National Research University Higher School of Economics

Faculty of Computer Science
Bachelor's Programme in Data Science and Business Analytics

THESIS PAPER
Research Project

The estimation of Value-at-Risk and Expected Shortfall based on deep generative models

Prepared by the student of Group 201 in Year 4,
Belonovskiy Peter Ilich

Thesis Paper Supervisor:
Associate Professor, Naumenko Vladimir Vladimirovich

Moscow
2024

Contents

Abstract

Value-at-Risk and Expected Shortfall are the most popular risk measures in the world. Financial institutes strongly benefit from the more accurate estimation of both measurers,due to substitutional cost savings and enhanced resilience to handle extreme scenarios. For a long time, simple approaches were used to estimate Value-at-Risk and Expected Shortfall. Their performance was acceptable but left room for improvement. In recent years, with the rapid development of Deep Learning, a specific type of models, called Deep Generative Models, were introduced for this problem. While there have already been some attempts to use Deep Generative Models for Value-at-Risk and Expected Shortfall estimation, in this thesis the completely unexplored approach of the usage of Denoising Diffusion Probabilistic Models, the subtype of Deep Generative Models, is researched. Results show promising performance of Denoising Diffusion Probabilistic Models in the tasks of Value-at-Risk and Expected Shortfall estimation, as well as potential directions for research and further improvements.

List of Key Words: Risk management, Value-at-Risk, Expected Shortfall, deep learning, Deep Generative Models, Denoising Diffusion Probabilistic Models, time series.

Introduction

Risk is one of the most important concepts in finance. Recent global financial crises, such as 2008 and COVID-19 in 2020, forced financial regulators to more attentively assess financial risk to make investors less exposed to sudden capital shortfalls.ă This resulted in the development of a mechanism that requires financial institutes to continuously calculate different risk measures and report them to the authorities. While there are different risk measures in the world, the two most common are Value-at-Risk (VaR) and Expected shortfall (ES). Value-at-Risk is the maximal portfolio loss that will not be exceeded with predefined probability over a certain period of time, while Expected shortfall is the expected loss above the VaR level. Both statistics became popular after the the introduction of the Basel II Accord in 2004, the set of laws and recommendations to banks. which relied on the VaR and ES to determine the stand-alone capital that financial institutes should allocate to protect from market risk exposures [1].

The integral part of both measures is the trustworthy loss distribution. In the ideal and unreal case of knowing the loss distribution exactly at a particular time, VaR and ES would be computed analytically. However, loss distribution is not known in advance, and it VaR and ES makes sense only if estimated for future time periods. This transfers the task of VaR and ES estimation to the prediction of a loss distribution or its parameters on future time steps. Common and conservative approaches tend to parametrize the distribution [23]. However, market data poses many stylized features that are impossible to assess with simple distributions [5]. Recent advances in deep learning (DL) and, more specifically, deep generative models, opened the door for novel approaches for the estimation of VaR and ES. The idea lies in scenario simulations, the process of generation of synthetic samples of a loss, which are extensively used in finance [15].

Deep generative models, such as VAEs, GANs, and Denoising Diffusions Probabilistic Models, have recently revolutionized the world of AI and gained enormous success in almost all applications that require generative abilities, such as image

synthesis, text-to-image and image-to-image conversion, in-painting, liquid simulation, and drug synthesis [6]. Their major strength comes from their superior ability to approximate complex multidimensional distributions with irregular shapes.ă This perfectly transfers to the financial datasets, which are popular for their heavy-tailed distributions with volatile clustering patterns. Recent works have already tried to use VAEs and GANs for the estimation of value at risk [13], [11], however, the applicability of Denoising Diffusions Probabilistic Models to the VaR and ES estimation remains unexplored.

The goal of this thesis is to research the applicability of Denoising Diffusions Probabilistic Models for the VaR and ES estimation. The novelty of the thesis lies in the usage of Denoising Diffusions Probabilistic models for VaR and ES estimation. This research is both relevant and significant, since more accurate estimates of VaR and ES will unlock for financial institutions substitutional cost reductions due to more rational risk-capital allocations.

The thesis is structured as follows: In the next chapter, the domain literature overview is presented. After that, two chapters describe the formal aspects of VaR and ES and Denoising Diffusions Probabilistic models. Then, the Data and Methodology chapter is presented. It describes the proposed approach to the estimation of VaR and ES with DDPMs, the data used in experiments, and some additional aspects of experiments. The Results and Analysis chapter presents the results of the experiments and their comprehensive discussion. The thesis is concluded with the Conclusion chapter, which discusses the results of the research and summarizes the achievements of this work.

Literature review

The topic of VaR and ES is well-versed in many scientific papers, such as [23] and [27]. These papers focus mainly on the traditional parametric methods of VaR and ES estimation. Successful attempts to estimate these risk measures with traditional deep learning were undertaken by [20] and [7]. Both used recurrent neural networks (RNNs) with probabilistic modification, which enabled the generation of samples. The pioneer of the usage of deep generative models for VaR and ES forecasts was [13], who was the first one to try GAN in this setting. He has used the traditional version of GAN and was the first one to beat benchmarks. Since then, many researchers have started to try GANs for VaR and ES estimations on different datasets. [30] tested GAN for an estimation of VaR at the portfolio level, and [29] modified the GAN to be more applicable to real financial fat-tailed distributions. Later on, [12] utilized another type of deep generative model, namely VAEs. Even though they beat benchmarks, results are inferior to GANs. Finally, there were not yet any scientific works that employed Diffusion Models for VaR and ES estimation. Nevertheless, diffusion generative models are widely used and are well studied in [21] and [6]. What is more significant, Denoising Diffusions Probabilistic models were a hot topic for recent research, and, as a result, several SOTA models for Denoising Diffusions Probabilistic models in time-series were developed. One such model is "Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting" (TimeGrad), which was a pioneer in using diffusion models for time-series forecasting. A comprehensive survey of denoising diffusion models for time-series is presented in [22].

Value-at-Risk and Expected Shortfall

Both Values-at-Risk and Expected shortfall are popular risk measures that are ubiquitously used around the world. Basically, VaR is the maximal loss the investment will encounter with a specified probability over a specified period of time. The phrase "1-day 5% VaR equals 100$" means that with 95% probability, the loss of an investment on the next day will not be more than 100$. VaR was first introduced in the 1980s and was popularized in the 1990s due to its simplicity and interpretability. For now, VaR is the gold standard of an industry and is used in many risk-management applications.

Nevertheless, VaR possesses some undesirable properties. Firstly, it does not take into account the severity of losses that are above the VaR level. This limitation is aptly captured by Einhorn and Brown's in [10], where they compared VaR to "an airbag that works all the time, except when you have a car accident". What is more, VaR is not an additive measure and cannot be straightforwardly split among the assets in a portfolio, which makes diversification harder. The expected shortfall effectively addresses both shortcomings. ES is defined as the expected loss of the portfolio above the VaR level. It varies for different loss sizes, even for investments with the same VaR, and could be easily split among the portfolio's assets. An example of VaR and ES is shown in Figure 3.1. The superiority of ES over VaR is shown in Figure 3.2.

3.1 Definition of Value-at-Risk and Expected Shortfall

Assume a portfolio V which has a value V_t at time t. Then, m-day α% Value-at-Risk is defined as:

$$P(V_m - V_0 < VaR_m^\alpha) = \alpha \tag{3.1}$$

The other way round, if L is a random variable for portfolio loss and $F_{L;t}$ is the

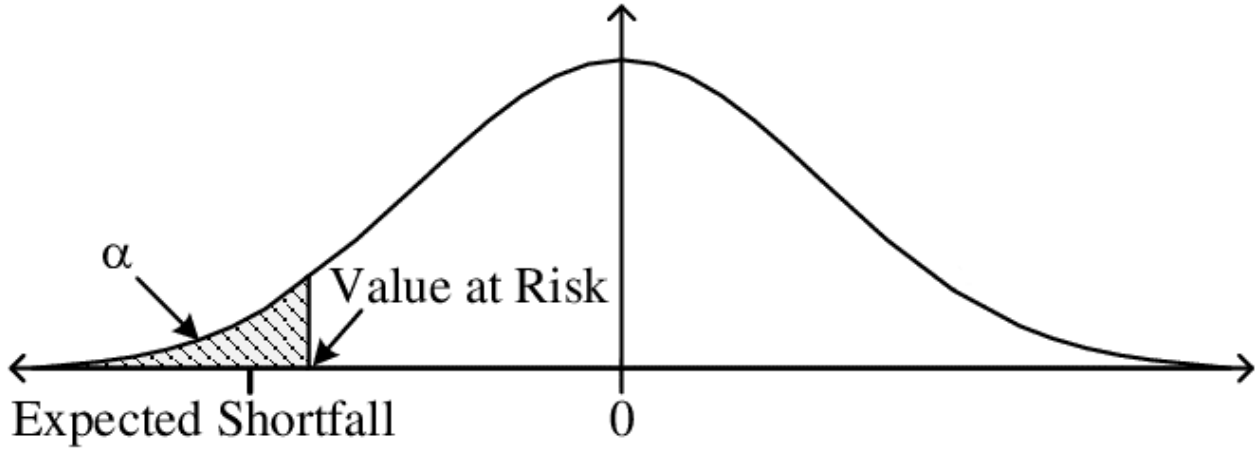

Figure 3.1: Value-at-Risk and Expected Shortfall.

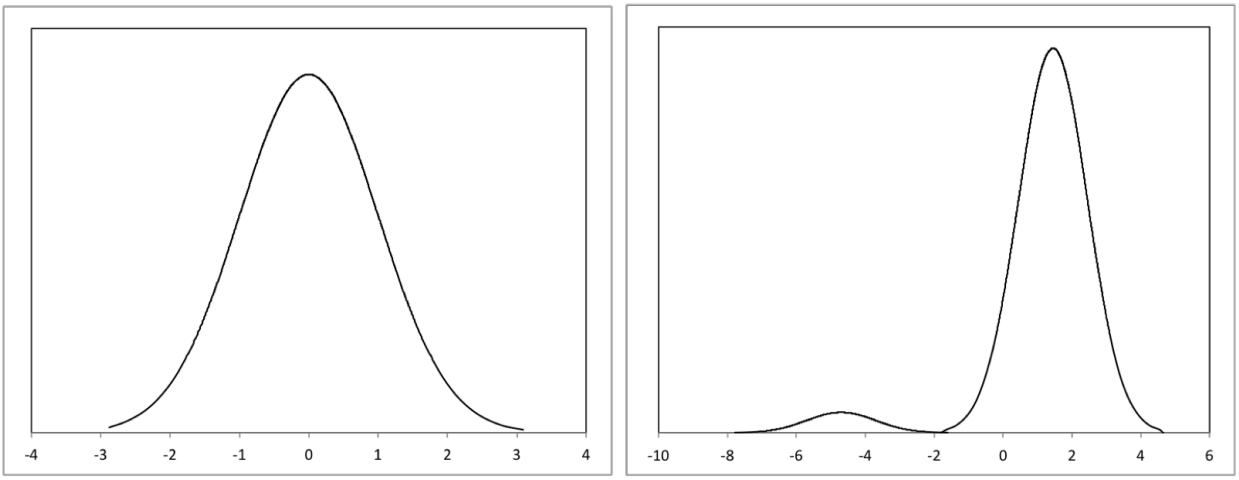

Figure 3.2: Two return distributions with the same 5% VaR of 1.65. Even though VaR is the same, the right plot is more risky. ES, on contrast, successfully captures this difference. ES of a right plot is higher than ES of a left plot.

CDF of the loss distribution at time t, than VaR can be defined as:

$$VaR_t^\alpha = F_{L;t}^{-1}(1-\alpha) \tag{3.2}$$

Which is simply $1-\alpha$ quantile of the loss distribution.

Expected shortfall boasts a more straightforward formulation. Intuitively, it addresses the question "If the worst-case scenario unfolds, how significant will our losses be?" ES is closely related to VaR, hence the definition:

$$ES_m^\alpha = \frac{\int_0^\alpha VaR_m^\gamma d\gamma}{\alpha} \tag{3.3}$$

where m stays for the time period.

From now on, it is clear that the key component of an estimation of VaR and ES is the determination of a loss distribution. The solution to the problem of estimation lies in predicting the future distribution of losses. The more accurately the predicted distribution fits the actual, the better the forecasts are.

3.2 Traditional approaches for VaR and ES estimation.

From the previous section, it is clear that the key task of an estimation of VaR and ES is the characterization of a probability density function of a loss distribution. The simplest approaches are parametric ones. They tend to describe the actual loss distribution by the family of common parametric probability distributions and estimate VaR and ES based on the fitted distribution. The most common method is the Variance-Covariance Method which models loss with a multivariate Gaussian distribution.

Variance-Covariance Method

Assume that our portfolio V consists of n stocks $X = \{X_1, X_2, ..., X_n\}$ with weights $w = \{w_1, w_2, ..., w_n\}$. Then fit the multivariate Gaussian distribution on portfolio loss,

$$L = -w^T X \sim \mathcal{N}(w^T \hat{\mu}, w^T \hat{\Sigma} w) \tag{3.4}$$

where μ and Σ are multivariate distribution parameters found by maximum likelihood estimation. Having the loss distribution, estimates of VaR and ES could be computed as follows [18]:

$$\widehat{VaR_\alpha} = -w^T \hat{\mu} + \sqrt{w^T \hat{\Sigma} w}\ \Phi^{-1}(\alpha) \tag{3.5}$$

$$\widehat{ES_\alpha} = -w^T \hat{\mu} + \sqrt{w^T \hat{\Sigma} w} \frac{\varphi(\Phi^{-1}(\alpha))}{1-\alpha} \tag{3.6}$$

where Φ and φ are the CDF and PDF of a distribution respectively. The main advantages of the method is the simplicity and speed.

Another simple method that is frequently used on practice is the historical simulation.

Historical Simulation

The method is very easy and straightforward. Assume you have N historical observations of a loss. Than, construct an order statistic based on that samples and estimate VaR and ES by [24]:

$$\widehat{VaR_\alpha} = L_m, \text{ where } m = \lfloor N(1-\alpha) \rfloor \tag{3.7}$$

$$\widehat{ES_\alpha} = \left(\sum_{i=\lfloor N\alpha \rfloor}^{N} L_i \right) / \lfloor N \cdot \alpha \rfloor \tag{3.8}$$

The formulas above correspond to the univariate scenario, when the portfolio consists of a single stock. If we have the portfolio P that consists of n stocks $X = \{X_1, X_2, ..., X_n\}$ with weights $w = \{w_1, w_2, ..., w_n\}$, we should firstly estimate individual VaR and ES and then aggregate them by the following formulas:

$$\widehat{VaR_P^\alpha} = \sqrt{VRV^T} \tag{3.9}$$

where $V = [w_1\widehat{VaR_{X_1}^\alpha}, w_2\widehat{VaR_{X_2}^\alpha}, ..., w_n\widehat{VaR_{X_n}^\alpha}]$ is a vector of weighted univariate VaR estimates and R is a correlation matrix computed on the returns of previous 90 days.

$$R = \begin{pmatrix} 1 & \rho_{1,2} & \dots & \rho_{1,n} \\ \rho_{2,1} & 1 & \dots & \rho_{2,n} \\ . & . & . & . \\ \rho_{n,1} & \rho_{n,2} & \dots & 1 \end{pmatrix}$$

Multivariate ES aggregation is much simpler:

$$\widehat{ES_P^\alpha} = w^T E \tag{3.10}$$

where $E = [\widehat{ES_{X_1}^\alpha}, \widehat{ES_{X_2}^\alpha}, ..., \widehat{ES_{X_n}^\alpha}]$ is a vector of univariate ES estimates.

These were some traditional approaches that would be used as a benchmarks in the experiments.

3.3 Backtesting

VaR and ES are measures, not metrics; therefore, approaches for the assessment of estimates are required. This is rather a complex question. Opposite to classical supervised ML tasks, we do not have ground truth values of VaR and ES because we never know accurately the loss distribution; hence, standard ML metrics such as *MSE, RMSE, Accuray, ROC-AUC* are inapplicable. Rather, hypothesis tests are used, and the comparison between different VaR and ES estimation algorithms is performed by the comparison of significances (p-values). Here, again, the tests for VaR are rather simple and straightforward, while ES tests are more complicated and, in fact, may not be that representative.

Tests for VaR

The main idea of VaR hypothesis tests is that the number of *exception*, the days at which the loss exceeded estimated VaR should not differ substantially from the significance level α. What is more, these exceptions should be independent, i.e previous exception should not influence further ones. Based on this assumptions, several tests were proposed [31]. Let $I_t = 1_{\{\widehat{VaR}_\alpha(t) < L_t\}}$ be and indicator function of an exception. Assume we have taken T periods for backtesting. Then, $M = \sum_{t=0}^{T} I_t$ be the total number of exceptions.

Kupiecs POF (1995) test

$$LR_{POF} = -2 \cdot ln \left(\frac{(1-\alpha)^{(T-M)} \alpha^M}{\left(1 - \frac{M}{T}\right)^{T-M} \frac{M}{T}^M} \right) \tag{3.11}$$

Under the null hypothesis of correct VaR estimate, the model is asymptotically χ^2 distributed with one degree of freedom. Higher p-value indicates better fit.

Time between failures test (Haas, 2001)

$$LR_{\text{tuff}} = -2 \sum_{i=1}^{M} \log \left(\frac{(1-\alpha)^{N_i - 1} \alpha}{\frac{1}{N_i} \left(1 - \frac{1}{N_i}\right)^{N_i - 1}} \right) \tag{3.12}$$

where N_i denotes the time gap between the $(i-1)$th violation and ith violation. The statistic is asymptotically chi-squared distributed with M degrees of freedom. Time between failures test is a more robust choice because it tests both the independence and coverage simultaneously.

Zero hypothesis of both tests accepts the estimates of VaR, hence higher p-values of hypothesis tests' correspond to better fits.

Tests for ES

As mentioned early, the situation with ES is more challenging. The reason for this is that ES, on contrast to VaR, does not posses import mathematical property called *elicitability* [16], which makes backtesting impossible. Nevertheless, later, several tests were still presented in [2]. The validity and reliability of these tests remain questionable. Let, again, $I_t = 1_{\{\widehat{VaR}_\alpha(t)<L_t\}}$ be and indicator function of an exception. Assume we have taken T periods for backtesting. Then, $M = \sum_{t=0}^{T} I_t$ be the total number of exceptions. Also, let L_t be the portfolio loss at time t.

Acerbi and Szekely first statistic (2014)

$$Z_1(L) = \frac{\sum_{t=1}^{T}\left(L_t I_t / \widehat{ES}_{\alpha,t}\right)}{M} + 1 \tag{3.13}$$

Acerbi and Szekely second test (2014)

$$Z_2(L) = \sum_{t=1}^{T} \frac{L_t I_t}{T\alpha \widehat{ES}_{\alpha,t}} + 1 \tag{3.14}$$

Both tests are significantly limited by the absence of a reference distribution, which makes hypothesis test impossible. However, estimates could still be compared to each other. Since the zero hypothesis of both tests is the acceptance of ES estimates, lower values of statistics correspond to better fits.

Deep Generative Models

Generative models are a type of machine learning models that model the data distribution $p(x)$, in contrast to common supervised learning tasks that model conditional discriminative probability $p(y|x)$. The generative task could be formulated as follows: *Having a training data from the distribution $q(x)$, the goal is to approximate this distribution by $p_\theta(x)$ to be able to sample new data similar to the training one*. Initially, generative models, such as GMMs and HMMs, were quite simple and were unable to capture the intricate patterns of data distribution. However, with the worldwide spread of Deep Learning in the last decade, the field of generative models underwent a radical transformation. Neural networks, famous for their ability to model complex feature transformations and interactions, became a versatile foundation for modeling the diverse and complex characteristics of a data distribution. Firstly, in 2013, VAE's were introduced and have primarily drawn the community's attention to deep generative models by showing outstanding results on generative tasks that were previously infeasible [19]. Then, in 2014, the architecture based on two adversarial neural networks called GAN was introduced [14]. This architecture has shown outstanding results in many generative tasks, especially image generation. GANs were the SOTA for generative tasks for many years, which is why first works on VaR and ES estimation were based on them. Finally, in 2020, Denoising Diffusion Probabilistic Models (DDPM) revolutionized the area of generative AI and achieved enormous success in many fields, ranging from computer vision to multi-modal learning and temporal data modeling [17], [9]. Time series forecasting is a special case of temporal data modeling. A DDPM leverages two Markov chains. The mission of the first one is to transform arbitrary data distributions into simple prior distributions by perturbing data to noise, and the mission of the second one is to transform noise back into data by learning the parametrized transition with deep neural networks. The typical scheme of a denoising diffusion is shown in Figure 4.1.

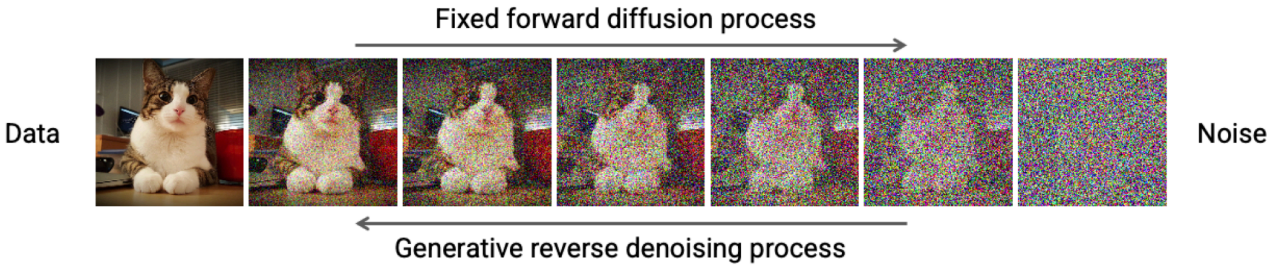

Figure 4.1: Typical Denoising Diffusion. Source: [22].

4.1 Denoising Diffusion Probabilistic Model

In this section, the preliminaries of diffusion models would be described. The forward process transforms and input x^0 to the noise x^K by gradually adding noise through K Markov Chain transitions. This is formally described as:

$$q\left(x^{1:K} \mid x^0\right) = \prod_{k=1}^{K} q\left(x^k \mid x^{k-1}\right) \text{ where } q\left(x^k \mid x^{k-1}\right) = \mathcal{N}\left(x^k; \sqrt{1-\beta_k}x^{k-1}, \beta_k \mathbf{I}\right) \tag{4.1}$$

$\beta_k \in [0,1]$ is a hyperparameter that determines the variance introduced at each step. x^k could ba achieved directly by the formulas of multiplications of Gaussians:

$$q\left(x^k \mid x^0\right) = \mathcal{N}\left(x^k; \sqrt{\alpha_k}x^0, (1-\alpha_k)\,\mathbf{I}\right) \tag{4.2}$$

where $\alpha_k = \prod_{i=1}^{k}(1-\beta_i)$.

The reverse process, on opposite, transforms noise back to data (x^K to x^0). By the following formula:

$$p_\theta\left(x^{0:K}\right) = p\left(x^K\right)\prod_{k=1}^{K} p_\theta\left(x^{k-1} \mid x^k\right) \text{ with } p_\theta\left(x^{k-1} \mid x^k\right) = \mathcal{N}\left(x^{k-1}; \mu_\theta\left(x^k, k\right), \sigma_k^2 \mathbf{I}\right) \tag{4.3}$$

Here $x^K \sim \mathcal{N}(0, \mathbf{I})$, $\sigma_k^2 = \frac{1-\alpha_{k-1}}{1-\alpha_k}\beta_k$ and $\mu_\theta\left(x^k, k\right)$, the removal of noise on k-th step, is modelled by a neural network. The illustration of a denoising diffusion process is shown in Figure 4.2.

Training the diffusion models is performed by minimising a KL-divergence loss.

$$\mathcal{L}_k = D_{KL}\left(q\left(x^{k-1} \mid x^k\right) || p_\theta\left(x^{k-1} \mid x^k\right)\right) \tag{4.4}$$

Intuitively, KL-divergence is smaller for the distributions that are close to each other,

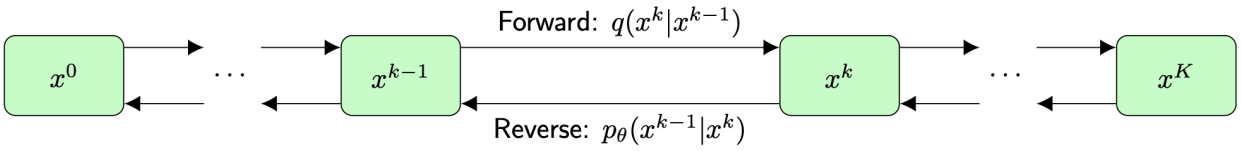

Figure 4.2: Illustration of a Denoising Diffusion process. Source: [22].

hence the minimization of loss implies finding the parametrized reverse distribution that matches the inverse of the forward noise adding distribution.

There are some additional empirically found technical details, mostly designed for the ease of training, such as the reparametrization of the $\mu_\theta\left(x^k, k\right)$ and alternative loss functions, however they are omitted here and could be found in [22].

4.2 Conditioning Diffusion Models

In some cases control over the generation is required. For example, generation of an image based on the text prompt, or, in our case, generation of loss distribution based on the observed portfolio values. To achieve this, the following approach have been developed.

Conditional Denoising Model

Conditional Denoising Model simply incorporates condition data $\mathbf{c}$ as an additional input to the denoising neural network:

$$p_\theta\left(x^{0:K} \mid \mathbf{c}\right) = p\left(x^K\right) \prod_{k=1}^{K} p_\theta\left(x^{k-1} \mid x^k, \mathbf{c}\right) \tag{4.5}$$

$$\text{with } p_\theta\left(x^{k-1} \mid x^k, \mathbf{c}\right) = \mathcal{N}\left(x^{k-1}; \mu_\theta\left(x^k, k \mid \mathbf{c}\right), \sigma_k^2 \mathbf{I}\right)$$

4.3 Diffusion Models for Time Series

Financial data is usually represented by the time-series, a sequence of evenly spread data points in time. VaR and ES are usually used for portfolios of liquid assets, such as stocks and FX, which are characterized by their price changing in time, which is apparently an example of time series. Having discussed the general paradigm of DDPMs, now it is time to delve into their usage with time series data.

First of all, the conditioning is performed on the previous historical observations, more specifically on their encoded representations. This conditioning could be implemented in multiple ways as stated in 4.2. Let X_{for} denote the forecast and X_{his} denote historical observations. Then Formula 4.3 looks like:

$$p_\theta\left(X_{for}^{0:K}|X_{his}\right) = p\left(X_{for}^{K}\right)\prod_{k=1}^{K} p_\theta\left(X_{for}^{k-1} \mid X_{for}^{k}, X_{his}\right) \text{ where } p\left(X_{for}^{K}\right) \sim \mathcal{N}(0, \mathbf{I}) \tag{4.6}$$

It is the overall idea of incorporating time series data in DDPMs. In the ongoing section the details on one particular implementation of DDPM for time series would be discussed.

4.4 TimeGrad

TimeGrad [4] is an autoregressive denoising diffusion model for multivariate probabilistic time series forecasting. This architecture was the first one to use DDPM in time series. TimeGrad is a conditional denoising model that utilizes basic denoising diffusion principles.

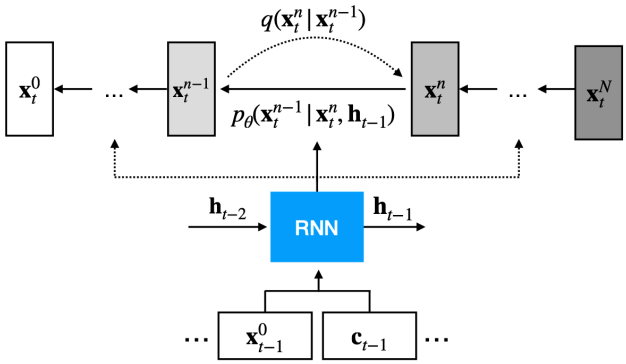

Figure 4.3: The scheme of a TimeGrad. Source: [4].

The overall scheme of a TimeGrad is shown in Figure 4.3. Conditional data, which includes historical observations and other features, is encoded using RNN and is added in the reverse process.

$$\mathbf{h}_t = \text{RNN}_\theta\left(\text{concat}\left(x_t^0, c_t\right), \mathbf{h}_{t-1}\right) \tag{4.7}$$

The denoising neural network $\mu_\theta\left(x^k, k, \mathbf{h}_t\right)$ in TimeGrad is a module with residual connections consisting of several sophisticated convolutional blocks. It finds useful relations between the historical observations, the diffusion time step, and the current

sample. Convolutional operation allows to grasp the interactions between different channels, making possible simultaneous multivariate time series probabilistic forecasting, which would be very useful in the financial domain, where portfolios consisting of many assets are modeled. The predictions are made in an autoregressive manner, meaning that to make predictions on $T + 1$, model predictions on T are used.

$$p_\theta\left(X_{for}^{0:K} \mid X_{his}\right) = \prod_{t=1}^{F} p\left(x_t^K\right) \prod_{k=1}^{K} p_\theta\left(x_t^{k-1} \mid x_t^k, \mathbf{h}_{t-1}\right) \tag{4.8}$$

The training is performed by sampling random windows of data from time series, splitting on test observation and context part, then sampling the random diffusion step k and minimizing the KL-divergence by 4.4. The test observation is taken as the real one, while the model tries to generate this sample as close as possible based on the historical observation and learned latent factors of distribution.ăThis model have achieved state-of-the-art results on several benchmarks, and remains the leading figure in the sphere of DDPMs for time series.

Data and methodology

5.1 Data

Diffusion models, like any other deep learning models, require a large amount of data for training. This may be a limitation of DDPMs for finance, since usually financial data is obtained on a daily basis, which makes its frequency too sparse. Nevertheless, by taking larger training periods and larger context windows, this limitation could be potentially overcome.

While there are many financial instruments, including stocks, forwards, bonds, commodities, options, etc., the most basic one, stock, was chosen for experiments. The argument is to test the performance of DDPM on a most widely used benchmark to grasp the general aspects of models' performance. The dataset consists of the daily prices of the 89 largest S&P500 companies in the date range $[2005-01-01, 2023-12-31]$. The data does not contain missing values, and the prices are adjusted for dividends. To stabilize the training, prices were transformed to the compound returns $r_t = ln\frac{P_t}{P_{t-1}}$ and normalized by standard scaling. The adjusted prices and compound returns of an Amazon stock are shown in Figure 5.1 as an example.

Finally, the data was split on 3 disjoint sets:

- **Training part** - $2005-01-03$ to $2020-12-31$ (4028 days in total)
- **Validation part** - $2021-01-01$ to $2022-05-31$ (355 days in total)
- **Test part** - $2022-06-01$ to $2023-12-31$ (398 days in total)

The estimation of scaling parameters and model training were performed on the training part. Tuning was performed on the validation part, and the final evaluation of performance was conducted on the test part.

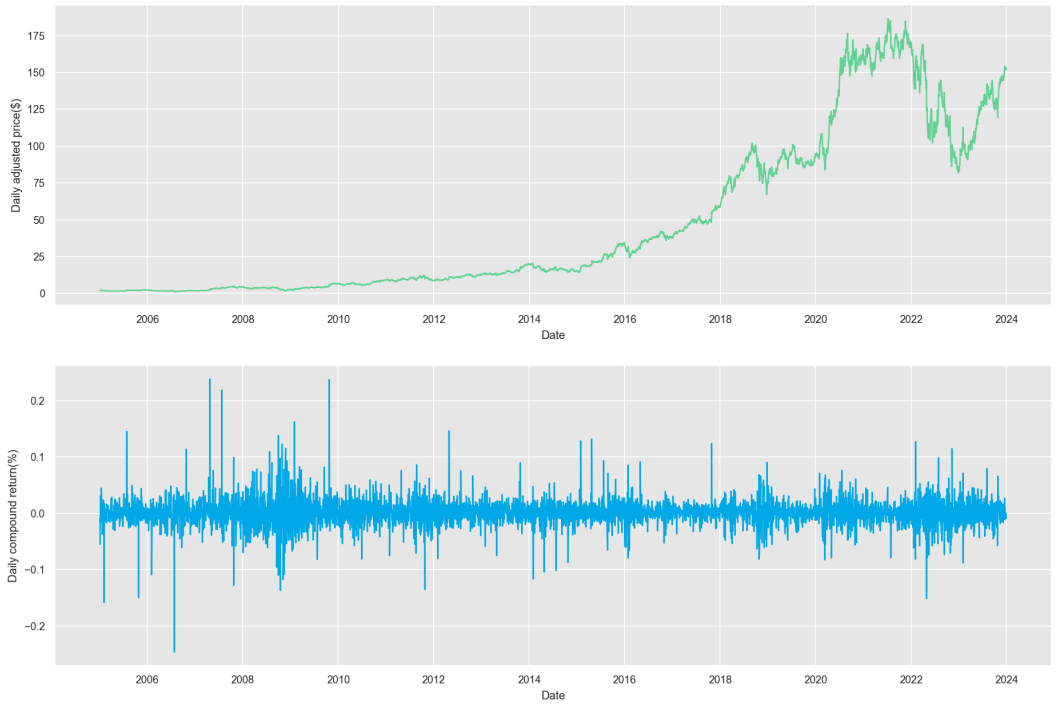

Figure 5.1: The adjusted prices and compound returns of an Amazon stock.

5.2 Proposed approach

The goal of this thesis is to research the possibility of an estimation of VaR and ES with Denoising Diffusion Probabilistic models. For this, TimeGrad architecture, as the most dominant representative of Denoising Diffusion Probabilistic models for time series, will be taken and modified. The main task is to accurately predict profit and loss distribution. For this, the generative abilities of TimeGrad will be utilized. By sampling 500 different scenarios of future return movements at each time step, more reliable profit and loss distributions, adjusted for the latent temporal effects, could be derived. TimeGrad successfully utilizes both the data about previous (or context) returns and the learned general aspects of a profit and loss distribution to produce diverse complex densities that capture various effects of return movements. The algorithmic procedure for an estimation of VaR and ES with TimeGrad is presented in Algorithm 1. An example of an estimation of VaR and ES by the proposed approach is shown in Figure 5.2

The approach is suitable for both univariate and multivariate estimation since TimeGrad is able to perform simultaneous multi-time series generation. The long-

Algorithm 1 Estimation of 1-day VaR and 1-day ES with TimeGrad

```
Fit TimeGrad on the training data
for all i in test period do
    Generate 500 observations of a loss at day i

    Estimate the VaR and ES based on the order statistics of generated samples
    similar to 3.2
end for
```

short-term memory network (LSTM), famous for its ability to analyze long sequences, was adopted as a recurrent neural network in TimeGrad. The default values of **50** and **2** were taken for the hyperparameters *hidden_size* and *num_layers*, respectively. The model training was performed with the *ADAM* optimizer with a learning rate of $\mathbf{1 \cdot 10^{-5}}$.

5.3 Modelling methodology

1-day is the most common prediction interval for both VaR and ES [26], hence 1-day VaR and ES estimates would be modeled. To get a more holistic assessment, the comparison will be conducted for both univariate and multivariate cases and on two significance levels $\alpha = \{0.05, 0.01\}$. In the univariate case, a portfolio consisting of 10 stocks is constructed, and its returns are taken as a single-time series. In the multivariate scenario, the same portfolio of 10 stocks is constructed; however, the returns of each stock in the portfolio are modeled separately. To get an unbiased comparison, the results would be averaged across five runs with different combinations of stocks in the portfolio. The combination of stocks in each portfolio is selected randomly. A context size of **90** days was chosen for all models, meaning that each prediction is made on the basis of **90** most recent return observations.

5.4 TimeGrad Parameters and Tuning

TimeGrad, similarly to any deep learning model, depends on many hyper parameters. Some of them, related to the LSTM, were discussed earlier. Still, there are many others that guide the diffusion process. This includes the variance injected at each step β_k, the number of diffusion steps K , learning rate and many more. The default values for these parameter were set as follows:

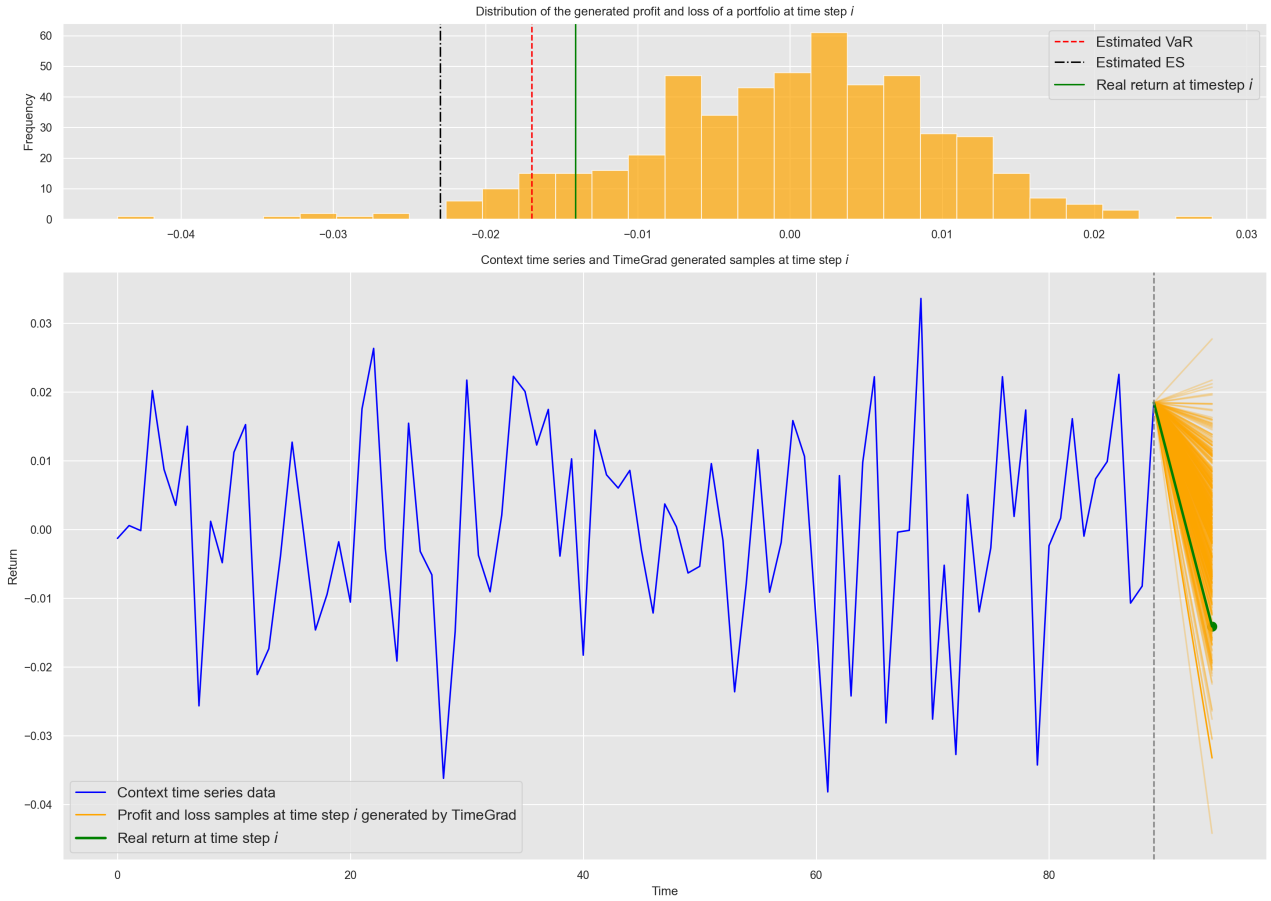

Figure 5.2: Demonstration of an estimation of VaR and ES with TimeGrad. At each test time step i, 500 profit and loss samples are generated by TimeGrad based on the context data. VaR and ES are estimated by the order statistics of the distribution.

- $\beta_k = 0.05$
- $K = 30$
- $lr = 0.001$
- $n\_epochs = 50$

Clearly, these values may not be optimal for the best performance. Thus, to obtain superior results, these hyperparameters should be tuned to find the best combination. The tuning was performed for both the univariate and multivariate cases on the validation data. Then, the model with the best parameters was trained again on the train and validation part and compared to others. Since there are 4 different measures used for the comparison and the tuning could be performed on the basis of only one measure, the objective value for maximization was chosen as an average of *Kupicks POF* p-value and *Haas TBF* p-value. It is worth mentioning here that such a scenario is targeted only at improvements in the VaR estimates, which does not surely imply

improvements in the ES estimates. This would be discussed further in the Results and Analysis section. Tuned values of parameters could be found in notebooks in repository.

5.5 Code implementation

TimeGrad model, as well as Historical simulation and Variance Covariance models were implemented via the Python programming language and the PyTorch library [3]. Diffusion steps in TimeGrad were constructed with the HuggingFace Diffusers library [8]. Additionally, Numpy library was used to implement the statistical tests, Optuna [25] to perform tuning and pandas, Seaborn and Scikit-learn [28] to conduct the experiments. The repository with all the code, experiments and visualisations could be found here: https://github.com/BELONOVSKII/var_es_dgm

Results and Analysis

6.1 Univariate scenario

	Kupicks POF ↑	Haas TBF ↑	Ace. and Sze. 1 ↓	Ace. and Sze. 2 ↓
TimeGrad	0.0049	0.0965	**1.918**	**0.0019**
TimeGrad tuned	0.4171	**0.4243**	1.9479	0.0046
Hist. Simulation	**0.4723**	0.2953	2.0062	0.0041
Var. Cov.	0.2016	0.2664	2.0258	0.0035

Table 6.1: Results of a 1-day univariate 5% VaR and ES estimation. Arrows indicate the direction of improvement. The best values are emphasized with bold.

	Kupicks POF ↑	Haas TBF ↑	Ace. and Sze. 1 ↓	Ace. and Sze. 2 ↓
TimeGrad	0.1401	0.275	**1.9525**	**0.00005**
TimeGrad tuned	**0.7062**	**0.5437**	2.009	0.0002
Hist. Simulation	0.4749	0.3584	2.0062	0.0002
Var. Cov.	0.6534	0.404	2.0765	0.0002

Table 6.2: Results of a 1-day multivariate 1% VaR and ES estimation. Arrows indicate the direction of improvement. The best values are emphasized with bold.

From Table 6.1, it can be seen that the proposed approach outperforms benchmarks on univariate 5% estimation on almost all tests. Default TimeGrad appears to be superior in ES estimation, which follows from the best results on both Ace. and Sze. 1 and Ace. and Sze. 2 tests. At the same time, the default TimeGrad shows bad results in the VaR estimation. However, tuned TimeGrad is, on contrast, superior at VaR estimation, which is reinforced by the best by far result on the Haas TBF test among all models while performing almost at par with Historical simulation method on the Kupicks POF. It should be noted that Haas TBF is a more robust test for an estimation of VaR; hence, tuned TimeGrad is better at VaR estimation than Historical simulation method. The Covariance method is inferior to other models on all tests.

Table 6.2 shows the results of a univariate estimation on 1%. Here, the dominance of TimeGrad is more impressive. The default version is by far the best in the ES estimation, while tuned TimeGrad is superior at VaR estimation. The only change is that the second best model here is Variance Covariance. Figure 6.1 shows estimated VaR and ES for all models.

An interesting inference could be made here. As mentioned in Section 5.4, the tuning was performed on the basis of improvement of the *Kupicks POF Haas TBF* tests. Such a scenario, apparently, corresponds to finding the optimal values for VaR estimates, and indeed, tuned TimeGrad is the best in VaR estimation. However, ES estimates of tuned TimeGrad are worse than the estimates of the default TimeGrad. This brings us to an interesting conclusion: VaR and ES estimates are somehow inversely related for TimeGrad, meaning that tuning for one measure leads to deterioration in an estimation of another measure. This non-obvious behavior could be studied in further works.

6.2 Multivariate scenario

	Kupicks POF ↑	Haas TBF ↑	Ace. and Sze. 1 ↓	Ace. and Sze. 2 ↓
TimeGrad	0.2934	**0.2629**	2.5789	0.008
TimeGrad tuned	0.4659	0.0748	2.6852	0.0057
Hist. Simulation	0.4584	0.035	2.4013	0.0072
Var. Cov.	**0.5942**	0.1037	**2.0205**	**0.0044**

Table 6.3: Results of a 1-day multivariate 5% VaR and ES estimation. Arrows indicate the direction of improvement. The best values are emphasized with bold.

	Kupicks POF ↑	Haas TBF ↑	Ace. and Sze. 1 ↓	Ace. and Sze. 2 ↓
TimeGrad	0.2162	**0.4407**	3.1863	0.0004
TimeGrad tuned	**0.7805**	0.3604	3.3593	0.0004
Hist. Simulation	0.4907	0.4092	2.894	**0.0003**
Var. Cov.	0.000004	0.0001	**1.7903**	0.0008

Table 6.4: Results of a 1-day multivariate 1% VaR and ES estimation. Arrows indicate the direction of improvement. The best values are emphasized with bold.

The results of a multivariate estimation performance of models are presented in Tables 6.3 and 6.3. Overall, the dominance of TimeGrad is less in multivariate scenario; however, it still outperforms other models on several metrics. On a 5%

Figure 6.1: Univariate VaR and ES estimates for the test period. Exceptions are highlighted with red stars.

estimation, TimeGrad is the best on a Haas TBF test, which implies good estimation performance of VaR by both coverage and independence. In a 1% scenario, TimeGrad shows complete superiority on VaR estimation by outperforming other models on both Kupicks POF and Haas TBF tests. The notable observation here is that TimeGrad performs well in a multivariate scenario even without the tuning. Indeed, the metrics of a default TimeGrad are more balanced than the metrics of the tuned TimeGrad in both 1% and 5% cases. As for the ES estimation, Varaince Covariance method appears to be the best, while TimeGrad is far behind it. Historical simulation method is inferior to other models in all tests except for Ace. and Sze. 2 in 1% estimation. Figure 6.2 shows estimated VaR and ES for all models. By comparing ther Figures 6.1 and 6.2, it can be seen that multivariate scenario produces more smooth results. This comes from the aggregation among assets which reduces fluctuations in the final time series.

Figure 6.2: Multivariate VaR and ES estimates for the test period. Exceptions are highlighted with red stars.

6.3 General Results

Overall, the proposed approach based on the TimeGrad shows promising results for VaR and ES estimation. The dominance of TimeGrad is clear in the univariate scenario and is slightly less pronounced in a multivariate case. In some cases, TimeGrad and tuned TimeGrad achieve almost twice the improvements in metrics. The most remarkable aspect of TimeGrad performance is its total supremacy in the Haas TBF test. In all 4 cases: 6.1, 6.2, 6.3, 6.4, TimeGrad is best on this test. Haas TBF is a more robust test for VaR estimates than Kupick POF test, because it seeks both for coverage and independence, so even the lags behind other models on Kupick POF test could be stated as insignificant and written off as randomness. This shapes the superior quality of DDPMs for VaR estimation and the prospective quality for ES estimation.

Another conclusion that could be drawn after experiments is the dependence of TimeGrad performance on hyper parameters. This is inherited from the deep learning nature of TimeGrad and Denoising Diffusion Probabilistic models in general, where a question of a proper hyper parameter choice is a famous pain. What is more, TimeGrad requires sufficient computational resources for training and inference compared to the alternatives. While Variance Covariance and Historical Simulation methods could be executed in a blink at an ordinary *CPU*, TimeGrad requires *GPU* and significantly more time on training. All experiments in this thesis were conducted on a Mac *M1 PRO GPU*.

Conclusion

In this thesis, the applicability of a type of Deep Generative models - Denoising Diffusion Probabilistic models for Value-at-Risk and Expected Shortfall estimation was researched. Firstly, the general theoretical knowledge about both risk measures was assembled and structured. This includes the discussion about their comprehensive comparison and applicability in particular situations. Next, the existing traditional models for an estimation of Value-at-Risk and Expected Shortfall were comparatively discussed, shaping the advantages and disadvantages of each. These traditional approaches serve as a benchmark for further experiments. Moreover, various statistical tests for both Value-at-Risk and Expected Shortfall were discussed.

Then, the theory behind Denoising Diffusion Probabilistic models was investigated. The modifications of these types of models for time series analysis were also well studied. Finally, the state-of-the-art Denoising Diffusion Probabilistic model for time series called TimeGrad was selected for experiments on VaR and ES estimation. An approach to an estimation of VaR and ES with TimeGard was proposed and described in detail.

Portfolios of stock returns were chosen as a target time series for experiments. Experiments included multiple runs with various setups, which ensured the absence of bias and spurious results. Overall, the proposed approach has shown promising and, in some cases, even superior performance, twice exceeding alternative models in particular metrics. The most notable finding is the total dominance of the proposed approach on the Haas TBF test, which is explained by the best quality of the produced VaR estimates. Nevertheless, some shortcomings, such as the strong dependence on hyper parameters, training time, and the great need for computational resources, were discovered in the proposed approach.

Overall, this thesis opened the door for research on the estimation of Value-at-Risk and Expected Shortfall with Denoising Diffusion Probabilistic Models. The obtained results show promising opportunity for obtaining better VaR and ES estimates by using Deep Generative models. This is useful both for financial institutions, which

could benefit from more efficient risk-capital allocations, and for the for the scientific community, which could further research and develop the findings presented in this thesis. Two major directions for further research include the usage of other assets, such as bonds, futures, and currencies, in the target portfolio and the usage of more sophisticated architectures of Denoising Diffusion Probabilistic models for time series.

Bibliography

1. A note on Basel III and liquidity / B. De Waal [et al.] // Applied Economics Letters. — 2013. — Vol. 20, no. 8. — P. 777–780.
2. *Acerbi C.*, *Szekely B.* Back-testing expected shortfall // Risk. — 2014. — Vol. 27, no. 11. — P. 76–81.
3. Automatic differentiation in PyTorch / A. Paszke [et al.]. — 2017.
4. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting / K. Rasul [et al.]. — 2021. — arXiv: 2101.12072 [cs.LG].
5. *Cont R.* Empirical properties of asset returns: stylized facts and statistical issues // Quantitative finance. — 2001. — Vol. 1, no. 2. — P. 223.
6. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models / S. Bond-Taylor [et al.] // IEEE transactions on pattern analysis and machine intelligence. — 2021. — Vol. 44, no. 11. — P. 7327–7347.
7. DeepVaR: a framework for portfolio risk assessment leveraging probabilistic deep neural networks / G. Fatouros [et al.] // Digital Finance. — 2022. — Apr. — Vol. 5. — DOI: 10.1007/s42521-022-00050-0.
8. Diffusers: State-of-the-art diffusion models / P. von Platen [et al.]. — 2022. — https://github.com/huggingface/diffusers.
9. Diffusion Models: A Comprehensive Survey of Methods and Applications / L. Yang [et al.]. — 2024. — arXiv: 2209.00796 [cs.LG].
10. *Einhorn D.*, *Brown A.* Private profits and socialized risk // Global Association of Risk Professionals. — 2008. — Vol. 42. — P. 10–26.
11. Estimating the Value-at-Risk by Temporal VAE / R. Sicks [et al.]. — 2021. — arXiv: 2112.01896 [cs.LG].
12. Estimating the Value-at-Risk by Temporal VAE / R. Buch [et al.] // Risks. — 2023. — Apr. — Vol. 11. — P. 79. — DOI: 10.3390/risks11050079.

13. *Fiechner L. B.* Risk Management with Generative Adversarial Networks : Master's thesis / Fiechner Lucas Benedikt. — University of Oxford, 2019.

14. Generative Adversarial Networks / I. J. Goodfellow [et al.]. — 2014. — arXiv: 1406.2661 [stat.ML].

15. *Glasserman P.* Monte Carlo methods in financial engineering. Vol. 53. — Springer, 2004.

16. *Gneiting T.* Making and evaluating point forecasts // Journal of the American Statistical Association. — 2011. — Vol. 106, no. 494. — P. 746–762.

17. *Ho J.*, *Jain A.*, *Abbeel P.* Denoising Diffusion Probabilistic Models. — 2020. — arXiv: 2006.11239 [cs.LG].

18. *Kim J.-H.*, *Park H.-Y.* Estimation of VaR and Expected Shortfall for Stock Returns // Korean Journal of Applied Statistics. — 2010. — Aug. — Vol. 23. — P. 651–668. — DOI: 10.5351/KJAS.2010.23.4.651.

19. *Kingma D. P.*, *Welling M.* Auto-Encoding Variational Bayes. — 2022. — arXiv: 1312.6114 [stat.ML].

20. *Lille W. K.*, *Saphir D.* Value at Risk Estimation with Neural Networks. A Recurrent Mixture Density Approach / Lille William Karlsson, Saphir Daniel. — KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ENGINEERING SCIENCES, 2021.

21. *Luo C.* Understanding Diffusion Models: A Unified Perspective. — 2022. — arXiv: 2208.11970 [cs.LG].

22. *Meijer C.*, *Chen L. Y.* The Rise of Diffusion Models in Time-Series Forecasting. — 2024. — arXiv: 2401.03006 [cs.LG].

23. *Nadarajah S.*, *Chan S.* Estimation Methods for Value at Risk: A Handbook of Extreme Value Theory and its Applications //. — 10/2016. — P. 283–356. — ISBN 9781118650196. — DOI: 10.1002/9781118650318.ch12.

24. *Nadarajah S.*, *Zhang B.*, *Chan S.* Estimation methods for expected shortfall // Quantitative Finance. — 2014. — Feb. — Vol. 14. — DOI: 10.1080/14697688.2013.816767.

25. Optuna: A Next-generation Hyperparameter Optimization Framework / T. Akiba [et al.]. — 2019. — arXiv: 1907.10902 [cs.LG].

26. *Pearson N. D.* Risk budgeting: portfolio problem solving with value-at-risk. — John Wiley & Sons, 2011.

27. Risk and portfolio analysis: Principles and methods / H. Hult [et al.]. — Springer, 2012.

28. Scikit-learn: Machine Learning in Python / F. Pedregosa [et al.] // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.

29. Tail-GAN: Learning to Simulate Tail Risk Scenarios / R. Cont [et al.]. — 2023. — arXiv: 2203.01664.

30. *Tobjork D.* Value at Risk Estimation with Generative Adversarial Networks / Tobjork David. — LUND UNIVERSITY, 2021.

31. *Zhang Y.*, *Nadarajah S.* A review of backtesting for value at risk // Communications in Statistics-Theory and methods. — 2018. — Vol. 47, no. 15. — P. 3616–3639.

Appendix

Repository with the code, tuned hyper parameters, experiments and visualisations: https://github.com/BELONOVSKII/var_es_dgm