of x and y separately.

In the analysis that follows, we adopt a number of tests to assess the similarity of the real and fake returns series generated by the application of GANs. A simple weak-form market efficiency test, see Fama (1965), involves regressing the return in a time series on one lag of itself. Summers (1986) exposed the fact that this type of test has really low power.

For this reason, we also plot periodograms of the real and fake returns series, plus construct auxiliary series featuring the first four moments and cumulants, which are obtained when we pair an observation of a series with a lag of itself over a particular interval. These series are included in the regression analysis to reveal further characteristics of the distributions of the real and fake series.

Random forest analysis

To further explore these relationships, a random forest analysis was undertaken using the R packages RandomForest, Liaw and Wiener (2002), and random Forest Explainer, Paluszynska et al. (2020).

Random forests are an ensemble learning method for use in classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. Ho (1995) was the first developer of Random forests, and he demonstrated that the building of trees in randomly selected subspaces of the feature space can improve accuracy. Breiman (2001) further developed the method and constructed the first Fortran program with Cutler for implementing the method, and the approach was later included in the R package RandomForest, used in our analysis. Breiman also constructed the concept of 'bagging' : bootstrap aggregation. This method of random sampling with replacement improves the stability and accuracy of machine learning algorithms used in statistical classification and regression. The process involves taking a training set with responses bagging (B times) repeatedly involves selecting a random sample with replacement of the training set and fitting trees to these samples: For $b = 1, \ldots, B$: Sample, with replacement, n training examples from X, Y; call these X_b, Y_b. Train a classification or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x'). \tag{7}$$

This procedure improves performance because it decreases the variance of the model without increasing bias.

The current paper is closest in conception and application to Takahashia et al. (2019), but there are fundamental differences. They explain that there are difficulties in modelling financial time series with stochastic processes such as the ARCH and the GARCH models. These describe a financial time-series as a series of random variables with temporally dependent parameters. However, it is difficult to recover all the major stylized facts with such explicit mathematical formulations. In effect, typical time-series models focus on particular aspects of a time series. They explore whether the application of GANs can capture a broad range of the typical characteristics of a financial time series in one-pass. However, Takahashia et al. (2019) apply GANs to capture stock price return series, not to stock prices in levels.

In this paper, we take a further step back. We apply GANs to capture the behaviour of two indices; namely the S&P500 Index and the FTSE100 Index in levels, using daily observations. Next, we convert both the real and the GAN derived fake index series into time series of returns and analyze the characteristics of the real and fake return series using a variety of metrics. Our results suggest that GANs are remarkably successful in

capturing a broad range of the characteristics of financial time-series data sets. However, there are flaws apparent in the lagged values of the return series when we convert the GAN levels series into synthetic return series.

III. Results

Data characteristics

We downloaded daily data for the S&P 500 index for a period from 3 January 2012 to 22 December 2022 from Yahoo Finance. This gave a total of 2063 daily observations. We did the same for the FTSE 100 Index taking a sample of daily prices beginning 18 January 2010 and ending 10 March 2023. The data was obtained using the R library package quantmod, Ryan and Ulrich (2023). This sample was then subdivided into an estimation period running from 18 January 2010 until 29 March 2019, with a total of 2325 observations, followed by a hold out predictive sample period running from 1 April 2019 up to 10 March 2023, which contained 995 observations.

The Python Pytorch library was used to fit the GANs analysis. The process of fitting the model generated a fake 500 index data set. Wasser GANs are employed in this paper to generate the series, see Arjovsky, et al. (2017). The purpose of the current paper is to explore how closely the fake S&P 500 and FTSE100 index series mimic the real series.

Table 1 presents descriptive statistics for both the fake and real series for both the S&P500 and the FTSE100 indices. On all metrics, the real and the fake series at both levels and log differences are remarkably similar. The mean and median of the two levels series in both cases are very similar. Their standard deviations, skewness and excess kurtosis are also similar. KPSS unit root tests with trend on both series reject the null of non-stationarity. Their Hurst exponents are almost the same at 1.02 and 1.01. This value suggests that both series have long-term positive autocorrelation or long memory.

We also took the logarithmic first differences of the two series to produce a real and a fake S&P500 and FTSE100 return series. Descriptive characteristics for the two series are shown in the bottom half of Table 1.

Once again in terms of their means, medians, minimums, maximums, standard deviation and excess kurtosis, they are very similar. There is less negative skewness on the fake series. KPSS unit root tests with a trend suggest that both the real and fake return series are stationary. Graphs of both the index levels and returns series are shown in Figure 4.

Time series analysis

Table 1 reports the Hurst exponents for the real and fake return series and the Hurst exponents for all series are now around 0.54 for the first

Table 1. Descriptive statistics S&P and FTSEpricelevels and returns series.

	Real S&P500 Prices	Fake S&P500 Prices	Real FTSE100 Prices	Fake FTSE 100 Prices
Mean	1598.5	1568.5	6452.8	7015.0
Median	1471.5	1480.0	6532.4	7190.4
Minimum	676.53	491.25	4805.8	5185.8
Maximum	2690.2	2624.4	7877.4	7973.5
St. Deviation	459.78	491.60	700.42	581.01
Ex. Kurtosis	−0.89893	−0.99214	−0.95336	0.11942
Skewness	0.31735	0.097534	−0.084101	−0.99242
KPSS test with trend	4.29702***	3.74915***	0.746557***	1.61003***
Hurst exponent	1.01981	1.01209	1.0012	0.979511
	Log difference Real S&P500 returns	Log difference Fake S&P500 returns	Log difference Real FTSE100 returns	Log difference Fake RTSE100 returns
Mean	0.00023073	0.00022175	0.000057577	0.000044577
Median	0.00059556	0.00056296	0.00068184	0.00055157
Minimum	−0.094695	−0.11871	−0.11512	−0.042946
Maximum	0.10957	0.11413	0.086664	0.035216
St. Deviation	0.012664	0.014761	0.011971	0.0061047
Ex. Kurtosis	10.901	12.234	14.443	9.1317
Skewness	−0.34878	−0.0068651	−1.1542	−1.0981
KPSS test with trend	0.0700386	0.0531918	0.0367634	0.0440312
Hurst exponent	0.548056	0.539908	0.538714	0.593451

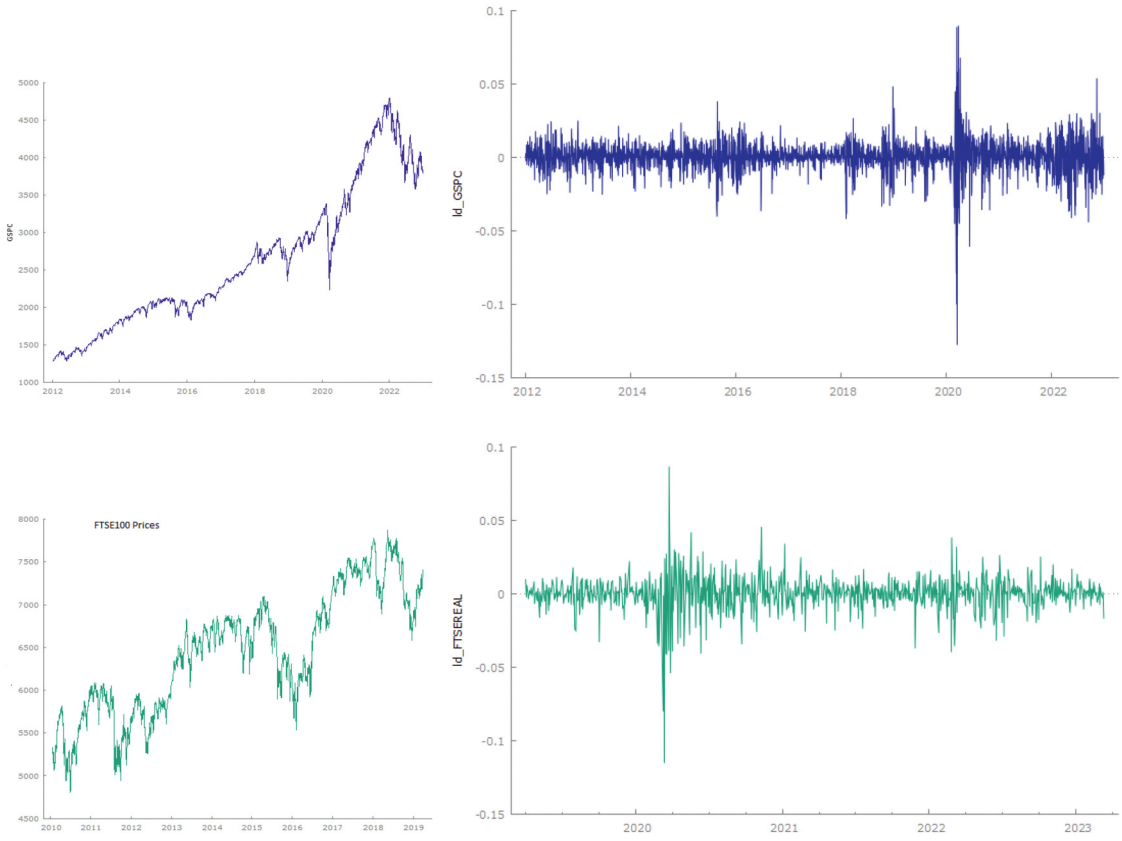

Figure 4. Plots of daily S&P500 and FTSE100levels and returns.

three series, with the fake FTSE100 return series with a slightly higher value of 0.59. These values suggest that any autocorrelations decay slowly with a tendency towards exhibiting long-memory. We fitted GARCH(1,1) models to the return series, estimated their periodograms and ran simple tests of market efficiency Bollerslev (1986). The GARCH(1,1) model results are shown in Table 2. The model failed to converge for the fake S&P500 series so an ARCH (1) model, Engle (1982), was estimated instead. The model was successful for both the real and fake FTSE100 series but the beta parameter for the fake FTSE100 series is much lower than usual, suggesting lower persistence. However, the conditional variances for the real and fake series behave similarly, though the model variance is more pronounced relative to the residuals in the fake series, as shown in Figure 5.

The relationship between the lag structure of the real and fake return series was examined by running a simple test of market efficiency, see Fama (1965), by regressing the current return for both series on one lag of itself. The results are shown in Table 3 and are quite striking. The real series exhibit behaviour that is consistent with weak-form market efficiency. The slope coefficient for the real S&P500 Index is negative and significant, but the Adjusted R-squared suggests that the relationship only explains 1% of the variation in real returns on the S&P 500 Index. The slope coefficient for the fake S&P 500 return series is also significant and large with a value of 0.47, and the

Table 2. GARCH(1,1) and ARCH(1) models fitted to real and Fake S&P 500 and FTSE100return series.

Coefficient	Real S&P500 returns	Fake S&P500 returns	Real FTSE100 returns	Fake FTSE100 returns
Constant	0.000615600***	n.a.	1350.29***	0.000477345***
Alpha(0)	2.27786e-06***	7.71906e-05***	145.623***	0.000003816***
Alpha(1)	0.119790***	0.761106***	0.960103 ***	0.624975***
Beta(1)	0.862700***	n.a.	0.0398970	0.264638 ***

***Indicates significance at 1% level.

Figure 5. Plots of conditional variances real and fake index Series.

Table 3. Market efficiency tests real and fake index returns series.

	Slope coefficient	Adjusted R-Ssquared
Real S&P500 return series	0.104274***	0.010515
Fake S&P 500 return series	0.467936***	0.218690
Real FTSE100 return series	0.0260714	0.000249
Fake FTSE100 return series	0.729418***	0.531648

***Indicates significance at 1% level.

Table 4. Engle-granger cointegration test.

	Slope coefficient Fake Series	Adjusted R-squared
Real S&P500 series	0.930453***	0.989720
Real FTSE100 series	1.06411***	0.984814

***Indicates significance at 1% level.

regression has an Adjusted R-Squared of 22%. This is not consistent with the existence of weak form market efficiency.

Similar problems are apparent in the results for the FTSE100 Index real and fake series. In the case of the real FTSE100 Index series, the regression of the return on one lag of itself is insignificant and results in a negligible Adjusted R square. In contrast, the fake FTSE100 Index regression produces a significant slope coefficient with a value of 0.73, and an Adjusted R square of 0.53. This clearly contravenes market efficiency.

The relationship between the levels of the real S&P 500 and fake S&P 500 series and the levels of the real and fake FTSE100 series was examined using an Engle and Granger (1987) bivariate cointegration test. The results in Table 4 show the slope coefficients are close to 1 in both cases and significant at the 1% level. The unit root test on the residuals of these regressions rejects the null of non-stationarity at the 1% level. These results provide strong evidence that the levels of the real and fake series, in both cases, are very similar.

Given the doubts raised by the fitting of GARCH models and the failure of weak form efficiency tests, in the case of the fake series, we decided to explore further using periodograms. There is evidence of slightly different behaviour in the lag structure of autocorrelations in the two return series. Figure 6 provides graphs of the periodograms of the two sets of series. A periodogram is an estimate of the spectral density of a signal. The term was coined by Schuster 1898. There is relatively more dependence in the fake

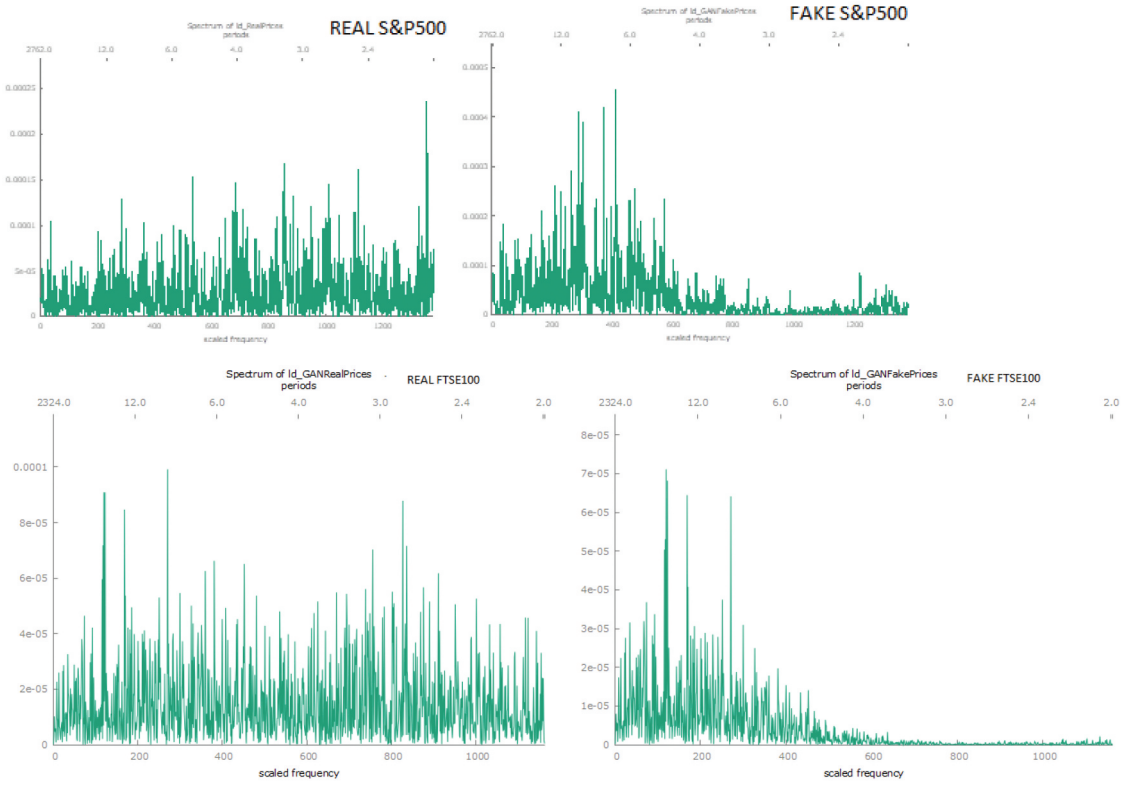

Figure 6. Periodograms real and fake return series.

series that then diminishes rapidly in comparison to the real series, in the cases of both indices.

Analysis using moments and characteristic functions

Given that the previous analysis had revealed differences in the behaviour of the lag structures of the real and GAN generated series, when converted into time series of continuously compounded returns, we decided to explore this further in relation to FTSE returns, by means of the second, third and fourth moments about the origin, of both the real and GAN generated series, and then augment this by applying Random Forest analysis to the same series.

Tables 5 and 6 show the results of regressing the real daily FTSE continuously compounded returns and the GAN generated FTSE synthetic returns on the second, third and fourth moments about the origin for these two sets of series.

The results in Table 5 confirm that there is a great deal of information in the first five lags of the second, third, and fourth moments about the origin of the real FTSE return. The second, third and fourth lags of the second moment about the origin are significant at the 5% level or better. The first, second and fourth lags of the third moment about the origin are significant at the 5% level, whilst the fourth moment about the origin, has third and fifth lags significant at the 5% level or better. The Adjusted R-squared of this regression equation is 7% and the F statistic is significant at a 1% level.

Table 6 reports a similar analysis for the GAN generated FTSE series.

The moments of the GAN generated synthetic FTSE daily returns series contain much more information than the real series. The first and second lags of the second moment about the origin are significant at the 5% level, the first and fourth lags of the third moment are also significant at the same level, whilst the first, second and fourth lags of the fourth moment about the origin are significant at a 1% level. The regression is highly significant, and the Adjusted R-squared is 44%, whilst for the real FTSE return series in Table 5, it was only 7%.

Random forest analysis

We used the R library packages 'randomforest', Liaw and Wiener (2002), and 'RandomForestExplained',

Table 5. OLS regression of FTSE daily continuously compounded returns on the second, third and fourth moments about the origin.

	Coefficient	Std. Error	*t*-ratio	p-value
const	0.000472157	0.000466099	1.013	0.3113
OFTSRETSQ_1	2.15379	1.59497	1.350	0.1772
OFTSRETSQ_2	3.34990	1.60408	2.088	0.0370
OFTSRETSQ_3	−7.42683	1.55599	−4.773	0.0000
OFTSRETSQ_4	−0.309003	1.68484	−0.1834	0.8545
OFTSRETSQ_5	−1.63797	1.69214	−0.9680	0.3333
OFTSRETCU_1	20.1945	10.8311	1.864	0.0626
OFTSRETCU_2	19.8593	10.7758	1.843	0.0656
OFTSRETCU_3	−18.5220	10.5445	−1.757	0.0793
OFTSRETCU_4	22.8636	11.0438	2.070	0.0387
OFTSRETCU_5	17.7148	11.2373	1.576	0.1153
OFTSERETSQSQ_1	225.629	182.473	1.237	0.2166
OFTSERETSQSQ_2	−293.767	182.185	−1.612	0.1072
OFTSERETSQSQ_3	386.654	175.787	2.200	0.0281
OFTSERETSQSQ_4	−3.08837	183.169	−0.01686	0.9866
OFTSERETSQSQ_5	368.773	186.900	1.973	0.0488
Mean dependent var	0.000039	S.D. dependent var	0.011995	
Sum squared resid	0.130210	S.E. of regression	0.011568	
R^2	0.084031	Adjusted R^2	0.069910	
$F(15, 973)$	5.950857	P-value(F)	5.57e−12	
Log-likelihood	3015.176	Akaike criterion	−5998.353	
Schwarz criterion	−5920.006	Hannan–Quinn	−5968.559	
$\hat{\rho}$	−0.005867	Durbin–Watson	2.009265	

OLS, using observations 2019-04-09 2023-03-10 (T = 989)
Dependent variable: FTSERET

Paluszynska et al., (2020), to analyse the four series of moments for the real and GAN generated series. The random forest models were fitted using regression weights.

In the interest of brevity, we shall not explore the results of the Random Forest analysis in great detail, but note that they have the advantage of utilizing non-linear methods of analysis.

We first fitted a Random Forest analysis to both the real daily FTSE return series and the GAN generated one and included five lags of daily returns and five lags of the second, third and fourth moments around the origin in the analysis. The results, for the real series, produced a mean value of the squared residuals of $1.039953e^{-10}$, and the model explained 75% of the variance. We then repeated the same analysis for the GAN generated synthetic series and the results had a mean value of the squared residuals of $1.832818e^{-10}$, and now explained 95.1% of the variance. This was

Table 6. OLS regression of the GAN generated FTSE synthetic daily continuously compounded returns on the second, third and fourth moments about the origin of the series.

	Coefficient	Std. Error	*t*-ratio	p-value
const	0.000385202	0.000184548	2.087	0.0371
OGANFTSERETSQ_1	−15.5783	3.84370	−4.053	0.0001
OGANFTSERETSQ_2	−8.40173	4.93569	−1.702	0.0890
OGANFTSERETSQ_3	5.61506	5.00488	1.122	0.2622
OGANFTSERETSQ_4	6.26030	4.66484	1.342	0.1799
OGANFTSERETSQ_5	4.92276	3.54511	1.389	0.1653
OGANFTSERETCU_1	1569.73	87.1355	18.01	0.0000
OGANFTSERETCU_2	−185.311	117.958	−1.571	0.1165
OGANFTSERETCU_3	118.154	121.429	0.9730	0.3308
OGANFTSERETCU_4	−314.233	115.799	−2.714	0.0068
OGANFTSERETCU_5	103.292	83.2402	1.241	0.2149
OGANFTSERETSQSQ_1	21709.0	2840.23	7.643	0.0000
OGANFTSERETSQSQ_2	8170.94	3033.06	2.694	0.0072
OGANFTSERETSQSQ_3	−3861.59	3003.05	−1.286	0.1988
OGANFTSERETSQSQ_4	−8200.21	2884.42	−2.843	0.0046
OGANFTSERETSQSQ_5	−3564.76	2700.30	−1.320	0.1871
Mean dependent var	0.000032	S.D. dependent var	0.006117	
Sum squared resid	0.020366	S.E. of regression	0.004575	
R^2	0.449072	Adjusted R^2	0.440579	
$F(15, 973)$	52.87417	P-value(F)	1.0e−114	

OLS, using observations 2019-04-09 2023-03-10 (T = 989)
Dependent variable: GANFTSERET

consistent with the OLS regression analysis which showed that the regression results for the synthetic FTSE daily return series had much higher explanatory power.

The analysis was then repeated, but in this analysis, we dropped the five lags of the real and synthetic daily FTSE return series and just included the lags of the moments about the origin. These results also confirmed the regression analysis. Figure 7 displays a plot of the minimum depth of the Random Forest analysis of the real FTSE daily returns using the same variables, including five lags, representing the moments about the origin applied in the regression analysis.

In a Random Forest analysis, minimal depth is the average distance between the root of a tree and the node/split where a given variable was used. Smaller values of the minimal depth indicate an early contribution of the variable, that is, more discriminating power.

It can be seen in Figure 7 that the third moment conveys the most information about the properties of the distribution of real daily FTSE returns, followed by the fourth moment. This makes intuitive sense, in that if the distribution was Gaussian, it would be defined by its first two moments. Given that financial return distributions are known to be peaked and fat-tailed, the third and fourth moments should convey information.

The OLS regression results reported in Table 5, suggest that there is more information in the fourth moment, in the sense that more lagged coefficients have greater significance, than those for the third. The Random Forest analyis, which has the advantage of being non-linear, suggests the reverse.

Figure 8 shows a similar analysis for the synthetic FTSE daily return series generated by the GAN.

Once again, it is the first lag of third moment about the origin that conveys the greatest information, followed closely by the first lags of fourth and second moments.

The Random Forest results are more clear cut than those of the regression analysis, in that they indicate clearly that, in the case of the synthetic return series, the first lag of the third moment, capturing relative skewness, is the most informative.

We have shown that GANs can be used to generate fake daily financial index series that closely mimic the behaviour of the real index series, particularly in the case of the levels, given that we used the levels of the indices to generate the fake index series. However, we transformed the series into continuously compounded return series and then explored in detail how the real and fake return series correspond to one another. The obvious question is what practical use can we make of this?

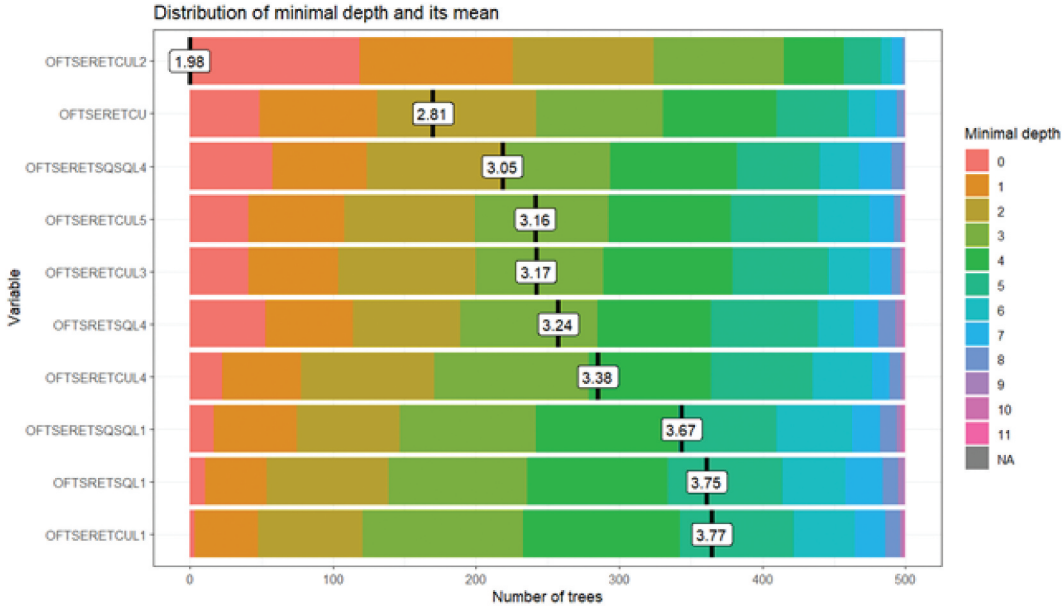

Figure 7. Random forest analysis minimum depth analysis of real FTSE daily returns.

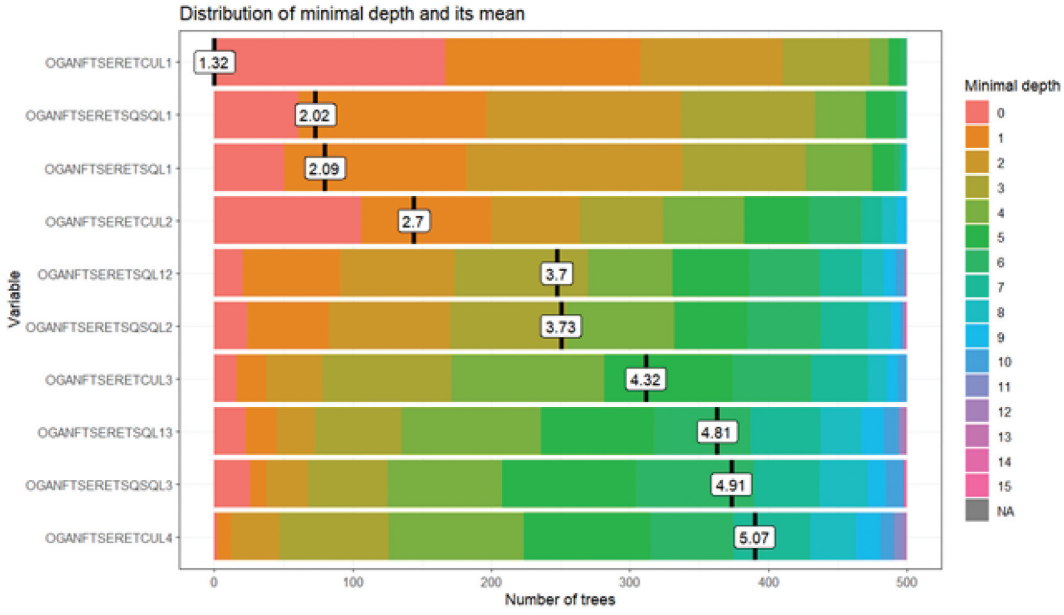

Figure 8. Random forest analysis minimum depth analysis of synthetic GAN generated FTSE daily returns.

One potential use is to generate fake but realistic data sets when we do not have a sufficient sample. There may be problems in gaining access to sufficient proprietary data sets. We have noted that in finance GANs have been applied for fraud detection or for fine-tuning trading strategies and that they may assist in determining factors driving price changes.

The GAN generated series and value at risk (var)

In the example that follows, we will employ the fake S&P 500 index series to generate an estimate of Value at Risk (VaR). We will use the GAN to make a prediction of the returns on the S&P500 Index in order to calculate the VaR for S&P500 for a period from 9 September 2019 until 22/12/22. Figure 9 shows the VaR for the real S&P500 Index over the same period and the VaR predicted by the GAN using data points from the predicted S&P500 Index series.

We note that this is a potentially optimistic exercise in that we have done a great deal of analysis in the previous sections analysing the different distributional characteristics of the real and synthetic GAN generated return series. Furthermore, in the most recent and comprehensive work on this topic, Cont et al. (2022) tailor their approach to explicitly modelling tail risk and combine it with investment strategies. Cont et al. (2022) suggest their proposed 'Tail-GAN', constitutes a novel approach for multi-asset market scenario simulation that focuses on generating tail risk scenarios for a user-specified class of trading strategies. In contrast to previous GAN-based market generators, which are trained using cross-entropy or Wasserstein loss functions, Tail-GAN starts from a set of benchmark trading strategies and uses a bespoke loss function to accurately capture the tail risk of these benchmark portfolios, as measured by their Value-at-Risk (VaR) and Expected Shortfall (ES). The exercise we use in this example uses the complete distribution and is not focussed on the specific modelling of tail risk.

Value-at-Risk (VaR) has been extensively embraced by regulators and practitioners in financial markets under the Basel II and III framework as the basis of risk measurement both for the purpose of ensuring regulatory capital adequacy and risk management and strategic planning at industry level. The Basel Accords refer to the banking supervision accords (recommendations on banking regulations) issued by the Basel Committee on Banking Supervision (BCBS). There have been a number of modifications over the years and the latest was in 2019, whilst the Basel Accords have been integrated into the consolidated Basel Framework, which comprises all of the current and forthcoming standards of the Basel

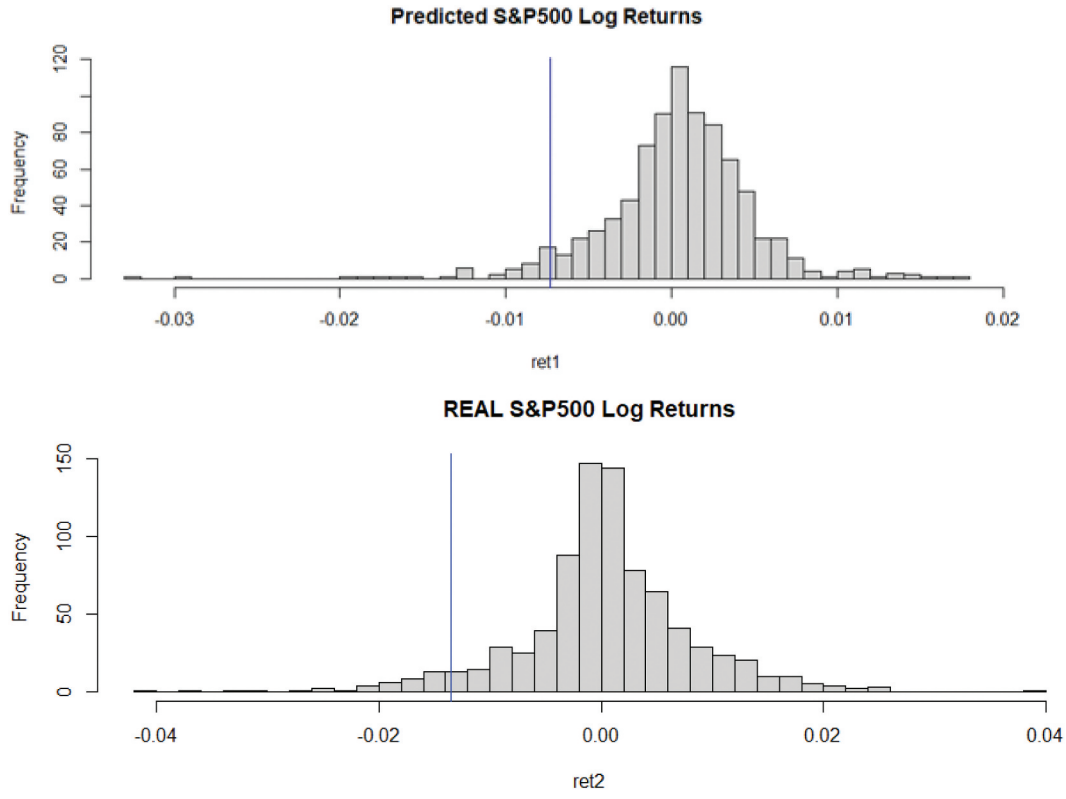

Figure 9. VaR calculations, predicted and actual S&P500 Index returns 9/9/2019 until 22/12/22.

Committee on Banking Supervision (see Basel Committee on Banking Supervision (2019)).

There are various ways of calculating VaR, but we will confine ourselves to a simple historical calculation, alternatives include the use of various modelling techniques and Monte-Carlo simulations.

We have calculated the 5% VaR for a $1000 invested for 5 days. If we take the predicted values from the GAN for this period, the mean return is 0.000329, the standard deviation is 0.004583777, and the calculated VaR for the suggested period is -$7.256978. If we take the actual returns for this period they have a mean value of 0.000360 and a standard deviation of 0.007872379. Thus, the actual standard deviation is almost double the prediction and the resultant 5% VaR is -$13.4579, which is almost twice as much.

Our results are not particularly accurate. However, it has to be borne in mind that we have calculated VaR for a period of around two and a quarter years. It is instructive to consider what took place in this period.

Our prediction period ran from 9 September 2019 until 22 December 22, and there were 829 predicted daily observations. On 13 March 2020, the then US President Donald Trump declared a national emergency. On the same day, a travel ban on non-US citizens travelling from Europe went into effect. Further stringent measures followed. Thus, the bulk of the period coincided with the COVID-19 pandemic and all the economic and social dislocations attached to it. Thus, it is not surprising the risk was under-estimated.

GANs appear to be very successful at replicating dataset characteristics, but perhaps unsurprisingly, less successful at forecasting. Perhaps, some sort of hybrid approach is required involving GANs augmented by time-series modelling, as suggested by Cont et al. (2022), if the focus is to be on tail risk.

IV. Conclusion

The paper examines the use of GANs to generate a fake S&P500 Index series plus a fake FTSE100 series, both of which in levels are indistinguishable from and cointegrated with the real series. It is only when the series is transformed into returns that higher-order lags of the two series behave differently. This is

apparent from fitting GARCH models, periodograms, and simple tests of weak form market efficiency. These tests reveal differences in the behaviour of the two series at the higher order lags.

To further explore these issues, we utilized the second, third and fourth moments about the origin and applied them in both regression and Random Forest analysis.

Goodfellow (2016) notes that there are no single compelling ways to evaluate a generative model. Models with good likelihood can produce bad samples, samples themselves are hard to evaluate, and good samples can have bad likelihood. Hence, the multiple metrics used in the evaluation in this paper.

It has to be borne in mind that we have set a high evaluation bar for the synthetic series produced by GANs, given that we transformed them into returns, and then ran most of the tests on the transformed series. Similarly, we used a radically different time period coinciding with the COVID-19 pandemic to estimate VaR. One of the problems is that nonstationary series are difficult to compare apart from by means of cointegration techniques. Nevertheless, the results in this paper suggest that much is to be gained from unsupervised learning techniques, as represented by the application of GANs.

Our interest in this topic was sparked by the fact that the construction and application of GANs is so unlike standard econometric modelling, yet they still seem to capture a great many features of the original financial data sets. It remains to be seen how useful GANs will prove to be in future empirical applications in finance.

Acknowledgements

A limited portion of the results in this paper were presented on 12 July at the MODSIM 2023 Conference held in Darwin, NT, Australia.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Arjovsky, M., S. Chintala, and L. Bottou. 2017. "Wasserstein Generative Adversarial Networks." *International Conference on Machine Learning*, 214–223, PMLR. https://proceedings.mlr.press/v70/arjovsky17a.html.

Basel Committee on Banking Supervision. 2019. *Explanatory Note on the Minimum Capital Requirements for Market Risk*. https://www.bis.org/bcbs/publ/d457.htm.

Bollerslev, T. 1986. "Generalized Autoregressive Conditional Heteroscedasticity." *Journal of Econometrics* 31 (3): 307–327. https://doi.org/10.1016/0304-4076(86)90063-1.

Breiman, L. 2001. "RandomForests." *MachineLearning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Cont, R., M. Cucuringu, R. Xu, and C. Zhang. 2022. "Tail-GAN: Learning to Simulate Tail Risk Scenarios." https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3812973.

Cornish, E. A., and R. A. Fisher. 1938. "Moments and Cumulants in the Specification of Distributions." *Review of the International Statistical Institute* 5:307–320.

Eckerli, F., and J. Osterrieder. 2021. "Generative Adversarial Networks in Finance: An Overview". *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3864965.

Engle, R. 1982. "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica* 50 (4): 987–1007. https://doi.org/10.2307/1912773.

Engle, R. F., and C. W. J. Granger. 1987. "Co-Integration and Error Correction: Representation, Estimation and Testing." *Econometrica* 55 (2): 251–276. https://doi.org/10.2307/1913236.

Fama, E. F. 1965. "The Behaviour of Stock Market Prices." *Journal of Business* 38 (1): 34–105. https://doi.org/10.1086/294743.

Fiore, U., A. D. Santis, F. Perla, P. Zanetti, and F. Palmieri. 2019. "Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection." *Information Sciences* 479:448–455. https://doi.org/10.1016/j.ins.2017.12.030.

Goodfellow, I. 2016. "NIPS Tutorial: Generative Adversarial Networks." arXiv:1701.00160. https://neurips.cc/virtual/2016/tutorial/6202.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. *Generative Adversarial Networks*. https://d i. rg/arXiv:1406.2661.

Ho, T. K. 1995. "Random Decision Forests." Proceedings of the 3rd International Conference on Document Analysis and Recognition, 278–282. Montreal, QC.

Koshiyama, A., N. Firoozye, and P. Treleaven. 2019. "Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination, arXiv preprint arXiv:1901.01751. (5). https://doi.org/10.1080/14697688.2020.1790635.

Kyle, A. S., and K. Todorov. 2023. "The Cumulant Risk Premium." *Bank for International Settlements, BIS Working Paper No 1128*.

Laplace, P.-S. 1774. "Mémoire sur la probabilité des causes par les évènements." *Mémoires de l'Academie Royale des Sciences Presentés par Divers Savan* 6:621–656.

Liaw, A., and M. Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22.

Martin, I. 2013. "Consumption-Based Asset Pricing with Higher Cumulants." *The Review of Economic Studies* 80 (2): 745–773. https://doi.org/10.1093/restud/rds029.

Mendel, J. 1991. "Tutorial on Higher-Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications." *IEEE Proc* 79 (3): 278–305. https://doi.org/10.1109/5.75086.

Nowozin, S., B. Cseke, and R. Tomioka. 2016. "F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization." *Advances in Neural Information Processing Systems* 29. arXiv:1606.00709. https://doi.org/10.48550/arXiv.1606.00709

Paluszynska, A., P. Biecek, and Y. Jiang. 2020. "_randomforestexplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance_. R Package Version 0.10.1." https://CRAN.R-project.org/package=randomForestExplainer.

Ryan, J. A., and J. M. Ulrich. 2023. "Quantmod: Quantitative Financial Modelling Framework, R Package Version 0.4.22." https://CRAN.R-project.org/package=quantmod.

Schuster, A. 1898. "On the Investigation of Hidden Periodicities with Application to a Supposed 26 Day Period of Meteorological Phenomena." *Terrestrial Magnetism* 3 (1): 13 41. Bibcode: 1898TeMag...3...13S. https://doi.org/10.1029/TM003i001p00013.

Summers, L. 1986. "Does the Stock Market Rationally Reflect Fundamental Values?" *The Journal of Finance* 41 (3): 591–601. https://doi.org/10.1111/j.1540-6261.1986.tb04519.x.

Takahashia, S., Y. Chena, and K. Tanaka-Ishiib. 2019. "Modeling Financial Time-Series with Generative Adversarial Networks." *Physica A: Statistical Mechanics and Its Applications*: 1–14. https://doi.org/10.1016/j.physa.2019.121261.

Wiese, M., R. Knobloch, R. Korn, and P. Kretschmer. 2020. "Quant GANs: Deep Generation of Financial Time Series." *Quantitative Finance* 20 (9): 1419–1440. https://doi.org/10.1080/14697688.2020.1730426.

Zhou, X., Z. Pan, G. Hu, S. Tang, and C. Zhao. 2018. "Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets." *Mathematical Problems in Engineering* 2018:1–11. https://doi.org/10.1155/2018/4907423.