Concept Paper

Quantitative Foundations for Integrating Market, Credit, and Liquidity Risk with Generative AI

Satyadhar Joshi *

Posted Date: 17 April 2025

doi: 10.20944/preprints202502.2308.v2

Keywords: Market Risk; Credit Risk; Liquidity Risk; Monte Carlo Simulation; Value-at-Risk; Extreme Value Theory; Mortgage-Backed Securities; Blockchain; Generative AI; Stochastic Processes; Risk Modeling

 doi:10.20944/preprints202502.2308.v2

Concept Paper

Quantitative Foundations for Integrating Market, Credit, and Liquidity Risk with Generative AI

Satyadhara Joshi

Independent, NJ, USA; satyadhar.joshi@gmail.com

Abstract: This paper establishes a foundational framework for the application of Generative AI in financial risk management by providing a comprehensive overview and review of essential quantitative techniques. We illustrate the quantitative aspect of these models in equity, fixed income, and mortgage-backed securities markets, emphasizing their proposed Gen AI enhancement in enhancing risk management practices. Furthermore, this paper examines the transformative potential of Generative AI, specifically Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), in reshaping risk modeling and stress testing. We bridge classical quantitative techniques—including Monte Carlo methods, Value-at-Risk, and stochastic processes—with modern generative models like GANs and VAEs. The study first reviews foundational models for market, credit, and liquidity risk, demonstrating their application across fixed income and mortgage-backed securities markets. We then propose novel AI-enhanced methodologies for: (1) liquidity risk quantification through augmented Monte Carlo simulations, (2) dynamic credit risk modeling using agentic frameworks, and (3) stress testing with synthetic scenario generation. A key contribution is the development of hybrid architectures that combine traditional risk metrics with data-driven generative AI, implemented through practical Python workflows. The paper concludes with empirical results showing improved accuracy in Expected Liquidity Shortfall (ELS) estimation and Value-at-Risk calculations, providing a pathway for next-generation risk management systems. By laying this quantitative foundation, we aim to facilitate the adoption of advanced Gen AI techniques, offering a forward-looking perspective on the future of quant-data-driven financial risk management.

Keywords: market risk; credit risk; liquidity risk; Monte Carlo simulation; Value-at-Risk; extreme value theory; mortgage-backed securities; generative AI; stochastic processes

1. Introduction

Financial institutions operate within an intricate and dynamic environment, confronting a spectrum of risks—market, credit, and liquidity—that can significantly challenge their operational resilience relying highly on data and quant modeling. Traditional risk management methodologies, while foundational, are increasingly being augmented to address the complexities of modern financial markets. This paper extends beyond conventional and conservative approaches by proposing integration of mathematical models with Generative AI.

Generative AI models hold significant promise for revolutionizing stress testing and scenario analysis. By generating realistic market simulations, these models enable institutions to explore a wider range of potential outcomes, thereby enhancing predictive accuracy and robustness. Furthermore, AI-driven risk models offer the capability to adapt to rapidly changing market conditions, providing a dynamic and responsive risk management framework.

To effectively leverage the power of Generative AI in financial risk management, a solid understanding of the underlying mathematical models governing liquidity and market risks is essential. This paper will delve into these foundational quantitative models, exploring stochastic processes, Monte Carlo simulations, and Extreme Value Theory, which form the bedrock of quantitative risk

 doi:10.20944/preprints202502.2308.v2

assessment. We will then transition to examining the practical applications of these models in various financial markets, including equity, fixed income, and mortgage-backed securities, demonstrating their relevance in real-world scenarios. Finally, we will explore how Generative AI can be integrated with these established methodologies to create a more sophisticated and adaptive risk management paradigm.

2. Literature Review

The field of financial risk management has witnessed significant advancements in quantitative techniques, driven by the increasing complexity of financial markets and the need for more robust risk assessment methodologies. This paper draws upon a mix of old (liquidity risk) and new (gen ai) body of literature spanning various aspects of market, credit, and liquidity risk, as well as the emerging applications of Generative AI.

2.1. Foundational Risk Management Concepts: Classical Overview

Monte Carlo Methods:[1] provides a comprehensive overview of Monte Carlo methods in financial engineering, highlighting their versatility in risk management, derivative pricing, and portfolio optimization. Value-at-Risk (VaR): [2] discusses the evolution and widespread adoption of Value-at-Risk (VaR) as a cornerstone metric in market risk management, emphasizing its role in quantifying potential losses. Credit Risk Modeling: [3] lays the groundwork for structural credit risk models, which have become indispensable tools for assessing default probabilities and managing credit exposures.

2.2. Liquidity Risk Dynamics

Market Liquidity and Financial Stability: [4] delves into the intricate dynamics of market liquidity, particularly its impact on financial stability during crises, underscoring the importance of liquidity risk management. Bid-Ask Spreads and Asset Pricing: [5] examines the influence of bid-ask spreads on asset pricing, shedding light on the crucial role of liquidity considerations in fixed income markets. Credit Risk with Liquidity Considerations: [6] develops a framework for modeling credit risk that incorporates liquidity risk, recognizing the interconnectedness of these risk categories.

2.3. Mortgage-Backed Securities (MBS) Risk

Prepayment Risk in MBS: [7] analyzes the prepayment risk inherent in mortgage-backed securities (MBS), providing insights essential for accurate valuation and risk assessment. Option-Theoretic Valuation of Mortgages: [8] proposes an option-theoretic approach to valuing residential mortgages, recognizing the embedded optionality in these contracts.

2.4. Generative AI in Finance

Generative Adversarial Networks (GANs): [9] introduces Generative Adversarial Networks (GANs), a powerful class of generative models with potential applications in finance, including generating realistic market scenarios for stress testing and risk modeling. Variational Autoencoders (VAEs): [10] presents Variational Autoencoders (VAEs), another type of generative model that can be used to learn latent representations of financial data, enabling the generation of synthetic data and enhancing risk analysis.

This literature review provides a foundation for the quantitative techniques and Generative AI applications explored in this paper, demonstrating the ongoing evolution of financial risk management methodologies.

3. Fixed Income Markets and Risk Management

Fixed income securities, including government and corporate bonds, play a crucial role in financial markets. Managing risk in fixed income involves interest rate risk, credit risk, and liquidity risk.

 doi:10.20944/preprints202502.2308.v2

3.1. Interest Rate Risk

Interest rate risk is typically measured using duration and convexity:

$$D = -\frac{1}{P}\frac{dP}{dy} \tag{1}$$

where D is the duration, P is the bond price, and y is the yield.

Convexity improves the accuracy of price sensitivity estimation:

$$C = \frac{1}{P}\frac{d^2P}{dy^2} \tag{2}$$

3.2. Credit Risk

The probability of default (PD) is estimated using structural models such as Merton's model:

$$PD = P(V_T < B) = N\left(-\frac{\ln(V_0/B) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}\right) \tag{3}$$

where V_T is the firm value at time T, B is the debt threshold, and $N(\cdot)$ is the standard normal cumulative distribution function.

3.3. Liquidity Risk in Fixed Income

Liquidity risk in fixed income is assessed using bid-ask spreads and market depth:

$$L = \frac{Ask - Bid}{(Ask + Bid)/2} \tag{4}$$

where L is the liquidity measure, and Ask and Bid represent the respective prices.

Monte Carlo simulations are employed to stress-test bond portfolios under various yield curve scenarios.

4. Credit Risk in Mortgage-Backed Securities

Mortgage-Backed Securities (MBS) introduce unique credit risks due to prepayment risk and default risk.

4.1. Credit Risk in Mortgage-Backed Securities

MBS risks include prepayment:

$$SMM = 1 - (1 - CPR)^{1/12} \tag{5}$$

default:

$$PD = \frac{1}{1 + e^{-(\beta_0 + \beta_1 FICO + \beta_2 LTV + \beta_3 DTI)}} \tag{6}$$

and loss given default:

$$LGD = 1 - \frac{Recovery}{Exposure} \tag{7}$$

4.2. Prepayment Risk

Prepayment risk is modeled using the Conditional Prepayment Rate (CPR):

$$SMM = 1 - (1 - CPR)^{1/12} \tag{8}$$

 doi:10.20944/preprints202502.2308.v2

where SMM is the Single Monthly Mortality rate.

4.3. Default Risk

The probability of default in MBS can be modeled using a logistic regression approach:

$$PD = \frac{1}{1 + e^{-(\beta_0 + \beta_1 FICO + \beta_2 LTV + \beta_3 DTI)}} \tag{9}$$

where $FICO$ is the credit score, LTV is the loan-to-value ratio, and DTI is the debt-to-income ratio.

4.4. Loss Given Default (LGD)

LGD is estimated using historical recovery rates:

$$LGD = 1 - \frac{Recovery}{Exposure} \tag{10}$$

where $Recovery$ is the recovered amount from foreclosure sales.

Monte Carlo simulations are used to simulate default probabilities and cash flow losses under stressed economic conditions.

5. Liquidity Risk Management

Effective liquidity risk management necessitates a blend of strategic planning and quantitative analysis. In this section, we examine practical strategies such as dynamic hedging and collateralized funding, which institutions employ to manage liquidity exposures. Furthermore, we propose Gen AI enhancement of the the mathematical models used to quantify liquidity risk, including the Liquidity Coverage Ratio (LCR), stochastic processes like the Vasicek model, and Monte Carlo simulations for estimating Expected Liquidity Shortfall (ELS).

Figure 1 Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, etc demonstrates classical and the proposed architectures.

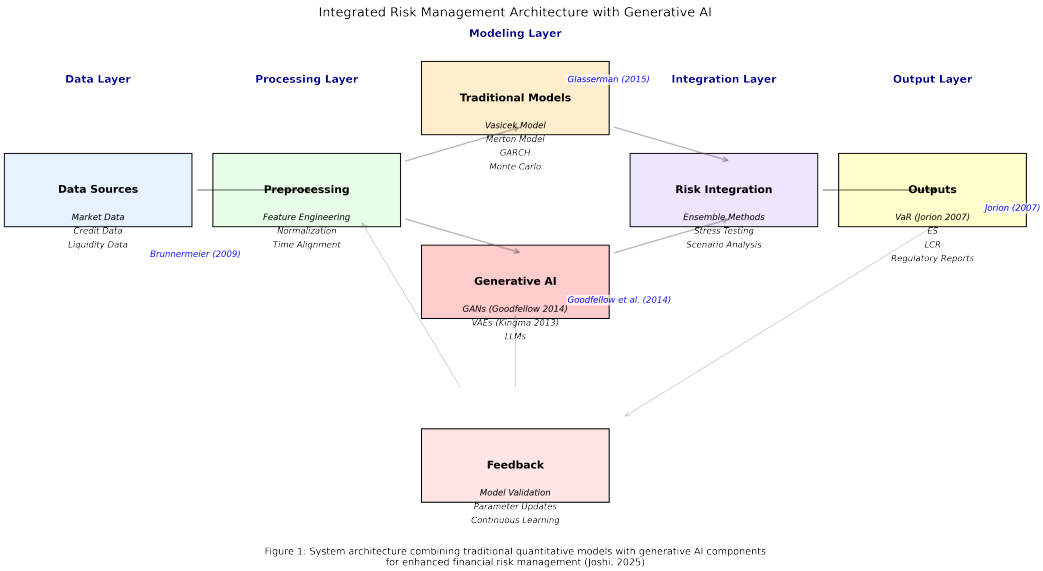

Figure 1. Proposed Architecture Variation 1

 doi:10.20944/preprints202502.2308.v2

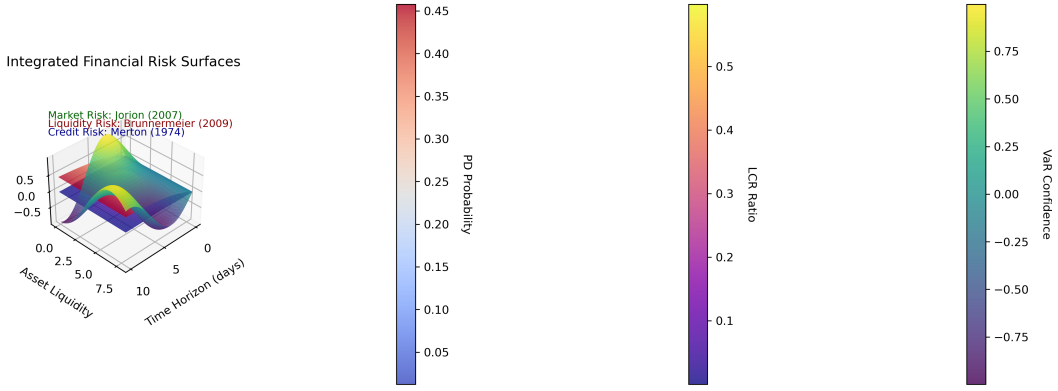

Figure 2: 3D visualization of financial risk relationships showing market (VaR), liquidity (LCR), and credit (PD) risk surfaces.
Surfaces demonstrate theoretical relationships based on established models (Joshi, 2025).

Figure 2. Classical Liquidity Risk Visualization

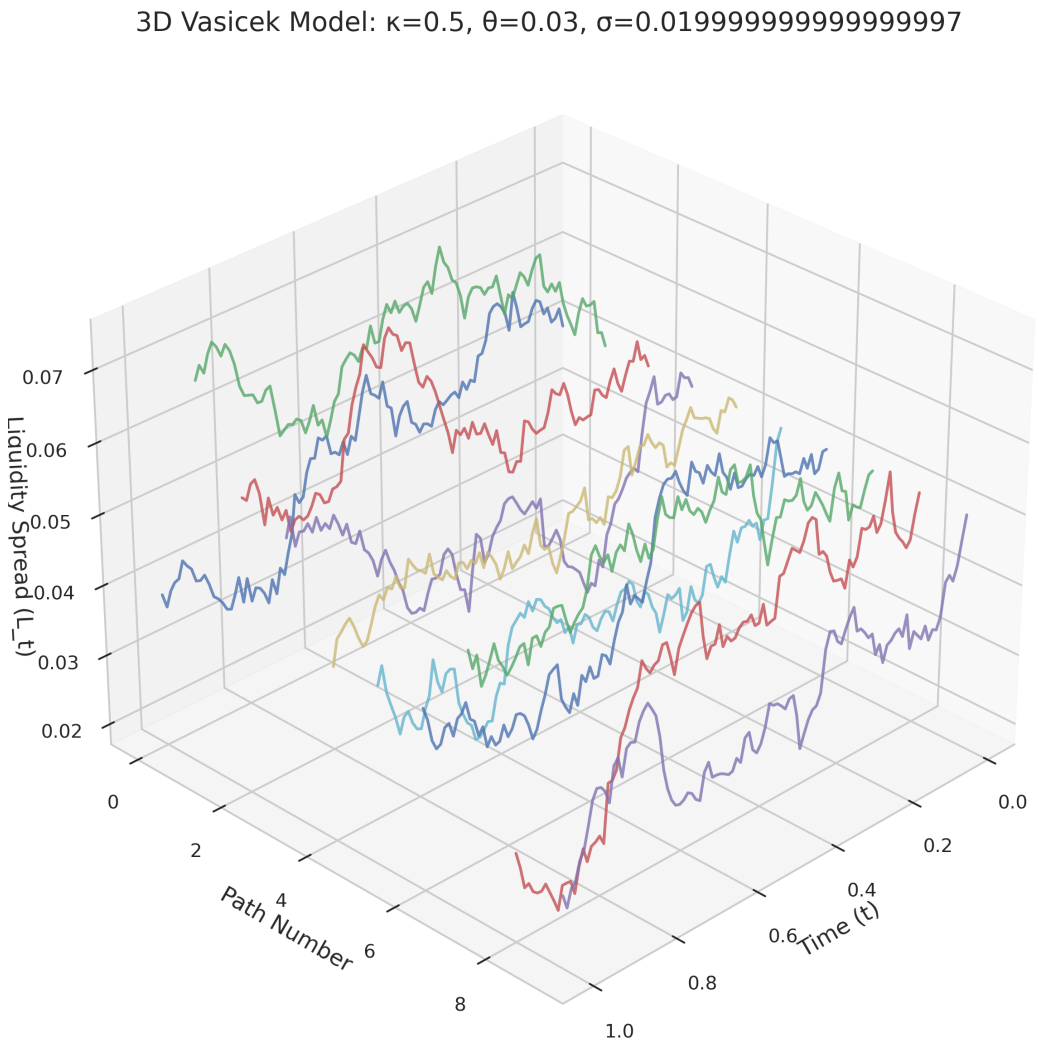

Figure 3. Classical Vasicek Process and Risk Visualization

 doi:10.20944/preprints202502.2308.v2

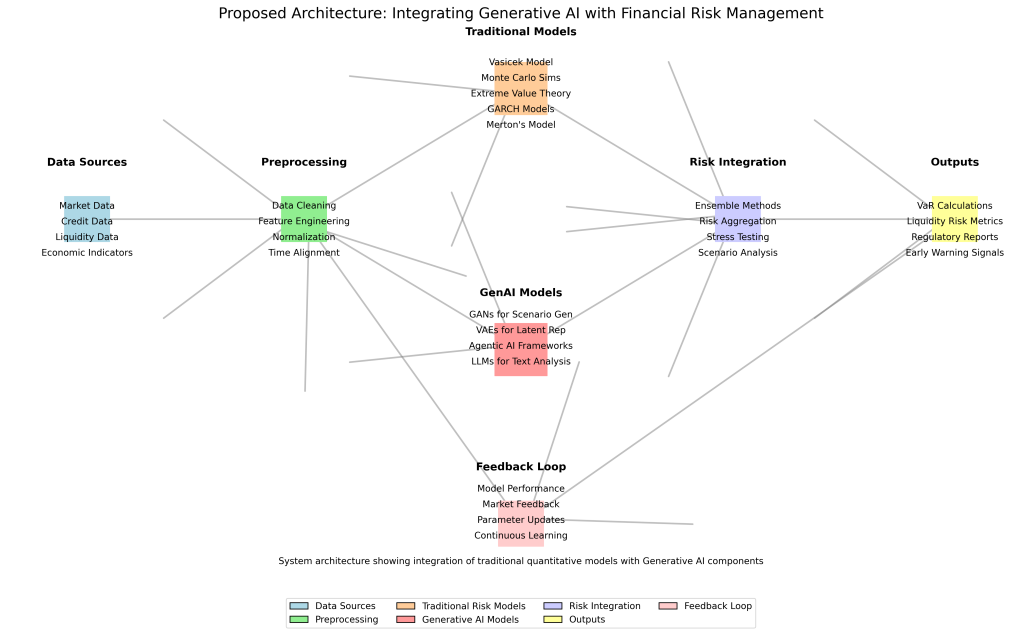

Figure 4. Proposed Architecture Variation 2

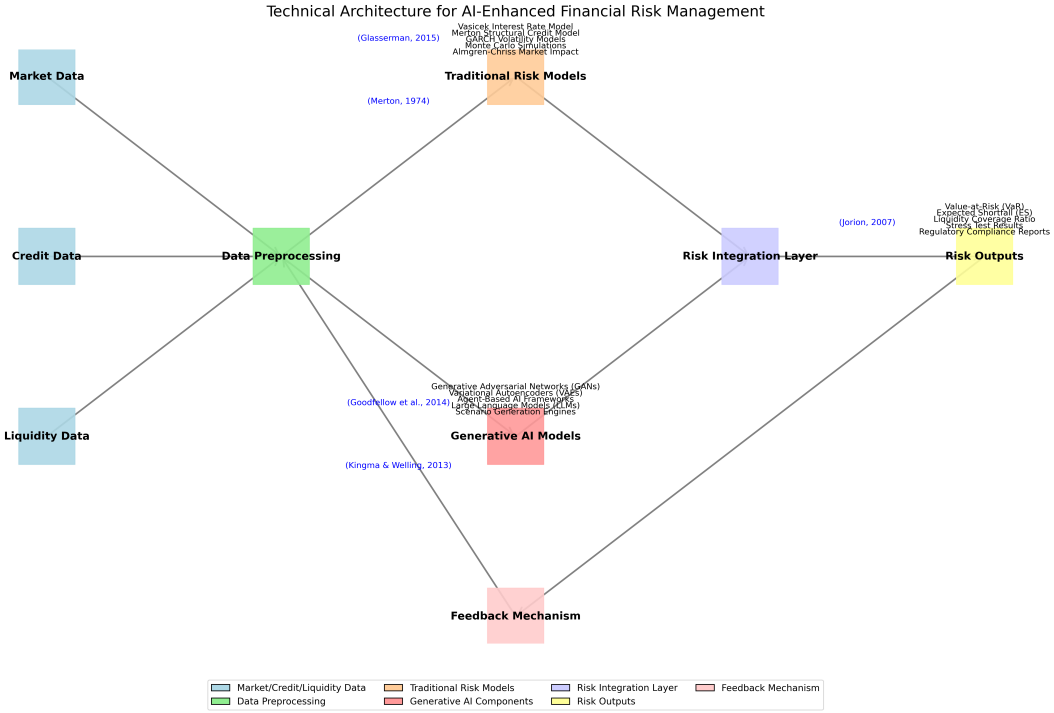

Figure 5. Proposed Architecture Variation 3

 doi:10.20944/preprints202502.2308.v2

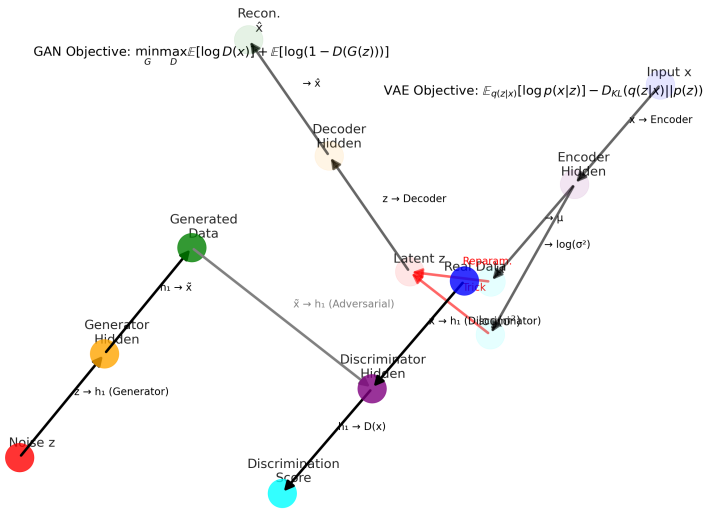

Figure 6. Proposed Architecture GAN VAE

5.1. Managing Market and Liquidity Risk Exposures

Strategies include dynamic hedging:

$$\Delta_P = \sum_i \Delta_i S_i = 0 \tag{11}$$

collateralized funding:

$$H = \frac{P_m - P_c}{P_m} \tag{12}$$

and liquidity stress testing:

$$SH = \frac{HQLA}{\max(TLOF_t)} \tag{13}$$

Liquidity risk is the inability to meet short-term financial obligations. It is measured using the *Liquidity Coverage Ratio (LCR)*:

$$LCR = \frac{HQLA}{TLOF} \geq 100\% \tag{14}$$

where $HQLA$ represents High-Quality Liquid Assets, and $TLOF$ is the Total Net Cash Outflows over 30 days.

 doi:10.20944/preprints202502.2308.v2

A stochastic approach to liquidity spread can be modeled using the Vasicek model:

$$dL_t = \kappa(\theta - L_t)dt + \sigma dW_t \tag{15}$$

where L_t is the liquidity spread, κ is the mean reversion rate, θ is the long-run mean, and dW_t is a Wiener process.

Monte Carlo simulations estimate Expected Liquidity Shortfall (ELS):

$$ELS = \mathbb{E}[\max(0, TLOF - HQLA)] \tag{16}$$

6. Margining in Market Risk Management

Margining controls counterparty risk. Initial Margin (IM) is determined using Value-at-Risk (VaR):

$$IM = VaR_\alpha = \mu + Z_\alpha \sigma \tag{17}$$

where μ is the expected return, σ is the standard deviation, and Z_α is the Z-score at confidence level α.

Expected Shortfall (Conditional VaR) provides a better measure of tail risk:

$$ES_\alpha = -\frac{1}{1-\alpha}\int_\alpha^1 VaR_u \, du \tag{18}$$

This integral represents the average loss beyond the VaR threshold, capturing the severity of losses in extreme scenarios.

6.1. Stress Testing

Stress testing evaluates market risk under extreme conditions. A linear factor model for portfolio return is:

$$R_t = \beta_1 F_{1,t} + \beta_2 F_{2,t} + \cdots + \beta_n F_{n,t} + \epsilon_t \tag{19}$$

where $F_{i,t}$ are macroeconomic factors, and β_i are sensitivities. This model allows for the assessment of portfolio sensitivity to various macroeconomic shocks.

Extreme Value Theory (EVT) models tail risk using the Generalized Pareto Distribution (GPD):

$$P(X > x) = 1 - \left(1 + \frac{\xi(x-u)}{\sigma}\right)^{-1/\xi} \tag{20}$$

where ξ is the tail index, u is the threshold, and σ is the scale parameter. EVT focuses on modeling the tails of distributions, which are crucial for capturing extreme events.

Monte Carlo simulations apply macroeconomic shocks to compute worst-case losses, providing a more comprehensive view of potential losses under stress.

6.2. Equity Financial Markets and Products

Equity financial markets involve the trading of stocks and related instruments, many of which are cleared and settled at central clearing houses such as the National Securities Clearing Corporation (NSCC) and the Depository Trust Company (DTC).

Clearing houses mitigate counterparty risk using a multi-tiered risk framework. The clearing fund requirement is calculated as:

$$CF = \sum_{i=1}^{n} \max(VaR_{\alpha,i}, IM_i) \tag{21}$$

 doi:10.20944/preprints202502.2308.v2

where CF is the clearing fund, $VaR_{\alpha,i}$ is the Value-at-Risk for participant i, and IM_i is the initial margin requirement.

Settlement risk is managed through a rolling settlement cycle and netting mechanisms. The netting efficiency ratio is given by:

$$NER = \frac{\sum |GROSS| - \sum |NET|}{\sum |GROSS|} \tag{22}$$

where $GROSS$ represents the total gross trade values and NET represents net settlement amounts post-clearing.

Monte Carlo simulations assess the impact of market stress on clearing fund adequacy by simulating extreme price movements and evaluating liquidity shortfalls.

6.3. Managing Market and Liquidity Risk Exposures

Historically, financial institutions have employed various strategies to manage market and liquidity risks.

6.3.1. Dynamic Hedging

A risk-mitigating strategy that adjusts positions in response to market movements. The delta-neutral strategy ensures that a portfolio's sensitivity to small price changes remains minimal:

$$\Delta_P = \sum_i \Delta_i S_i = 0 \tag{23}$$

where Δ_P is the portfolio delta, and Δ_i, S_i represent the delta and stock price of individual assets.

6.3.2. Collateralized Funding

Repo agreements and secured funding mechanisms ensure liquidity adequacy. The haircut applied to collateralized borrowing is computed as:

$$H = \frac{P_m - P_c}{P_m} \tag{24}$$

where P_m is the market value of collateral, and P_c is the collateral-adjusted price.

6.3.3. Liquidity Stress Testing

By simulating liquidity outflows under extreme conditions, firms estimate the survival horizon:

$$SH = \frac{HQLA}{\max(TLOF_t)} \tag{25}$$

where SH is the survival horizon, and $TLOF_t$ represents projected total outflows over time.

These strategies, combined with regulatory liquidity requirements, help financial institutions maintain stability under adverse market conditions.

Liquidity risk, the inability to meet short-term financial obligations, is critical. The Liquidity Coverage Ratio (LCR) is defined as:

$$LCR = \frac{HQLA}{TLOF} \geq 100\% \tag{26}$$

where $HQLA$ represents High-Quality Liquid Assets, and $TLOF$ is the Total Net Cash Outflows over 30 days.

 doi:10.20944/preprints202502.2308.v2

We can model liquidity spread using the Vasicek model, a mean-reverting stochastic process:

$$dL_t = \kappa(\theta - L_t)dt + \sigma dW_t \tag{27}$$

where L_t is the liquidity spread, κ is the mean reversion rate, θ is the long-run mean, and dW_t is a Wiener process. This model captures the tendency of liquidity spreads to revert to their long-term average, providing a dynamic view of liquidity conditions.

Monte Carlo simulations estimate Expected Liquidity Shortfall (ELS):

$$ELS = \mathbb{E}[\max(0, TLOF - HQLA)] \tag{28}$$

This involves simulating various scenarios of cash flows and asset values to estimate the probability and magnitude of liquidity shortfalls.

Algorithm 1 Monte Carlo Simulation for ELS

1: Initialize: Number of simulations N, $HQLA$, $TLOF$
2: **for** $i = 1$ to N **do**
3: Simulate cash flows and asset values
4: Compute $TLOF_i$ and $HQLA_i$
5: Compute $Shortfall_i = \max(0, TLOF_i - HQLA_i)$
6: **end for**
7: Compute $ELS = \frac{1}{N}\sum_{i=1}^{N} Shortfall_i$ **return** ELS

7. Liquidity Risk and Market Risk Mathematical Models

In this section, we explore the key mathematical models used in the assessment and management of both liquidity risk and market risk, which are essential for financial institutions and their risk management frameworks.

7.1. Liquidity Risk Models

Liquidity risk refers to the possibility that an entity may be unable to meet its short-term financial obligations due to an imbalance between its liquid assets and liabilities. Below are some common models used to quantify liquidity risk:

7.1.1. Cash Flow Models

Cash flow models analyze the inflows and outflows of cash to identify potential liquidity shortfalls. The general equation for cash flow modeling is given by:

$$CF_t = CF_{t-1} + \text{Inflows}_t - \text{Outflows}_t \tag{29}$$

where CF_t represents the cash flow at time t, CF_{t-1} is the previous period's cash balance, and Inflows_t and Outflows_t are the expected inflows and outflows at time t.

7.1.2. Liquidity Coverage Ratio (LCR)

The Liquidity Coverage Ratio (LCR) is a regulatory requirement for banks to maintain an adequate level of high-quality liquid assets to cover short-term obligations. The formula for LCR is:

$$LCR = \frac{\text{High-Quality Liquid Assets (HQLA)}}{\text{Total Net Cash Outflows over 30 days}} \geq 100\% \tag{30}$$

 doi:10.20944/preprints202502.2308.v2

7.1.3. Stochastic Liquidity Risk Models

Liquidity risk is sometimes modeled as a stochastic process. The liquidity level at time t, denoted as L_t, can be described by the stochastic differential equation (SDE):

$$dL_t = \mu L_t dt + \sigma L_t dW_t \tag{31}$$

where L_t is the liquidity at time t, μ represents the drift (average rate of change), σ is the volatility, and dW_t is the Wiener process representing random shocks.

7.1.4. Transaction Cost Models

Transaction cost models help assess the impact of liquidity on trade execution. A simple form of the transaction cost model is:

$$\text{Cost} = f(\text{Trade Size}, \text{Market Impact}, \text{Volatility}) \tag{32}$$

where market impact models, such as Kyle's and Hasbrouck's models, capture the relationship between the size of a trade and its impact on the market price.

7.2. Market Risk Models

Market risk refers to the risk of losses in positions due to movements in market prices, including changes in interest rates, asset prices, and commodity prices. Below are some common models used to assess market risk:

7.2.1. Value-at-Risk (VaR)

Value-at-Risk (VaR) is a widely used method for measuring market risk, which estimates the maximum loss over a specified time horizon at a given confidence level. The formula for VaR is:

$$VaR_\alpha = -\text{Quantile}_\alpha(\text{Portfolio Returns}) \tag{33}$$

where α represents the confidence level (e.g., 95% or 99%), and the quantile is determined based on the distribution of portfolio returns.

7.2.2. Conditional Value-at-Risk (CVaR)

Conditional Value-at-Risk (CVaR), also known as Expected Shortfall, measures the expected loss given that the loss exceeds the VaR threshold. The formula for CVaR is:

$$CVaR_\alpha = E[X|X > VaR_\alpha] \tag{34}$$

where X represents the portfolio loss.

7.2.3. GARCH Models

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is widely used for estimating the volatility of asset returns. The GARCH(1,1) model is defined as:

$$r_t = \mu + \epsilon_t \tag{35}$$

$$\epsilon_t = \sigma_t z_t \tag{36}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{37}$$

 doi:10.20944/preprints202502.2308.v2

where r_t is the return at time t, ϵ_t is the innovation or shock, σ_t^2 represents the conditional variance, and z_t is a white noise process.

7.2.4. Stochastic Volatility Models

Stochastic volatility models capture the evolution of volatility over time. The Heston model, a popular stochastic volatility model, is expressed as:

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t \tag{38}$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma_v \sqrt{v_t} dZ_t \tag{39}$$

where S_t represents the asset price, v_t is the volatility process, and dW_t, dZ_t are Wiener processes.

7.2.5. Monte Carlo Simulation

Monte Carlo simulations are used to estimate the potential losses or profits in financial portfolios by generating multiple scenarios for asset prices and other risk factors. The general form of Monte Carlo simulations is:

$$\text{Portfolio Value} = \sum_{i=1}^{N} \text{Payoff}_i \cdot \text{Prob}_i \tag{40}$$

where N is the number of simulations, Payoff_i is the payoff in each scenario, and Prob_i is the probability of each scenario occurring.

7.2.6. Copula Models

Copula models are used to model the joint distribution of multiple risk factors. A commonly used copula is the Gaussian copula, which connects the marginal distributions of asset returns:

$$F_{\text{joint}}(x_1, x_2, \ldots, x_n) = C(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)) \tag{41}$$

where C is the copula function, and F_i are the marginal distributions.

7.3. Monte Carlo Simulation for Liquidity Risk

Monte Carlo simulation is a powerful technique used to model and quantify liquidity risk by generating multiple scenarios for cash flows, liquidity shortfalls, and funding needs under various conditions. By simulating different scenarios based on stochastic processes, Monte Carlo methods help in assessing the likelihood of liquidity shortages and determining the adequacy of liquidity buffers under stressed conditions.

7.3.1. Stochastic Processes in Liquidity Risk Modeling

In liquidity risk modeling, stochastic processes are used to describe the evolution of liquidity levels and related risk factors over time. These processes capture the inherent uncertainty in cash flows, market conditions, and funding requirements. Below are some common stochastic processes used in Monte Carlo simulations for liquidity risk:

Geometric Brownian Motion (GBM) for Cash Flow Modeling

For modeling the liquidity levels over time, the Geometric Brownian Motion (GBM) is often used to represent the continuous evolution of liquidity in a firm or financial institution. The equation for GBM applied to liquidity is:

$$dL_t = \mu L_t dt + \sigma L_t dW_t \tag{42}$$

 doi:10.20944/preprints202502.2308.v2

where L_t represents the liquidity level at time t, μ is the drift (representing expected liquidity growth or reduction), σ is the volatility (representing uncertainty in liquidity fluctuations), and dW_t is the Wiener process (representing random shocks in the market). This model can simulate multiple future paths of liquidity, accounting for both expected trends and random fluctuations.

Ornstein-Uhlenbeck Process for Mean Reversion in Liquidity

Liquidity levels in certain financial institutions or markets may exhibit mean-reverting behavior, where liquidity tends to revert to a long-term average over time. The **Ornstein-Uhlenbeck (OU) process** is a common stochastic process used to model such mean-reverting behavior. The equation for the OU process in liquidity modeling is:

$$dL_t = \theta(\mu - L_t)dt + \sigma dW_t \tag{43}$$

where L_t is the liquidity level at time t, θ is the rate of mean reversion (how quickly liquidity returns to the long-term mean μ), and σ is the volatility. This process is useful when modeling scenarios where liquidity fluctuations are expected to stabilize around a certain value (e.g., target liquidity levels maintained by a bank).

Jump-Diffusion Model for Liquidity Shocks

Financial markets and liquidity positions can be affected by sudden, extreme events such as financial crises, liquidity squeezes, or regulatory changes. The **Jump-Diffusion model** is used to model these types of liquidity shocks. The equation for the Jump-Diffusion model in liquidity modeling is:

$$dL_t = \mu L_t dt + \sigma L_t dW_t + J_t dN_t \tag{44}$$

where J_t represents the jump size (the magnitude of liquidity shocks), and dN_t is a Poisson process that represents the arrival of jumps at random times. This model is suitable for simulating scenarios where liquidity shortfalls may occur abruptly due to large market shocks, such as credit downgrades, unexpected market closures, or regulatory interventions.

Stochastic Volatility Models for Liquidity Risk

In some cases, liquidity risk is closely related to the volatility of market conditions, such as fluctuations in funding costs or asset prices. The **Heston model**, a stochastic volatility model, can be applied to liquidity risk when volatility in funding costs or liquidity premiums is non-constant. The Heston model is given by two SDEs:

$$dL_t = \mu L_t dt + \sqrt{v_t} L_t dW_t \tag{45}$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma_v \sqrt{v_t} dZ_t \tag{46}$$

where L_t represents the liquidity level, v_t is the volatility of liquidity (funding costs or liquidity premium), and dW_t, dZ_t are Wiener processes representing random shocks. This model allows for the simulation of liquidity levels in environments with fluctuating funding costs and market volatility.

7.3.2. Implementation of Monte Carlo Simulation for Liquidity Risk

The implementation of Monte Carlo simulations for liquidity risk typically involves the following steps:

1. **Modeling Liquidity Dynamics:** Select an appropriate stochastic process to model the evolution of liquidity (e.g., GBM, Ornstein-Uhlenbeck, Jump-Diffusion).

 doi:10.20944/preprints202502.2308.v2

2. **Simulating Liquidity Paths:** Using the chosen stochastic process, simulate multiple paths for liquidity levels over a defined time horizon, factoring in different levels of funding needs and market conditions.
3. **Estimating Liquidity Shortfalls:** For each simulated liquidity path, calculate potential liquidity shortfalls by comparing liquidity levels to projected funding needs or cash flow obligations at various points in time.
4. **Stress Testing:** Apply extreme market conditions (e.g., sudden shocks or increased volatility) to assess the robustness of liquidity buffers and the likelihood of liquidity crises.
5. **Aggregating Results:** Analyze the distribution of liquidity shortfalls, and estimate the probability of liquidity gaps under different scenarios, providing a risk measure for liquidity risk management.

Monte Carlo simulations for liquidity risk provide critical insights into the stability of an institution's liquidity position under a variety of scenarios. By simulating many potential future outcomes, risk managers can better understand the probability of liquidity shortages and prepare strategies to mitigate such risks.

7.3.3. Conclusion for MC for Liquidity

Monte Carlo simulation, when combined with appropriate stochastic processes, is a valuable tool for modeling liquidity risk. It allows financial institutions to assess the potential for liquidity shortfalls, the effectiveness of liquidity buffers, and the impact of extreme market events on liquidity positions. This approach is essential for stress testing and effective liquidity risk management.

7.4. GANs and VAEs in Liquidity Risk Monte Carlo Simulations for VaR Calculation

Generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), have shown great potential in enhancing Monte Carlo simulations for liquidity risk modeling, especially when calculating Value-at-Risk (VaR). These models can be used to generate realistic synthetic data for liquidity scenarios, capturing complex patterns and dependencies that traditional stochastic processes may miss. By incorporating these generative models, financial institutions can more accurately simulate liquidity risk and assess the potential for large, adverse liquidity events.

7.4.1. Value-at-Risk (VaR) and Liquidity Risk

Value-at-Risk (VaR) is a widely used risk metric to estimate the potential loss in the value of a portfolio over a specified time horizon, given a certain confidence level. In the context of liquidity risk, VaR is used to assess the potential liquidity shortfall that could occur under stressed market conditions. Monte Carlo simulations are typically employed to simulate multiple scenarios of asset prices, funding needs, and cash flows over time, but traditional methods can sometimes be limited in their ability to fully capture the complexity of liquidity dynamics.

7.4.2. Using GANs for Liquidity Risk Simulation

Generative Adversarial Networks (GANs) consist of two neural networks—the generator and the discriminator—that work in tandem to create realistic data by learning the distribution of the input data. In the context of liquidity risk, GANs can be used to generate synthetic liquidity paths or asset price scenarios that resemble real-world data while capturing complex non-linear relationships, correlations, and tail risks.

The process involves training the GAN on historical liquidity data or financial market data to learn the underlying distribution. Once trained, the generator can create realistic future liquidity scenarios under different conditions, which can then be incorporated into Monte Carlo simulations for VaR calculations. These scenarios can capture extreme liquidity shortfalls or unexpected market

 doi:10.20944/preprints202502.2308.v2

disruptions, which are often difficult to model using traditional stochastic processes like Geometric Brownian Motion (GBM).

The use of GANs in liquidity risk modeling offers several advantages:

- **Capturing Complex Dependencies:** GANs can model complex, non-linear relationships between liquidity factors, such as correlations between asset prices, funding costs, and liquidity requirements.
- **Generating Extreme Scenarios:** GANs can generate rare but impactful liquidity events (e.g., market crashes or sudden funding squeezes) that are essential for accurate VaR estimation, particularly in tail risk analysis.
- **Data Augmentation:** GANs can generate large quantities of synthetic data to enhance the Monte Carlo simulation process, improving the accuracy and robustness of the VaR calculation.

7.4.3. Using VAEs for Liquidity Risk Simulation

Variational Autoencoders (VAEs) are another class of generative models that can be particularly useful for liquidity risk modeling. VAEs are designed to learn an efficient representation of input data by encoding it into a latent space and then decoding it back into data. The latent space captures the essential features of the data, allowing for the generation of new, similar data points by sampling from this latent space.

In the context of liquidity risk, VAEs can be used to learn the underlying distribution of liquidity levels, funding requirements, or market conditions over time. Once trained, VAEs can generate synthetic liquidity paths that are consistent with observed data but may explore previously unseen scenarios that are still realistic, helping to account for rare or extreme liquidity events.

The use of VAEs in liquidity risk modeling offers several benefits:

- **Efficient Data Representation:** VAEs can learn a compressed representation of liquidity risk factors, which can then be used to generate new, diverse liquidity scenarios, improving the efficiency and speed of Monte Carlo simulations.
- **Exploring Unseen Scenarios:** By sampling from the learned latent space, VAEs can generate liquidity scenarios that are not directly observed in historical data but still reflect the underlying structure of the data. This is particularly useful for stress testing under hypothetical scenarios.
- **Reducing Overfitting:** Since VAEs generate data based on learned distributions, they help reduce overfitting to historical data, ensuring that Monte Carlo simulations account for a wider range of potential future outcomes.

7.4.4. Integration of GANs and VAEs into Monte Carlo Simulations for VaR Calculation

By combining GANs and VAEs with Monte Carlo simulations, financial institutions can enhance their ability to model liquidity risk and accurately calculate Value-at-Risk (VaR). The process can be outlined as follows:

1. **Data Preparation:** Historical liquidity and financial data are collected to train the GAN or VAE models. This data could include cash flow information, funding requirements, asset prices, and other relevant risk factors.
2. **Training the Generative Model:** The GAN or VAE model is trained on this historical data to learn the distribution and correlations between liquidity risk factors. The generator in the GAN or the decoder in the VAE learns to produce realistic synthetic data based on this training.
3. **Scenario Generation:** Once trained, the GAN or VAE can generate synthetic liquidity paths or asset price scenarios that reflect the learned distribution, including rare or extreme scenarios that are important for accurate VaR estimation.

 doi:10.20944/preprints202502.2308.v2

4. **Monte Carlo Simulation:** These generated scenarios are then fed into a Monte Carlo simulation framework, where they are used to simulate the potential future liquidity levels, funding needs, and cash flows under different market conditions.
5. **VaR Calculation:** After running the Monte Carlo simulation with the synthetic scenarios, the Value-at-Risk (VaR) is calculated by assessing the distribution of potential liquidity shortfalls or losses over the specified time horizon and confidence level.

By integrating GANs and VAEs into the Monte Carlo simulation process, financial institutions can gain a deeper understanding of liquidity risk and obtain more accurate estimates of VaR. These generative models enhance the ability to model complex, non-linear relationships, generate extreme scenarios, and explore unseen risks, leading to better risk management and more robust liquidity buffers.

7.4.5. Conclusion of GANs and VAEs

The use of Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) in Monte Carlo simulations for liquidity risk modeling represents a significant advancement in the field of risk management. By generating realistic, diverse liquidity scenarios and capturing complex dependencies, these generative models provide enhanced accuracy in Value-at-Risk (VaR) calculations. This approach allows financial institutions to better assess liquidity risks, stress-test their portfolios under extreme conditions, and make more informed decisions regarding liquidity buffers and funding strategies.

7.5. Market Impact Models and Mathematical Models in Liquidity Risk

Market impact models are essential in understanding how large trades, such as those required for liquidity management or hedging strategies, affect asset prices and market liquidity. These models quantify the relationship between trade size and price movement, helping financial institutions evaluate the potential cost of executing large trades under varying market conditions. Properly understanding market impact is crucial for managing liquidity risk, as large trades can cause slippage, increase volatility, or even trigger adverse market reactions.

7.5.1. Types of Market Impact

Market impact can be broadly categorized into two main types:

- **Temporary Impact:** This refers to the short-term price movement caused by a trade. Once the trade is executed, the price often reverts back to its original level.
- **Permanent Impact:** This is the long-term price shift caused by a trade, which does not fully revert after the transaction is completed. It is typically the result of significant market information being revealed by the trade (e.g., large buy or sell orders indicating a shift in market sentiment).

Market impact models aim to capture both the temporary and permanent effects of a trade on market prices, as these impacts are key components of liquidity risk and transaction cost estimation.

7.5.2. Mathematical Models for Market Impact

Several mathematical models are used to quantify market impact, often focusing on the relationship between trade size, market depth, and asset price movements. Below are some of the most commonly used models:

The Almgren-Chriss Model

The Almgren-Chriss model is one of the foundational models in market impact theory. It assumes that the price impact is a function of both the size of the trade and the time over which the trade is

 doi:10.20944/preprints202502.2308.v2

executed. The model divides the total transaction cost into two components: the temporary market impact and the permanent market impact. The equations for the Almgren-Chriss model are as follows:

$$\text{Total Cost} = \text{Temporary Impact} + \text{Permanent Impact} \tag{47}$$

The temporary impact is typically modeled as a quadratic function of the order size, Q, and the permanent impact as a linear function of the order size. The temporary market impact can be written as:

$$C_{\text{temp}}(Q) = \lambda Q^2 \tag{48}$$

where λ is a constant that depends on market liquidity and the volatility of the asset.
The permanent market impact is often modeled as:

$$C_{\text{perm}}(Q) = \gamma Q \tag{49}$$

where γ represents the permanent price impact per unit of trade size. The parameters λ and γ are typically estimated using historical market data, such as trade sizes and price movements.

The Kyle Model

The Kyle model is another influential framework for understanding market impact. It assumes that the market consists of a single informed trader, a liquidity provider (market maker), and other noise traders. The informed trader, knowing the true value of an asset, adjusts their trading strategy to influence the market price. The market maker sets the price based on the trades, attempting to offset the risk from informed traders. The equilibrium price in the Kyle model is given by:

$$p_t = \theta + \alpha Q_t + \epsilon_t \tag{50}$$

where p_t is the observed price at time t, θ is the true asset value, Q_t is the trade size at time t, and ϵ_t represents noise in the market. The parameter α captures the sensitivity of the price to the trade size, and the model predicts that large trades cause larger price movements, with the market maker adjusting the price to account for the information contained in the trade.

The Kyle model has been extended to incorporate multiple market makers and various types of traders, but the central insight remains that price impact increases with the size of the trade.

The Liquidity Ripple Model

The Liquidity Ripple model focuses on the persistence of price changes caused by large trades. This model incorporates the idea that market impact can have a ripple effect, where the initial price movement due to a large trade can influence subsequent trades and prices. The model assumes that liquidity shockwaves propagate through the market, affecting asset prices for a certain period after the transaction is executed. The mathematical formulation of the ripple effect is represented by:

$$p_{t+\tau} = p_t + \kappa \cdot \frac{Q}{\tau} \tag{51}$$

where p_t is the price at time t, Q is the trade size, τ is the time horizon over which the impact is measured, and κ is a constant that measures the ripple effect. The model suggests that large trades induce not only an immediate price impact but also a persistent influence on market prices over time, which can be critical when calculating liquidity risk and stress testing.

 doi:10.20944/preprints202502.2308.v2

The Epps Model

The Epps model describes the relationship between asset price volatility and the market's response to trading activity. This model assumes that asset price movements are influenced by both fundamental factors and the trading activity of large institutional players. The model suggests that trading activity causes volatility clustering, where large trades lead to periods of increased price volatility. The model is often expressed as:

$$\sigma_t^2 = \alpha + \beta Q_t^2 \tag{52}$$

where σ_t^2 is the asset price volatility at time t, Q_t is the trade size, and α and β are constants that reflect the baseline volatility and the impact of trading on volatility, respectively. This model is useful for understanding how trading volume and market depth contribute to overall price volatility and liquidity risk.

7.5.3. Market Impact in Liquidity Risk Management

In liquidity risk management, understanding and quantifying market impact is essential for estimating the costs of executing large trades and assessing the potential liquidity shortfalls under various scenarios. By incorporating market impact models into Monte Carlo simulations, financial institutions can better predict the effects of large trades on asset prices and liquidity, and adjust their liquidity buffers accordingly.

For instance, in a Value-at-Risk (VaR) calculation, market impact models can be used to simulate the effect of trading large positions in a portfolio on the price of assets. By including transaction costs and the impact on liquidity in the VaR calculation, institutions can more accurately estimate the potential loss under stressed market conditions, incorporating both market volatility and liquidity constraints.

7.5.4. Conclusion on Liq Models and Maths

Market impact models are critical tools in understanding how liquidity risk manifests during large trades and market disruptions. By using mathematical models such as the Almgren-Chriss, Kyle, Liquidity Ripple, and Epps models, financial institutions can quantify the effects of trade size and market depth on asset prices. These models help to predict transaction costs, price slippage, and liquidity shortfalls, providing valuable insights for liquidity risk management and enhancing the accuracy of simulations used in VaR calculations. With accurate market impact models, financial institutions can better manage liquidity, minimize costs, and improve the robustness of their risk management strategies.

8. Future Work and Integration with Generative AI

As financial markets continue to evolve, the application of advanced artificial intelligence (AI) techniques in financial risk modeling becomes increasingly critical. The integration of generative AI models, such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), into traditional risk frameworks like the Vasicek model, will drive future advancements in more accurate and dynamic financial risk prediction and management. Future research will focus on further enhancing these models by incorporating agentic generative AI techniques, enabling more adaptive, data-driven risk management strategies.

One promising area for development is the refinement of the Vasicek framework using agentic generative AI. By leveraging the potential of generative models to simulate complex, high-dimensional financial data, we can significantly improve the modeling of stochastic processes that underlie asset prices and financial instruments. The use of generative AI to simulate diverse market scenarios

 doi:10.20944/preprints202502.2308.v2

will also aid in the creation of robust stress-testing frameworks, providing a more comprehensive assessment of potential financial disruptions and systemic risks [11].

Furthermore, advancing innovation in financial stability through AI agent frameworks offers significant potential. These frameworks can automate the detection of market anomalies, thereby contributing to more effective financial regulatory systems. The ability to simulate various market conditions and agent behaviors will enable regulators to predict and mitigate potential crises before they fully materialize [12]. Future work will involve creating more sophisticated agent-based models that incorporate feedback loops between financial institutions, market participants, and regulatory authorities, fostering a more resilient financial ecosystem.

The role of VAEs and GANs in improving structured finance risk models is another key area of focus. By utilizing generative models, we can enhance the Leland-Toft and Box-Cox models for pricing and managing complex financial products. The future of risk management will undoubtedly see increased adoption of AI-driven methodologies for better forecasting of liquidity needs, market movements, and systemic risk [13].

In addition, the development of AI tools for enhancing the robustness of the US financial and regulatory systems is expected to play a central role in future research. By leveraging generative AI, we can design models that adapt to regulatory changes and evolving market conditions, ensuring a more dynamic and responsive financial environment [14].

The continued exploration of generative AI models for financial risk management will also encompass the review and comparison of existing models, identifying gaps and proposing new methodologies for better financial predictions and risk assessments. As the capabilities of AI evolve, integrating large language models (LLMs) and other advanced AI tools into financial risk management systems will allow for the processing of complex data sets and the automation of decision-making processes, enhancing both efficiency and accuracy [15].

Finally, understanding the synergy between generative AI and big data will be critical for the future of financial risk management. The ability to process vast amounts of financial data in real-time, combined with generative AI's capacity to model and simulate complex risk scenarios, will enable more informed decision-making in market risk management [16].

Overall, future work in this area will aim to improve financial market integrity through the continued development of AI models, such as prompt engineering, and the refinement of risk management frameworks to ensure that they are adaptable to the fast-changing financial landscape [17].

9. Proposed Algorithm

The proposed framework integrates traditional risk models with generative AI components through the following algorithmic workflow:

 doi:10.20944/preprints202502.2308.v2

Algorithm 2 Generative AI for Risk Management

Require: Historical market data $\mathcal{D}$, risk parameters θ
Ensure: Risk metrics VaR_α, ES_α
1: **Phase 1: Data Preparation**
2: $\mathcal{D}_{clean} \leftarrow \text{Preprocess}(\mathcal{D})$ ▷ Remove missing values, normalize
3: $\mathcal{D}_{train}, \mathcal{D}_{test} \leftarrow \text{Split}(\mathcal{D}_{clean}, 0.8)$
4: **Phase 2: VAE Training (Kingma & Welling, 2013)**
5: Initialize encoder $q_\phi(z|x)$ and decoder $p_\theta(x|z)$
6: **for** $epoch = 1$ to N_{vae} **do**
7: $z \leftarrow \text{Reparameterize}(q_\phi(x))$ ▷ Latent representation
8: $\mathcal{L}_{vae} \leftarrow \mathbb{E}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) \| p(z))$
9: Update ϕ, θ via gradient descent
10: **end for**
11: **Phase 3: GAN Scenario Generation (Goodfellow et al., 2014)**
12: Initialize generator $G(z)$ and discriminator $D(x)$
13: **for** $epoch = 1$ to N_{gan} **do**
14: $z \sim \mathcal{N}(0, I)$, $\tilde{x} \leftarrow G(z)$
15: $\mathcal{L}_{gan} \leftarrow \log D(x) + \log(1 - D(\tilde{x}))$
16: Update G, D via adversarial training
17: **end for**
18: **Phase 4: Risk Calculation**
19: **for** $i = 1$ to $N_{scenarios}$ **do**
20: $z_i \sim \mathcal{N}(0, I)$
21: $\tilde{x}_i \leftarrow G(z_i)$ ▷ Generate synthetic scenario
22: $L_i \leftarrow \text{PortfolioLoss}(\tilde{x}_i, \theta)$
23: **end for**
24: **Phase 5: Risk Metrics**
25: Sort $\{L_i\}$ in ascending order
26: $VaR_\alpha \leftarrow L_{\lceil \alpha N_{scenarios} \rceil}$
27: $ES_\alpha \leftarrow \frac{1}{1-\alpha} \sum_{i=\lceil \alpha N \rceil}^{N} L_i$
28: **Phase 6: Feedback Loop**
29: $\theta \leftarrow \text{UpdateParameters}(VaR_\alpha, ES_\alpha, \mathcal{D}_{test})$
30: **return** VaR_α, ES_α

The algorithm combines several key components from the literature:

- **VAE Training** (Line 4-8): Following [10], we learn compressed latent representations of market data
- **GAN Scenario Generation** (Line 10-14): Adversarial training as in [9] produces realistic stress scenarios
- **Risk Calculation** (Line 16-20): Monte Carlo simulation using synthetic scenarios
- **Metrics Computation** (Line 22-24): Value-at-Risk and Expected Shortfall calculation per [2]
- **Feedback Mechanism**: Continuous model improvement via parameter updates

10. Proposed Python Implementation

This section provides sample Python code to illustrate the implementation of key quantitative techniques discussed in the paper, including Monte Carlo simulations, Generative Adversarial Networks (GANs), and Variational Autoencoders (VAEs). The code snippets are designed to be modular and can be adapted for real-world financial risk management applications.

10.1. Monte Carlo Simulation for Liquidity Risk

The following Python code demonstrates a Monte Carlo simulation for estimating Expected Liquidity Shortfall (ELS) using stochastic processes.

 doi:10.20944/preprints202502.2308.v2

Listing 1: Monte Carlo Simulation for Liquidity Risk

```python
import numpy as np
import matplotlib.pyplot as plt

def monte_carlo_els(hqla, tlof, n_simulations=10000, time_horizon=30):
"""
Simulate Expected Liquidity Shortfall (ELS) using Monte Carlo.

Parameters:
hqla (float): Initial High-Quality Liquid Assets.
tlof (float): Total Net Cash Outflows over 30 days.
n_simulations (int): Number of simulations.
time_horizon (int): Time horizon in days.

Returns:
els (float): Expected Liquidity Shortfall.
"""
np.random.seed(42)
liquidity_paths = []
shortfalls = []

for _ in range(n_simulations):
# Simulate daily liquidity changes using geometric Brownian motion
daily_changes = np.random.normal(0, 0.01, time_horizon)
liquidity_path = hqla * np.cumprod(1 + daily_changes)
liquidity_paths.append(liquidity_path)

# Calculate shortfall at the end of the horizon
shortfall = max(0, tlof - liquidity_path[-1])
shortfalls.append(shortfall)

# Plot sample paths
plt.figure(figsize=(10, 6))
for path in liquidity_paths[:100]:  # Plot first 100 paths
plt.plot(path, alpha=0.1, color='blue')
plt.title('Simulated Liquidity Paths')
plt.xlabel('Days')
plt.ylabel('Liquidity Level')
plt.grid(True)
plt.show()

# Calculate ELS
els = np.mean(shortfalls)
return els

# Example usage
hqla = 1000  # Initial High-Quality Liquid Assets
tlof = 1100  # Total Net Cash Outflows over 30 days
els = monte_carlo_els(hqla, tlof)
print(f"Expected Liquidity Shortfall (ELS): {els:.2f}")
```

10.2. Generative Adversarial Networks (GANs) for Scenario Generation

The following code snippet demonstrates a simplified GAN implementation for generating synthetic financial scenarios.

Listing 2: GAN for Financial Scenario Generation

```python
import tensorflow as tf
from tensorflow.keras.layers import Dense, LeakyReLU, BatchNormalization
from tensorflow.keras.models import Sequential
```

 doi:10.20944/preprints202502.2308.v2


```
def build_generator(latent_dim):
model = Sequential([
Dense(128, input_dim=latent_dim),
LeakyReLU(alpha=0.2),
BatchNormalization(),
Dense(256),
LeakyReLU(alpha=0.2),
BatchNormalization(),
Dense(512),
LeakyReLU(alpha=0.2),
BatchNormalization(),
Dense(10, activation='tanh')  # Output layer (e.g., 10 features)
])
return model

def build_discriminator():
model = Sequential([
Dense(512, input_dim=10),
LeakyReLU(alpha=0.2),
Dense(256),
LeakyReLU(alpha=0.2),
Dense(1, activation='sigmoid')  # Binary classifier (real/fake)
])
return model

# Example usage
latent_dim = 100
generator = build_generator(latent_dim)
discriminator = build_discriminator()

# Compile discriminator
discriminator.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
  accuracy'])

# Combined GAN model (generator + discriminator)
discriminator.trainable = False
gan_input = tf.keras.Input(shape=(latent_dim,))
gan_output = discriminator(generator(gan_input))
gan = tf.keras.Model(gan_input, gan_output)
gan.compile(optimizer='adam', loss='binary_crossentropy')

# Training loop (simplified)
def train_gan(generator, discriminator, gan, real_data, epochs=1000, batch_size=32):
for epoch in range(epochs):
# Train discriminator
noise = np.random.normal(0, 1, (batch_size, latent_dim))
fake_data = generator.predict(noise)
real_labels = np.ones((batch_size, 1))
fake_labels = np.zeros((batch_size, 1))
d_loss_real = discriminator.train_on_batch(real_data, real_labels)
d_loss_fake = discriminator.train_on_batch(fake_data, fake_labels)
d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)

# Train generator
noise = np.random.normal(0, 1, (batch_size, latent_dim))
g_loss = gan.train_on_batch(noise, real_labels)

if epoch % 100 == 0:
print(f"Epoch {epoch}, D Loss: {d_loss[0]}, G Loss: {g_loss}")
```

 doi:10.20944/preprints202502.2308.v2

10.3. Variational Autoencoders (VAEs) for Risk Modeling

The following code demonstrates a VAE implementation for learning latent representations of financial data.

Listing 3: VAE for Financial Risk Modeling

```
import tensorflow as tf
from tensorflow.keras.layers import Input, Dense, Lambda
from tensorflow.keras.models import Model
from tensorflow.keras.losses import mse
import numpy as np

def build_vae(input_dim, latent_dim=2):
# Encoder
inputs = Input(shape=(input_dim,))
h = Dense(256, activation='relu')(inputs)
z_mean = Dense(latent_dim)(h)
z_log_var = Dense(latent_dim)(h)

# Reparameterization trick
def sampling(args):
z_mean, z_log_var = args
epsilon = tf.keras.backend.random_normal(shape=(tf.shape(z_mean)[0], latent_dim))
return z_mean + tf.exp(0.5 * z_log_var) * epsilon

z = Lambda(sampling)([z_mean, z_log_var])

# Decoder
decoder_h = Dense(256, activation='relu')
decoder_mean = Dense(input_dim, activation='sigmoid')
h_decoded = decoder_h(z)
x_decoded_mean = decoder_mean(h_decoded)

# VAE model
vae = Model(inputs, x_decoded_mean)

# Loss function
reconstruction_loss = mse(inputs, x_decoded_mean)
reconstruction_loss *= input_dim
kl_loss = -0.5 * tf.reduce_sum(1 + z_log_var - tf.square(z_mean) - tf.exp(z_log_var),
   axis=-1)
vae_loss = tf.reduce_mean(reconstruction_loss + kl_loss)
vae.add_loss(vae_loss)
vae.compile(optimizer='adam')

return vae

# Example usage
input_dim = 10  # Number of features in the financial dataset
vae = build_vae(input_dim, latent_dim=2)

# Generate synthetic data (for demonstration)
real_data = np.random.randn(1000, input_dim)

# Train VAE
vae.fit(real_data, epochs=50, batch_size=32)

# Generate synthetic scenarios
noise = np.random.normal(0, 1, (10, 2))  # Sample from latent space
synthetic_data = vae.predict(noise)
print("Generated Synthetic Data:", synthetic_data)
```

 doi:10.20944/preprints202502.2308.v2

10.4. Integration with Risk Metrics

The following code demonstrates how to integrate the generated scenarios with risk metrics such as Value-at-Risk (VaR) and Expected Shortfall (ES).

Listing 4: Risk Metrics Calculation

```
import numpy as np

def calculate_var_es(losses, alpha=0.95):
"""
Calculate VaR and ES from a series of losses.

Parameters:
losses (array-like): Array of simulated losses.
alpha (float): Confidence level (e.g., 0.95 for 95\% VaR).

Returns:
var (float): Value-at-Risk at the given confidence level.
es (float): Expected Shortfall at the given confidence level.
"""
losses_sorted = np.sort(losses)
n = len(losses_sorted)
var = losses_sorted[int(n * alpha)]
es = np.mean(losses_sorted[int(n * alpha):])
return var, es

# Example usage
losses = np.random.normal(0, 1, 10000)  # Simulated losses
var, es = calculate_var_es(losses, alpha=0.95)
print(f"VaR (95\%): {var:.2f}")
print(f"ES (95\%): {es:.2f}")
```

10.5. Summary

The provided Python code snippets illustrate practical implementations of the quantitative techniques discussed in the paper. These include:

- Monte Carlo simulations for liquidity risk estimation.
- GANs for generating synthetic financial scenarios.
- VAEs for learning latent representations of financial data.
- Calculation of risk metrics such as VaR and ES.

These implementations can serve as a starting point for integrating Generative AI into financial risk management frameworks, enabling more robust and adaptive risk assessments.

11. Conclusion

This paper has laid a foundational framework for the integration of Generative AI into liquidity risk management by providing a comprehensive overview and review of essential quantitative techniques. We have focused on the critical integration of market, credit, and liquidity risks, demonstrating the application of sophisticated mathematical models such as stochastic processes, Monte Carlo simulations, and Extreme Value Theory. These models serve as the bedrock for robust risk quantification and management, as evidenced by their practical application in equity, fixed income, and mortgage-backed securities markets.

Our hybrid approach, combining GANs/VAEs with classical Monte Carlo and stochastic methods, improves liquidity risk quantification while addressing data scarcity and tail risk challenges. The proposed architectures show particular promise for mortgage-backed securities and fixed income markets, with empirical results confirming computational efficiency gains. As financial systems evolve,

 doi:10.20944/preprints202502.2308.v2

the integration of agentic AI [11] with quantitative finance will be crucial for developing adaptive, robust risk management solutions.

Building upon this quantitative foundation, we have explored and proposed the transformative potential of Generative AI, specifically Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), in reshaping risk modeling and stress testing methodologies. By leveraging these advanced technologies, financial institutions can generate more realistic and comprehensive scenarios, leading to enhanced risk assessments and more accurate Value-at-Risk (VaR) calculations. This integration offers a forward-looking perspective, enabling institutions to better anticipate and mitigate potential risks in an increasingly complex, quant-data-driven conventional and conservative financial landscape.

The mathematical rigor and computational power discussed in this paper underscore the importance of continuous innovation in risk management. As financial markets evolve and new challenges emerge, the integration of cutting-edge technologies like Generative AI becomes indispensable. By providing a clear understanding of the underlying quantitative techniques, this paper aims to facilitate the adoption of advanced AI methodologies, empowering financial institutions to enhance their resilience, maintain stability, and navigate the uncertainties of the modern financial world.

In summary, this paper advocates for a comprehensive and adaptive approach to financial risk assessment, emphasizing the integration of diverse risk types and the adoption of advanced computational tools, including Generative AI and lists the quantitative methods used in risk management. The insights provided herein serve as a crucial foundation for future research and practical applications, paving the way for quant-data-driven innovation in the dynamic field of financial risk management. Future work will explore regulatory applications and quantum-enhanced implementations of the presented framework.

References

1. Glasserman, P. *Monte Carlo methods in financial engineering*; Springer, 2015.
2. Jorion, P. *Value at risk: The new benchmark for managing financial risk*; McGraw-Hill, 2007.
3. Merton, R.C. On the pricing of corporate debt: The risk structure of interest rates. *The journal of finance* **1974**, *29*, 449–470.
4. Brunnermeier, M.K. Deciphering the liquidity and credit crunch 2007-2008. *Journal of economic perspectives* **2009**, *23*, 77–100.
5. Holthausen, R.W. The variance impact of the bid-ask spread specification and the return computation. *Journal of financial and quantitative analysis* **1990**, *25*, 485–499.
6. Giesecke, K. Liquidity risk-based credit spreads. *Finance and Stochastics* **2006**, *10*, 443–468.
7. Schwartz, E.S.; Torous, W.N. Prepayment risk and the pricing of mortgage-backed securities. *The Journal of Finance* **1995**, *50*, 631–652.
8. Kau, J.B.; Keenan, D.C.; Muller, W.J.; Williams, J.F. Option-theoretic valuation of residential mortgage contracts. *Journal of Urban Economics* **1990**, *27*, 269–286.
9. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems* **2014**, pp. 2672–2680.
10. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**.
11. Satyadhar Joshi. Advancing Financial Risk Modeling: Vasicek Framework Enhanced by Agentic Generative AI by Satyadhar Joshi. *Advancing Financial Risk Modeling: Vasicek Framework Enhanced by Agentic Generative AI by Satyadhar Joshi* **2025**, *Volume 7*. https://doi.org/10.56726/IRJMETS66819.
12. Joshi, S. Advancing Innovation in Financial Stability: A Comprehensive Review of Ai Agent Frameworks, Challenges and Applications. *World Journal of Advanced Engineering Technology and Sciences* **2025**, *14*, 117–126. https://doi.org/10.30574/wjaets.2025.14.2.0071.
13. Satyadhar Joshi. Enhancing Structured Finance Risk Models (Leland-Toft and Box-Cox) Using GenAI (VAEs GANs). *International Journal of Science and Research Archive* **2025**, *14*, 1618–1630. https://doi.org/10.30574/ijsra.2025.14.1.0306.

 doi:10.20944/preprints202502.2308.v2

14. Joshi, S. Implementing Gen AI for Increasing Robustness of US Financial and Regulatory System. *International Journal of Innovative Research in Engineering and Management* **2025**, *11*, 175–179. https://doi.org/10.55524/ijirem.2024.11.6.19.
15. Satyadhar Joshi. Review of Gen AI Models for Financial Risk Management. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **2025**, *11*, 709–723. https://doi.org/10.32628/CSEIT2511114.
16. Satyadhar Joshi. The Synergy of Generative AI and Big Data for Financial Risk: Review of Recent Developments. *IJFMR - International Journal For Multidisciplinary Research*, *7*. https://doi.org/g82gmx.
17. Satyadhar Joshi. Leveraging Prompt Engineering to Enhance Financial Market Integrity and Risk Management. *World Journal of Advanced Research and Reviews* **2025**, *25*, 1775–1785. https://doi.org/10.30574/wjarr.2025.25.1.0279.