I SIREPRODIS

I SIMPÓSIO REGIONAL DE ENGENHARIA DE PRODUÇÃO E SISTEMAS

Otimização de Processos e Gestão de Cadeias Produtivas

22 e 23 de agosto de 2025

GENERATIVE ADVERSARIAL NETWORKS FOR GASOLINE CRACK SPREAD RISK ANALYSIS

Marcus Vinicius de Oliveira[1]

[1]AlfaZeta Analytics and Consulting, contato@alfazeta.com.br

DOI:10.5281/zenodo.16990044

Abstract

Managing risks associated with commodities is crucial to ensure that business operations lead to favorable financial results and reduce the risk of short-term financeability problems. To achieve this, it is necessary to create scenarios for commodity prices that accurately reflect their probability distributions. This paper presents an implementation of the well-established TimeGAN architecture for generating multiple scenarios of gasoline crack spread, with the objective of supporting risk management and business decisions. This approach offers a complementary approach to traditional stochastic models based on Stochastic Differential Equations for time series simulation and risk analysis. It leverages the powerful capabilities of Generative Adversarial Networks (GANs) to produce realistic scenarios, particularly in capturing complex probabilistic distributions without needing any assumptions about the data distribution. By accurately modeling the probabilistic distribution of critical risk factors, the GAN framework enables more reliable estimation of their potential impact on business performance, making it a robust tool for financial risk assessment.

Keywords: Generative Adversarial Networks, Time Series Generation, Risk Analysis, Gasoline prices.

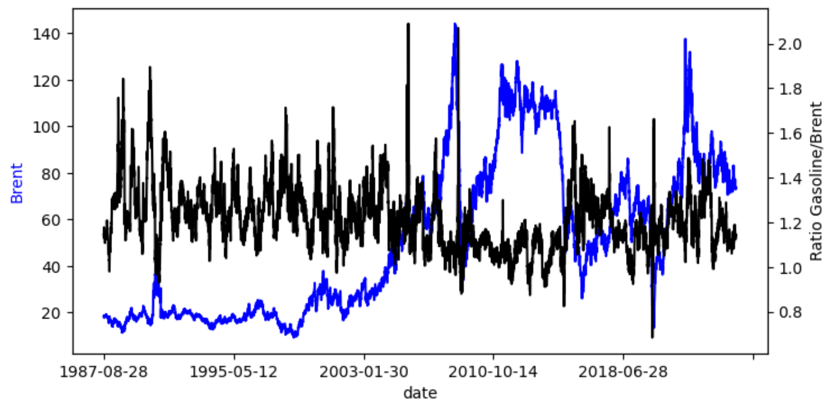
1. INTRODUCTION

Commodities are fundamental to the functioning of the global production chain, as they are essential inputs for various sectors of the economy. In addition to food and energy, there are commodities that serve as basic products for the manufacture of goods, directly influencing the efficiency and competitiveness of companies. Among these commodities, oil and refined products can be even more important because they are necessary to enable the production process, including that of other commodities with significant impact in all sectors of the economy.

Beyond challenges involved in operations, the business impact of commodities is also reflected in production costs. Since commodity prices are set on the international market, their volatility, as illustrated in Figure 1, impacts companies' results and can be significant, especially in the short term, when conditions for adaptation are very limited.

The impact of high commodity price volatility can be mitigated through hedging strategies that utilize financial instruments like futures contracts. These strategies can be employed by both commodity producers and consumers to lock in future purchase and sale prices, thereby enabling better margin management. However, the effectiveness of hedging strategies, as well as the quantification of companies' risk exposure, relies on estimates of future price behavior. To achieve this, models must be developed to represent the expected future price movements of these products. These models can leverage relevant characteristics of each series to make forecasts, such as stationarity and mean-reversion properties, which are typically observed in commodities.

Figure 1. Time series of Brent crude oil price and the ratio between gasoline USG prices and brent price



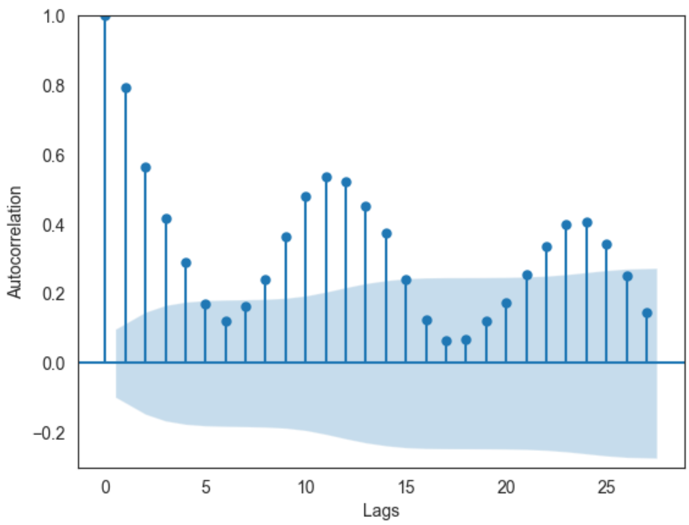
In addition to volatility, another characteristic observed in Figure 1 is that the ratio between gasoline and Brent prices (black time series) appears to be stationary, which is very positive for both stochastic modeling and the application of machine learning models. To confirm the stationarity of the series, the Augmented Dickey-Fuller Test (ADF) was used, and the results are presented in Table 1. Since the null hypothesis of the test is the presence of a unit root and the p-value obtained is very low, 1.07E-09, it can be concluded that the series is stationary. Therefore, the benchmark model that will be used to evaluate the performance of the GAN-based model is a simple autoregressive model.

Table 1. Augmented Dickey Fuller(ADF) test results

ADF Results		**Critical Values**		
ADF Statistic	-6.9328	**1%**	**5%**	**10%**
p-value	1.07E-09	-3.4311	-2.8619	-2.5669

Figure 2 presents the autocorrelation of the gasoline/brent ratio series, demonstrating significant seasonal behavior. This behavior is recognized as relevant for gasoline; however, since our interest in this work is to address prediction horizons of up to one year and there is no observable seasonality within this interval, the model will not be explicitly conditioned by time. Since the GAN model will be trained using rolling windows extracted from the series, it would be possible to include time as an additional series in order to provide the model with explicit time information so that it can learn seasonal behavior. If a methodology such as the one presented in this work is used for time horizons longer than one year or for series with seasonality shorter than the prediction period, it is recommended to treat seasonality explicitly.

Figure 2. Autocorrelation of time gasoline/brent ratio showing seasonality with period of 12 months.



Rising raw material prices, for example, can lead to increased operating costs, reducing margins and compromising the financeability of operations. In this scenario, risk analysis plays a key role, providing a probabilistic view of the potential impacts of commodity prices on companies' cash flow. To achieve this, a probabilistic analysis of commodity prices must be developed, enabling the risk manager to understand the likelihood of adverse scenarios occurring and what efforts should be made to minimize the impacts of these scenarios.

Financial risk management depends on assumptions about the behavior of risk factors relevant to business performance. A company with revenue denominated in foreign currency, for example, has its results strongly impacted by the exchange rate. The exchange rate, in turn, has a non-zero correlation with interest rates and interest rate differentials between countries. As both interest rates and exchange rates, along with various other market variables, fluctuate over time, these factors lead to uncertainties about business performance. Although we decided to implement the methodology for gasoline crack spread modeling, it would be possible to test its performance for many other assets, including exchange rates and interest rates.

The sophistication of financial instruments and the wide range of structured operations, along with the significant number of players using high-frequency strategies, necessitates a rapid understanding of the market context by models. These models must quickly adapt their probability distributions to accurately quantify the involved risks and adjust strategies, if necessary. Additionally, for a quantitative risk assessment, it is necessary to generate time series for uncertainty variables in such a way that both their probability distributions and correlations between variables adequately represent the probability distributions at each moment. Since the institutions are exposed to various risks, it is crucial to have well-represented joint distributions. This justifies the pursuit of methodologies that effectively capture the probability distributions at a given moment and generate meaningful scenarios.

The Oil & Gas sector is characterized by numerous risks, particularly those linked to product prices, with crude oil prices being the most significant. The pricing of refined products such as gasoline, heating oil and diesel, which is often influenced by crude oil prices, also plays a crucial role. A key variable in the financial modeling of assets in this sector is commonly referred to as the crack spread. This is defined as the difference between the price of a product and the price of a benchmark crude oil, such as Brent or WTI.

In this work, we are interested in simulating the crack spread of gasoline with respect to Brent used as crude oil benchmark, that is defined as the difference between Gasoline and Brent prices over time. While it is common to refer to the crack spread instead of the ratio, in practice, both the spread and the ratio are equivalent. With the simulations of Brent and the ratio, it is possible to derive the simulation of gasoline and subsequently calculate the crack spread. The choice to model the gasoline/brent ratio or the difference between the gasoline and brent depends on the performance achieved by the model. In this study, we are using U.S. Gulf Coast Conventional Gasoline as the reference for the price of gasoline.

The generation of financial time series can be accomplished using simulations of stochastic processes. Various models can be employed, depending on the properties of the series and the available data. Some stochastic processes are based solely on the behavior of the series in question, such as the Geometric Brownian Motion (GBM), while others may be based on the autoregressive behavior of the series, such as ARIMA models, or incorporate the effect of exogenous series, like ARIMAX. There are also more sophisticated models, such s the Schwartz-Smith model widely used for commodity modeling, which leverages additional information provided by the futures contract structure, as presented in (AIUBE et al, 2008).

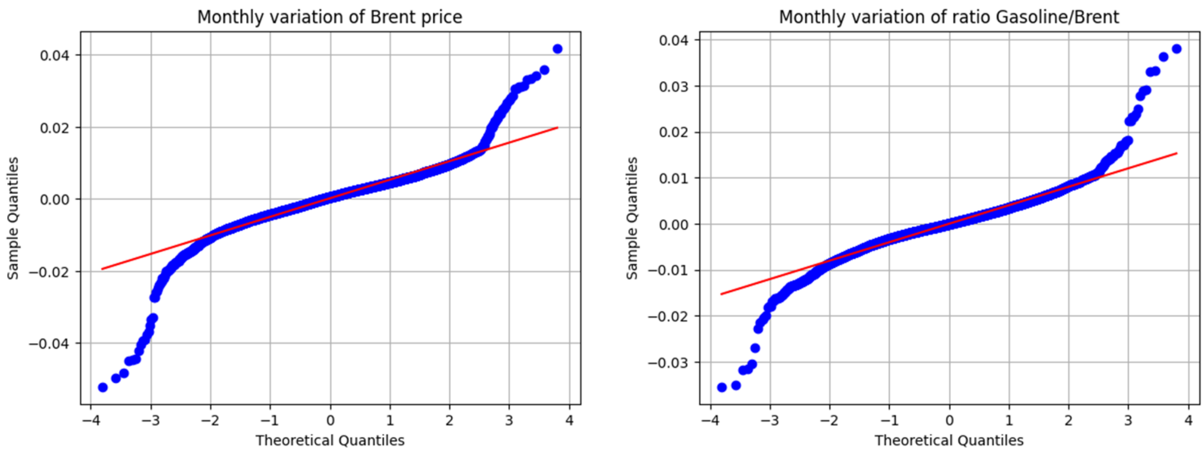
Oil price simulation can be performed using stochastic process modeling, as presented in (SCHWARTZ and SMITH, 2000), in which the spot price is modeled by two factors, one short-term and one long-term. The short-term factor (χ_t) has mean-reversion dynamics with coefficient κ (mean reversion speed) and volatility σ_χ, and the long-term factor (ξ_t) has random walk dynamics with drift μ_ξ and volatility σ_ξ. The two processes are correlated through the increments dz_ξ and dz_χ, such that $dz_\xi . dz_\chi = \rho_{\chi,\xi}$ (correlation coefficient $\rho_{\chi,\xi}$). Equation (1) presents the mathematical formulation of the model known as Schwartz-Smith 2 factors model. Other versions of the model has been proposed in literature, including more factors or modifying the dynamics of the factors such as allowing the mean reversion of long-term factor.

$$\begin{cases} St = \exp(\xi_t + \chi_t) \\ d\xi_t = \mu_\xi dt + \sigma_\xi dz_\xi \\ d\chi_t = -\kappa \chi_t dt + \sigma_\chi dz_\chi \end{cases} \tag{1}$$

However, stochastic process modeling, while quite robust and accepted in the literature, has the disadvantage of requiring assumptions about the data-generating process, which is not necessary when using a generative model, such as Generative Adversarial Networks (GAN), originally introduced in (GOODFELLOW et al., 2014). This is particularly important in financial time series, which typically exhibit heavy-tailed behavior, as illustrated in the Q-Q plots presented in Figure 3, in which both the monthly price variation of Brent Crude Oil price and the Gasoline/Brent ratio exhibit significant deviation from normal distribution. Therefore, the objective of this work is to train a GAN model capable of generating probabilistic scenarios for oil prices and gasoline crack spreads with a one-year horizon, contributing to better planning by companies and investors.

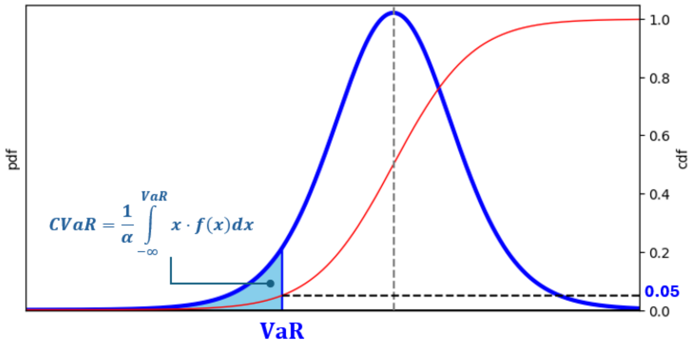
It is important to emphasize that the methodology used in this work, which construct a GAN-based model, does not invalidate the use of stochastic models to analyze commodity price behavior. We believe that the methodology presented here is complementary, especially because it helps to fill a gap in traditional stochastic models, as it does not make assumptions about probability distributions, allowing for a more robust characterization of the assets price. On the other hand, some stochastic models can use some related assets prices to estimate their parameters, such as the methodology proposed in (SCHWARTZ and SMITH, 2000) that uses prices of futures contracts as observers of the short and long term factors, enabling to incorporate more context on the forecast, in this particular case the context provided by the market consensus expressed in futures contracts prices.

Figure 3. QQ Plot of monthly variation for Brent crude oil price and gasoline/brent ratio.



Risk metrics such as Value at Risk (VaR) and Cash Flow at Risk (CFaR), when calculated using a model that ignores the existence of heavy tails, tend to underestimate risk. Therefore, methodologies that contribute to modeling asset prices without assuming a specific probability distribution have advantages, as they better represent the intensity of heavy tails. VaR is a metric widely used in Financial Risk Analysis due to its simplicity, as it quantifies the maximum expected loss at a given confidence level by analyzing the probabilistic distribution of outcomes. This metric depends on the distribution assumed in the simulation of the variables that impact the variable in question. In this context, the use of GANs to generate scenarios for certain financial series, such as commodity prices, can be particularly relevant. In Figure 4, VaR is defined as the percentile; for example, if you need a confidence level of 95%, the VaR will be equal to the P5 of the probability distribution of outcomes. Sometimes, VaR is defined in terms of the difference from the expected value of the probability distribution, such that VaR is the difference between P5 and the mean value of the probability distribution. Regardless of the probability distribution, VaR is a straightforward consequence of the probabilistic simulation of the variables influencing the target outcome variable, such as Cash Flow or Net Income.

Figure 4. Schematic representation of Value at Risk (VaR) for a given distribution.



It's also important to note that VaR models aren't necessarily sufficient to quantify risk, especially since they may be little influenced by the tails of the distribution. To this end, the use of methodologies capable of generating extreme scenarios that may not have been realized to date may be relevant. Therefore, generative models like the one presented in this work can be very useful because, although trained to mimic the past behavior of the assets, can eventually generate more extreme scenarios, that can be used to improve the risk management and to perform stress tests.

2. RELATED WORKS

One of the key tasks in quantitative analysis is estimating the optimal joint probability density for the variables of interest. This can be approached through various methods, including Vector Autoregression and the creation of generative models utilizing Generative Adversarial Networks (GANs). While GANs are well-known for their outstanding performance in image generation, they also prove to be highly effective for time series forecasting. In the work of (RIZZATO et al, 2023), the authors illustrate the use of Generative Adversarial Networks (GANs) for producing synthetic financial scenarios. The paper investigates the application of a GAN as an economic scenario generator (ESG), leveraging a Markovian framework along with multi-layer neural networks to facilitate variable transitions that respect the historical correlation structure among them. This approach permits the creation of multivariate trajectories and allows for conditioning based on specific scenarios involving one or more state variables. Moreover, the algorithm autonomously derives dependencies from empirical data, eliminating the need for regular calibration.

A limitation of the GAN-based model is its dependence on historical events, which constrains its capability to generate new dependencies. However, it may respond more swiftly to shifts in risk factors when compared to traditional models. In summary, the GAN-based method is practical and could act as a valuable alternative or benchmark to standard internal models.

In the study by (ZHANG et al., 2019), the authors employ an LSTM-based Generative Adversarial Network (GAN) to model a historical period for financial indices and stock prices. The architecture proposed consists of an LSTM generator, while the discriminator is structured as a Multi-Layer Perceptron (MLP) with three hidden layers. This model learns the underlying data structure from historical information and can generate new scenarios that align with that structure.

(YOON at al., 2019) introduced TimeGAN, a novel framework for generating time-series data that merges unsupervised GAN with the control over temporal dynamics found in supervised autoregressive models. TimeGAN employs a jointly trained embedding network and supervised loss to produce realistic time-series data that surpasses some of the benchmarks outlined in the paper. The TimeGAN proposal represented a significant advance in the generation of synthetic time series using GANs, as it allows for the generation of realizations that adequately respect the temporal structure of the series. This had been a major challenge in applying GANs to time series generation until that point.

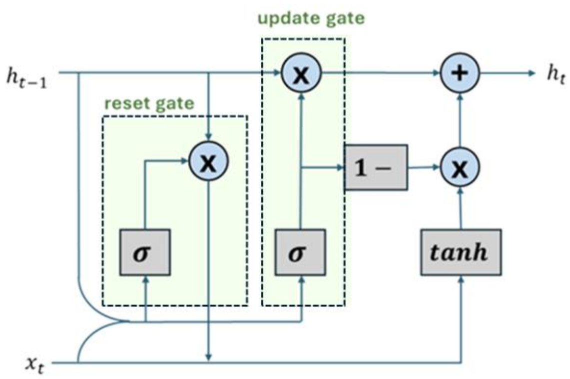
In the work of (SONKIYA at al., 2021), the authors suggest a method for forecasting stock prices. They utilize a version of BERT, a pre-trained transformer model for natural language processing developed by Google, to conduct sentiment analysis on news and headlines related to Apple Inc., a NASDAQ-listed company. Subsequently, a Generative Adversarial Network (GAN) is used to forecast the stock price of Apple Inc., integrating technical indicators, historical prices, stock indexes from various countries, and certain commodities, along with sentiment scores. The effectiveness of the proposed method is evaluated against baseline models, including Long Short Term Memory (LSTM), Gated Recurrent Units (GRU), vanilla GAN, and Auto-Regressive Integrated Moving Average (ARIMA) models. In this instance, the paper does not concentrate on synthetic time series generation but rather on time series forecasting, which represents another application of GANs. Nevertheless, it is important to note that one of the benefits of utilizing GANs for asset price predictions is their ability to generate multiple realizations when supplied with random seeds.

In the context of financial markets, big data typically refers to a large number of features rather than a high number of observations. High-frequency data is a notable exception, as it can produce millions of observations per asset. However, when formulating strategies with a longer time horizon, the number of realizations available for training the models is significantly diminished. Unlike image-based applications, conventional data augmentation techniques such as flipping, cropping, and rotating do not apply to financial time series. However, some research, such as (SILVA and SHI, 2019), proposes methods for augmenting time series by generating synthetic data through GANs.

3. FUNDAMENTALS

Recurrent Neural Networks emerged to enable neural network models to learn the temporal structure of data. Since the goal is to generate time series using GANs, it makes sense for the modules that comprise the network to be built using recurrent networks. Among the most popular recurrent network architectures are Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU). The architecture presented in Figure 5 illustrates the recurrent neural network adopted for this work, specifically the GRU, introduced by Cho et al. (2014).

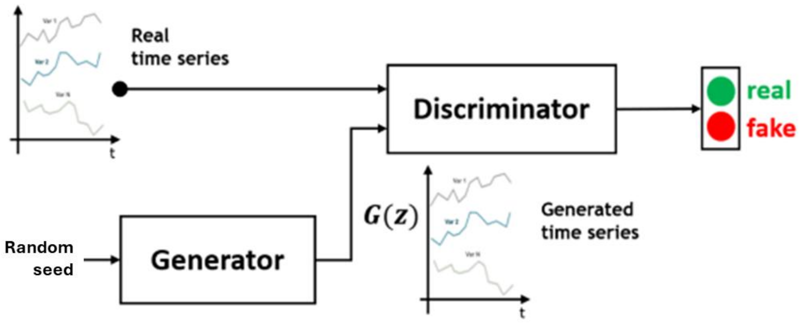
Figure 5. Schematic representation of Gate Recurrent Unit (GRU) model.



GRU reduces the complexity of its predecessor architecture, LSTM, by comprising only two gates. The Update Gate regulates the amount of information that should be retained from the previous state, helping to control how much of the past is relevant to the present and allowing the network to maintain the persistence of significant information over time. The Reset Gate, in turn, controls the amount of information from the previous state that should be forgotten, helping the network to focus more on new inputs. This gate is crucial for handling input sequences that may contain irrelevant information from the past. The weights and biases of each gate are learned during training. We will not detail the equations of the architecture, as it is currently well known and widely used.

Figure 6 schematically presents the original GAN, introduced in (GOODFELLOW et al., 2014), in which a neural network, the Generator, generates synthetic realizations from a random seed without any additional input. Another neural network, the Discriminator, is trained to determine whether the generated series is real or not (generated by Generator), receiving, in addition to the generated series, the true realizations. The two networks are trained in an adversarial schema so that, at the end of the training process, the Generator can generate synthetic realizations very close to the true data distribution and Discriminator is well trained to recognize fake samples. One challenge for GAN's application to generate time series is that it should learn to generate synthetic series respecting not only the correlations between the desired series (in this case brent price and gasoline crack spread) but also the autocorrelation of the series.

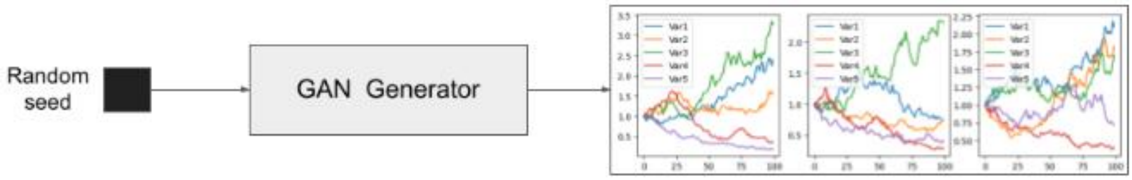
Figure 6. Conceptual representation of Generative Adversarial Network architecture.



Considering the excellent performance of GANs for generating realistic images, using mainly convolutional neural networks, capable of efficiently processing inputs with spatial structure, it makes sense to investigate the feasibility of using GANs to generate time series, as presented in (SILVA and SHI 2019) and (GEISSLER et al, 2022).

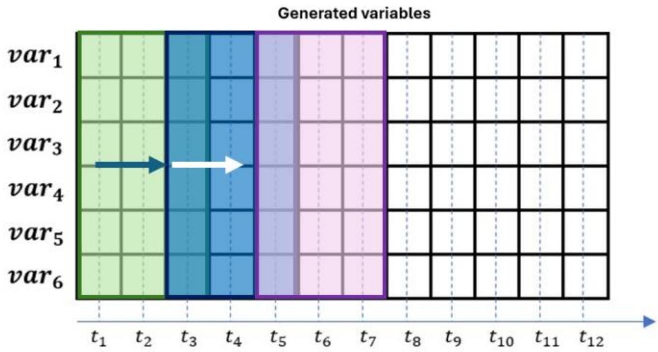
In the particular application of this work, we are interested in creating joint probability distribution of gasoline crack spread and Brent crude oil price, without the need to make any assumptions about the behavior of these variables, which would be necessary in the case of a stochastic process. Figure 7 shows what is expected of the trained model, which should receive a random seed and generate scenarios for some time series, maintaining consistency with their temporal structure and the correlation between them. The final product is, therefore, a time series generator that can be utilized for various applications, with a particular emphasis on risk analysis and risk management in investment portfolios.

Figure 7. GAN application to generate multiple scenarios of time series.



Typically, GAN architectures for image processing are built using convolutional networks that involve, among other operations, the convolution of filters of different sizes with the images to perform spatial processing and capture their properties. A key aspect is that, to represent correlations between variables, the filter size must cover all variables within the same time interval, as shown in Figure 8. This is because there can be no difference in the positioning of the series in the matrix, as the matrix construction is merely a representation, and there is no spatial structure in the data, unlike what occurs with images.

Figure 8. Example of a generated matrix consisting of 6 variables and 12 timesteps and filters used to process the matrix in case of implementation using convolutional neural networks.

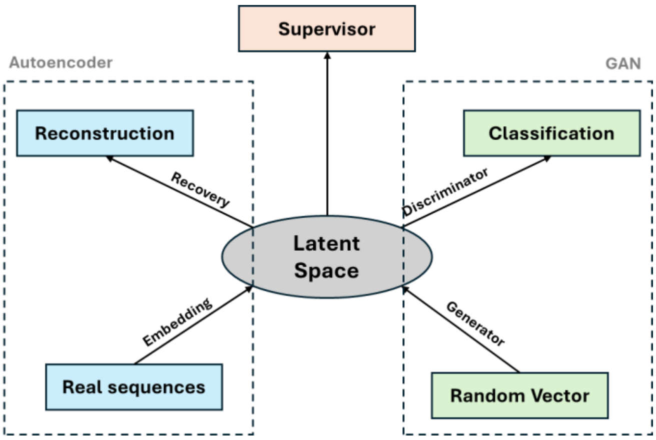


Furthermore, even with the adopted filters covering all variables simultaneously, there remains the issue that each filter only processes a defined number of time intervals, which limits its ability to capture temporal structures. This challenge could be addressed by using multiple filters, allowing the network to learn correlations between values from different time intervals. Therefore, considering the experiments conducted by the authors, utilizing recurrent neural networks for the generator and discriminator is a natural choice, as these networks are designed to process sequences. Results related to implementations using convolutional networks will not be presented, as they performed significantly worse than those using recurrent networks.

One alternative is to utilize Conditional GAN (CGAN) architecture, as introduced in (MIRZA and OSINDERO, 2014). However, although the performance of this architecture in generating realistic images is usually good, the results achieved with the CGAN implementation were not satisfactory for time series generation. Therefore, we implemented the architecture introduced in (YOON et al, 2019) which shows a significant improvement in performance of time series generation. Figure 9 presents a schematic representation of the architecture introduced in (YOON et al, 2019), which is composed of 3 parts, a GAN, an autoencoder and a supervisor.

The GAN used consists of the original structure, with a generator and a discriminator. The generator creates synthetic time series samples, while the discriminator evaluates whether the samples are real or fake (generated by the generator). The autoencoder is responsible for learning a latent representation of the time series and is trained to reconstruct the input from this representation. This is a fundamental modification to the original GAN to preserve the temporal structure, which was the main challenge encountered when attempting to use the original GAN structure for time series generation. The TimeGAN supervisor acts as an orchestrator, guiding the learning process, helping to ensure that the generated samples not only appear realistic but also maintain temporal coherence and feature dependencies observed in the training data.

Figure 9. TimeGAN architecture consisting of 3 modules (Autoencoder, GAN and Supervisor), following the proposal presented in (YOON et al, 2019).



One way to assess whether the realized data are consistent with the synthetic data distribution produced by the GAN is to retrospectively evaluate a distance metric between a point and a distribution. In (VENTURINI and ALEJANDRO, 2015), the authors discuss metrics in probabilistic analysis. For this work, we decided to use the Mahalanobis distance (MAHALANOBIS, 1936) because it is a simple and effective metric for this application. The Mahalanobis distance between a point $\boldsymbol{x} \in \mathbb{R}^n$ and a distribution characterized by a mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and a covariance matrix $\boldsymbol{\Sigma}$ is defined by Equation (2) The number of timesteps predicted is not specified. The Mahalanobis distance accounts not only for the point distribution but also for the autocorrelation within the series.

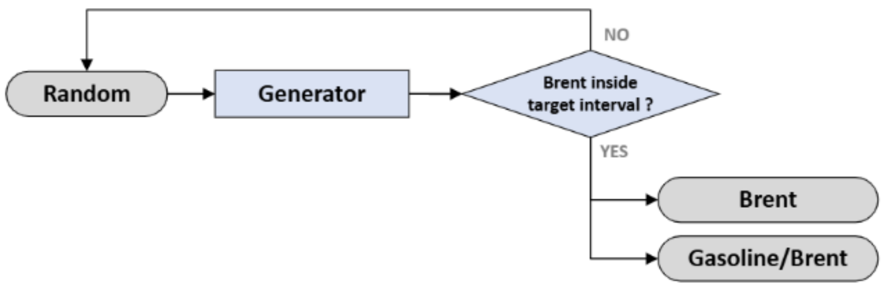
$$dist_M = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})} \tag{2}$$

Another way to assess whether the generated scenarios are consistent with the historical data is to analyze the correlation matrix between the series. Naturally, the GAN is not expected to generate scenarios with correlations between the series that are very different from those observed in historical data. However, the correlations are not expected to be the same, as they are not imposed but rather learned during training. Moreover, from a risk management perspective, it is beneficial for the GAN to generate scenarios that are significantly different from those observed, alerting risk managers to the possibility of changes in the behavior of the analyzed series, such as volatility beyond expectations or even correlation breakdowns.

4. METHODOLOGY

It is important to note that the framework presented in (YOON et al., 2019) is capable of generating time series that preserve the relationships between the series and the temporal structure, but it is not capable of using conditional data, which was the original idea. To this end, we will apply an a posteriori filtering flowchart to obtain the desired conditional probability distribution. Figure 10 presents a quite simple methodology for estimating and simulating the price of gasoline through the GAN trained to adjust the joint distribution of the Brent variables and the Gasoline/Brent ratio. Basically, the GAN is simulated multiple times using random inputs, and then a filter is applied to validate only the realizations where Brent falls within a defined interval $(brent_{min}, brent_{max})$. This way, simulations of the ratio $\frac{gasoline}{brent}$ are obtained that are conditioned to $price_{brent} \in (brent_{min}, brent_{max})$.

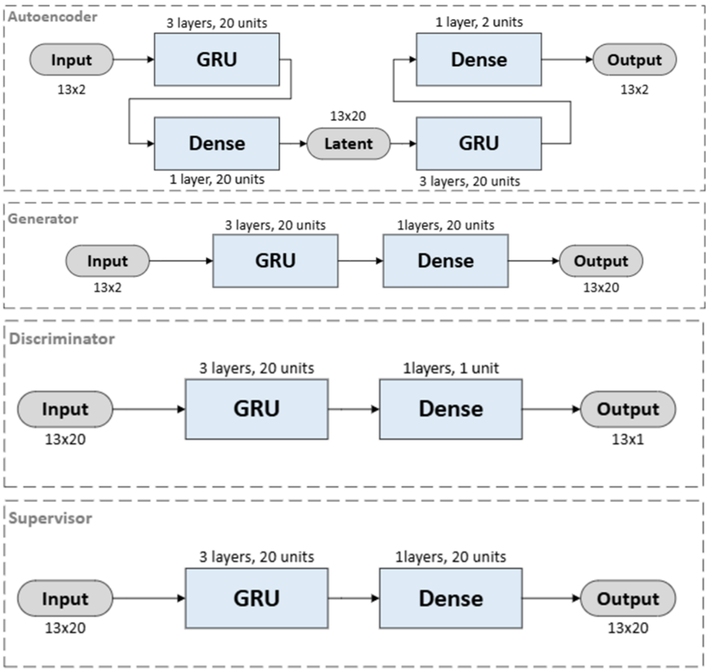
Figure 10. Schematic representation of Gasoline simulation conditioned by Brent price range.



Conditioning through a posteriori filtering may seem computationally expensive. However, it is important to remember that model evaluation after training is very inexpensive and is performed tensorly in TensorFlow. Thus, the computational cost of implementing the diagram shown in Figure 10 is not prohibitive and allows for easy adaptation of the conditioning data without issues. For example, one could condition on a time average of Brent or even on properties such as Brent volatility, thereby filtering only scenarios that meet this condition. Since the objective of this work is to analyze the relevance of using the crack-spread time series scenario generation technique with TimeGAN, we did not evaluate other methods of generating conditioning scenarios.

Figure 11 shows the architecture used. All modules were implemented using the GRU recurrent neural network architecture followed by dense layers with the specified number of units. The GRU model was chosen because its performance is similar to LSTM with fewer parameters and, consequently, less prone to overfitting. Thirteen timesteps were generated instead of 12 (since the forecasts are monthly) in case the user wishes to normalize to the first month. However, the number of forecast steps can be adjusted depending on the application. The architecture was implemented in Python using Tensorflow Keras. To prevent negative price values and to enhance training stability due to the limited outputs of each layer, all activation functions used were sigmoid. The optimizer employed was Adam with default parameters.

Figure 11. TimeGAN architecture implementation.



5. RESULTS

Figure 12 presents the results of the TimeGAN implementation without conditioning, demonstrating that the architecture is visually capable of generating realizations that capture the time series' temporal structure well. It is also possible to note in Figure 12 that the correlation between the crack-spread and Brent prices, considering the synthetic realizations and the realized data, is close.

Note, however, that some difference is expected and, to some extent, even desirable, as the model is free to generate realizations outside the realized intervals, which is positive from a risk analysis perspective, since it generates realizations that never occurred but could occur in the future.

Another relevant aspect is that Pearson's correlation only measures the linear relationship between variables, whereas GAN can also capture non-linear relationships that may exist between them. This is another reason why correlations are not necessarily close. However, when correlations are low, the correlations between synthetic generations are expected to be high, as if they were high.

Figure 12. Correlation between gasoline spread and brent prince in both test dataset and synthetic series.

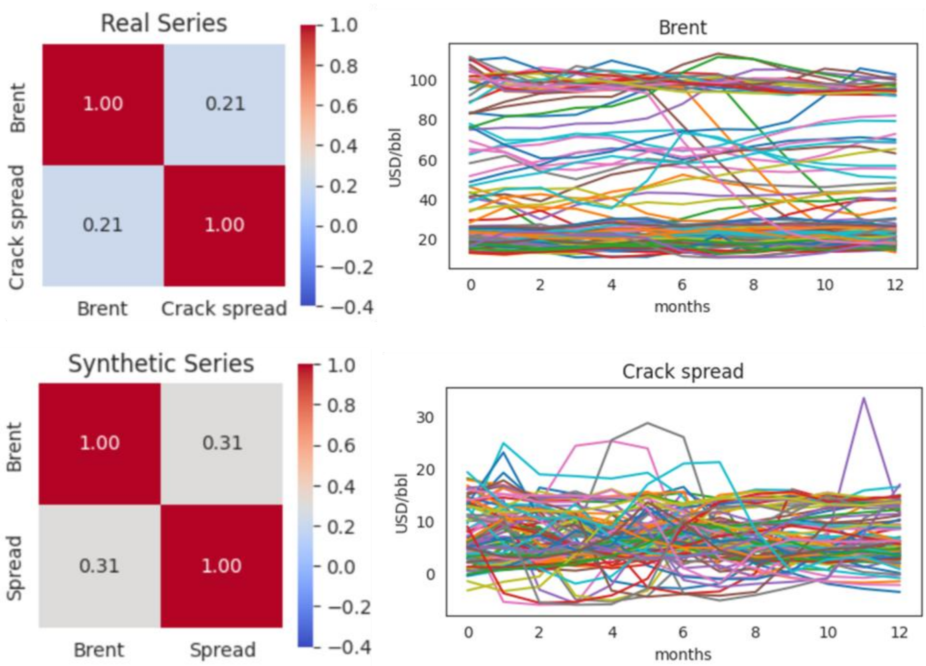


Figure 13 shows the Mahalanobis distance between the realizations of the synthetic distribution of the data generated by the GAN from 2021-01-01 onwards. For comparison purposes, we consider as a benchmark an autoregressive model for the gasoline/brent ratio, which, as can be seen in Figure 1, has a sufficiently robust stationarity aspect to assume mean reversion. Note that the Mahalanobis distance of the GAN-based model is systematically lower than the Mahalanobis distance for the first-order autoregressive model.

It is important to emphasize, however, that the selected benchmark is very simple, so the results obtained with the GAN cannot be considered better than those obtained with other, more sophisticated stochastic models. However, it is notable that the performance is systematically better than that of the autoregressive model, since, based on the authors' experience in modeling stochastic processes, models of this type tend to produce results that are either similar in performance to more sophisticated models or, when surpassed, the performance difference is usually not as significant. Figure 14 illustrates the results derived from the application of a GAN model, which has been adjusted using historical Brent data and the Gasoline/Brent ratio.

Figure 13. Mahalanobis Distance using GAN and Autoregressive model for crack spread.

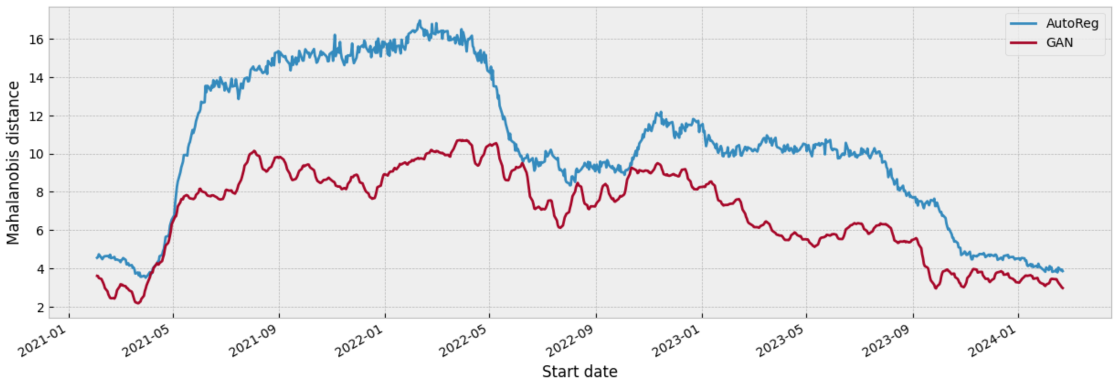
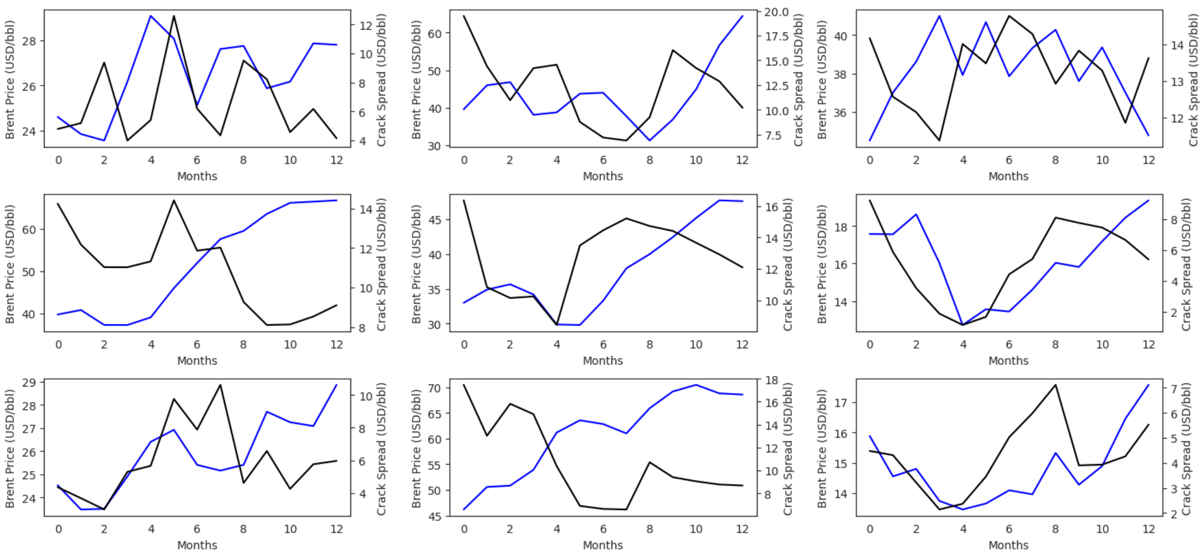


Figure 14. Example of 1-year synthetic Brent (blue) and gasoline crack spread (black) generated series.



This process was followed by the implementation of the flowchart depicted in Figure 10. The results were filtered to include only simulations with a Brent value within an interval per barrel, representing the demand to generate simulations of the Gasoline/Brent ratio, assuming that monthly average Brent price will be within this range during the analyzed horizon. Other limits may also be applied. Figure 15 shows the average Mahalanobis distance for the conditioned version of the crack spread, showing that GAN model outperforms the benchmark model in all conditioning ranges.

Figure 15. Malahanobis distance of GAN model filtered a posteriori and autoregressive model.

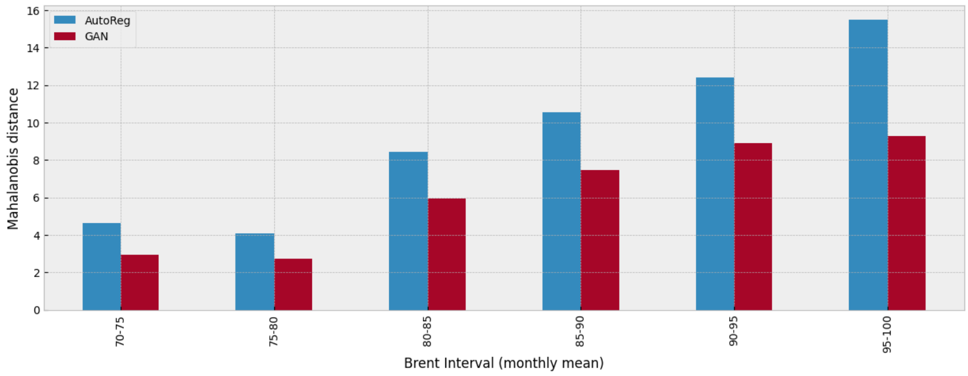
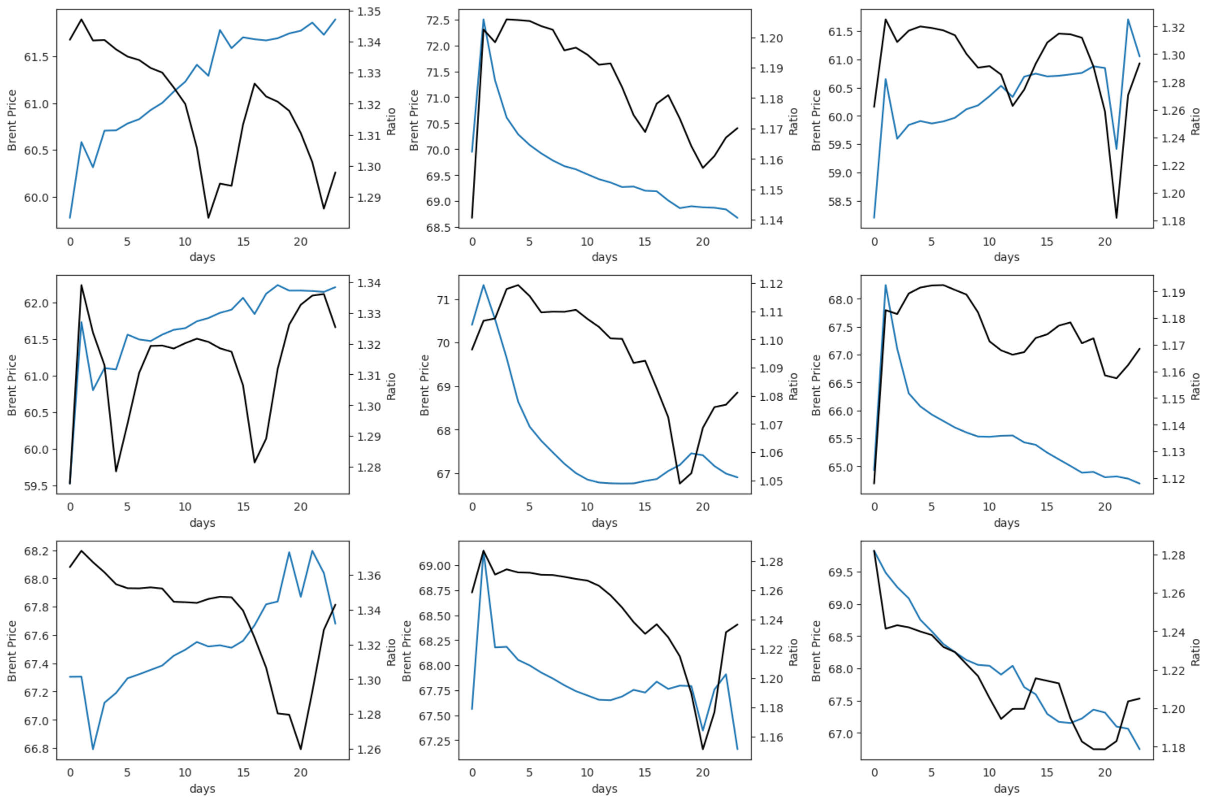


Figure 16. Example of a posteriori filtered simulations considering brent withing the interval 60-70 USD/bbl duting one month, brent (blue) and gasoline/brent ratio (black).



The objective of this work was to generate time series scenarios over a one-year horizon. However, the question remains whether modeling would be effective for a shorter horizon, specifically one month. Since the expected variation in Brent crude oil prices is smaller over a shorter horizon, it makes sense to condition the Brent price variation range rather than the average. Therefore, the prediction horizon was modified while maintaining all other model hyperparameters to qualitatively assess whether the generated scenarios are plausible. Figure

16 presents the results obtained, suggesting that the proposed methodology is also robust for shorter-term scenarios. It is important to emphasize, however, that these tests are preliminary and only indicate that the methodology can be applied to different prediction horizons.

The feasibility of using GANs to generate scenarios, whether conditioned on another variable or not, is well established in this work and in literature. However, since there are much simpler methods for generating series that maintain correlations, the question remains as to whether we truly need to use GANs instead of these models. One significant drawback of GANs is their lack of explainability and complexity. To preserve the temporal structure, it was necessary to divide the architecture into three modules, utilize a latent space, and implement adversarial training.

Conversely, a key feature of GANs is their lack of strict assumptions about the distribution of the data. Instead, they are designed to learn the underlying distribution of the training data. This lack of rigid assumptions about data distribution can be seen as an advantage compared to some traditional econometric models, which often rely on specific assumptions about the form of data distribution, such as normality and linearity. Due to its flexibility, GANs can capture complex, high-dimensional distributions that might be challenging for econometric models to handle effectively.

5. CONCLUSIONS

Considering the importance of generating multiple scenarios to support probabilistic analysis for companies and investments, especially in relation to the commodities market, the use of methodologies that do not require assumptions about the probability distribution of asset prices is relevant. This approach helps avoid underestimating the occurrence of more extreme scenarios, including those that support the construction of a VaR model. In this context, the use of GANs is warranted.

This paper presents the application of the GAN architecture known as TimeGAN to model gasoline crack spread prices. The initial proposal to use a conditional GAN to model the crack spread conditioned on the price of Brent crude oil was unsuccessful, as the architecture was unable to adequately capture the time structure of the series.

After implementing the TimeGAN architecture, following the proposal of (YOON et al, 2019), it was found that the network is capable of adequately representing the time structure of the series. Backtests conducted from 2021 to 2025 demonstrate that the proposed model outperforms an autoregressive benchmark. Despite its simplicity, this suggests that the GAN model has the potential to generate interesting probabilistic forecasts for the financial time series.

The correlation between Brent price and gasoline crack spread observed historically is well reflected in the scenarios generated by the GAN. It should be noted, however, that this correlation is not imposed, but rather learned by the network during training. It is also worth mentioning that the GAN was not expected to accurately reproduce the correlation between the series, as it is free to create scenarios that may be significantly different from those observed historically. This is very positive from a risk analysis perspective, as it allows for the evaluation of different scenarios and enhances the robustness of the risk analysis.

The proposed posteriori filtering methodology aims to generate the desired conditional distribution, although it is not generated directly from the network, but rather from a posteriori selection of scenarios of interest. Thus, it is understood that both the TimeGAN architecture implemented based on the proposal by (YOON et al, 2019) and a posteriori filtering have potential practical applications in financial risk management for companies subject to risks related to commodity prices, particularly gasoline. Considering the similar dynamic characteristics of other crack spreads, such as those of heating oil and diesel, the same methodology can be tested to generate probabilistic scenarios for these time series.

A future continuation of this work involves incorporating the conditioning variable into TimeGAN training, allowing the model to directly generate gasoline crack-spread data without the need for posteriori filtering. This enhancement would enable the model to learn conditional probability distribution. While this represents a significant step in this direction, the expected practical results will need to be compared with those presented in this work, which proved satisfactory when compared to a simple benchmark.

REFERÊNCIAS

AIUBE, F. A. L. **Modelos Quantitativos em Finanças: Com Enfoque em Commodities**. Bookman, 2012.

AIUBE, F. A. L., BAIDYA, T. K. N. and TITO, E. A. H. **Analysis of commodity prices with the particle filter**. Energy Economics 30(2): 597–605.

CHO, K. et al. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1724-1734). Association for Computational Linguistics, 2014.

GEISSLER, C., MORIZET, N., RIZZATO, M. and WALLART, J. **Generative adversarial networks applied to synthetic financial scenarios generation**. Physica A: Statistical Mechanics and its Applications. 2023

GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. and BENGIO, Y. **Generative adversarial networks**, Proceedings of the International Conference on Neural Information Processing Systems (NIPS) p. 2672–2680, 2014.

LU, J. and DING, D. **A hybrid approach on conditional gan for portfolio analysis**. arXiv, 2022. URL=https://arxiv.org/abs/2208.07159.

MAHALANOBIS, P. **On the generalized distance in statistics**, Proceedings of the National Institute of Sciences (Calcutta), 1936.

MARIANI, G., ZHU, Y., LI, J., SCHEIDEGGER, F., ISTRATE, R., BEKAS, C. and MALOSSI, A. C. I. **Pagan: Portfolio analysis with generative adversarial networks**, 2019.

MIRZA, M and OSINDERO;, S. **Conditional generative adversarial nets**. arXiv preprint arXiv:1411.1784 870, 2014. 414, 2014.

MOHAMMED, S. **Research on financial risk prevention and control methods based on big data**, 2019.

RIZZATO, M. et al. **Generative adversarial networks applied to synthetic financial scenarios generation**. Physica A: Statistical Mechanics and its Applications, v. 623, p. 128899. ISSN 0378-4371, 2023.

SCHWARTZ E. S. and SMITH, J. E. **Short-term variations and long-term dynamics in commodity prices.** Management Science, 46(7):893–911, 2000.

SILVA, B. D. and SHI, S. S. **Style transfer with time series: Generating synthetic financial data**, 2019.

SONKIYA, P., BAJPAI, V. and BANSAL, A. Stock price prediction using bert and gan, 2021.

VENTURINI, M. and ALEJANDRO, G. **Statistical distances and probability metrics for multivariate data, ensembles and probability distributions.** URL:https://api.semanticscholar.org/CorpusID:124313030, 2015.

YOON, J., JARRETT, D. and VAN DER SCHAAR, M. **Time-series generative adversarial networks**, 32, 2019.

ZHANG, K., ZHONG, G., DONG, J., WANG, S. and WANG, Y. **Stock market prediction based on generative adversarial network**. Procedia Computer Science 147: 400–406. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, 2019.