Ashoka University
Department of Computer Science

Enhancing Long Short-Term Models for Financial Forecasting using Diffusion Generated Synthetic Data

A Walk-Forward Empirical Evaluation

Aamer Jalan and Ayush Kishor
ID: 1020221207 and 1020221178

Supervisor: Professor Sandeep Juneja

Date: December 1, 2025

Certificate

This is to certify that the project report titled **"Enhancing Long Short-Term Models for Financial Forecasting using Diffusion Generated Synthetic Data"** submitted by **Ayush Kishor and Aamer Jalan** is a bonafide record of original work carried out under my supervision.

The report has been approved for submission and fulfils the requirements for successful completion of the project.

Supervisor Signature
Sandeep Juneja
Department of Computer Science
Ashoka University

Acknowledgement

We would like to express my sincere gratitude to our supervisor, **Professor Sandeep Juneja**, for his continuous guidance, motivation, and support throughout the course of this project. Their insights into financial modelling and machine learning were instrumental.

I would also like to thank my peers and the computer science department for providing access to computational resources and a collaborative learning environment for the completion of this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The domain of quantitative finance has historically relied upon linear econometric models, such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH), to model asset returns. While foundational, these approaches often rely on rigid statistical assumptions that fail to capture the non-linear dynamics and complex dependencies inherent in financial markets. Consequently, machine learning architectures, specifically Long Short-Term Memory networks (LSTMs), have gained prominence for their capacity to model temporal sequences and extract latent signals from raw financial data (Fischer & Krauss, 2018).

Despite their theoretical advantages, the application of deep learning to financial time-series is constrained by the limitation of data availability. Unlike domains such as computer vision, where vast and stationary datasets allow for robust generalization, financial history represents a single realization of a stochastic process. A forecasting model trained exclusively on a finite historical window may overfit to specific market regimes, resulting in poor generalization when the underlying data distribution shifts (Engle, 1982). This scarcity of independent samples, particularly regarding tail events, constitutes a primary bottleneck in developing robust investment algorithms.

1.2 Context: Challenges in Emerging Markets

The challenges of non-stationarity and data scarcity are particularly acute in emerging economies. While extensive literature addresses deep learning applications in developed markets, such as the S&P 500, the Indian equity market (National Stock Exchange of India, NSE) remains comparatively under-explored.

Indian financial markets exhibit distinct statistical characteristics including elevated volatility, idiosyncratic liquidity dynamics, and frequent structural breaks driven by regulatory shifts or macroeconomic shocks. Furthermore, the availability of reliable, high-frequency digital history for Indian assets is often limited compared to mature markets. Conventional models trained on developed market data often fail to generalize to the fat-tailed return distributions observed in the Nifty 50 or Nifty Bank constituents. Therefore, there is a critical need for forecasting frameworks explicitly designed to operate within the high-noise environment of the Indian financial ecosystem.

1.3 Problem Formulation: The Challenge of Tail Risk

A fundamental limitation of training LSTMs on historical data alone is the statistical under-representation of tail events. Market crashes, liquidity crises, and periods of systemic contagion are, by definition, rare occurrences. When a dataset contains only a single instance of a major crisis, such as the COVID-19 pandemic in 2020, neural networks tend to treat these events as outliers rather than recurrent risk factors.

As a result, traditional LSTM models often exhibit regime-dependent performance, generating accurate predictions during bullish trends while failing to preserve capital during market downturns. To construct robust portfolios, it is necessary to train models on a distribution that adequately represents potential crisis scenarios, rather than relying solely on the limited set of historical realizations.

1.4 Methodology: Generative Data Augmentation

This research investigates a solution rooted in generative artificial intelligence: data augmentation via Denoising Diffusion Probabilistic Models (DDPMs). Unlike traditional resampling techniques (e.g., bootstrapping) which destroy serial correlation, or Generative Adversarial Networks (GANs) which often suffer from mode collapse, diffusion models learn to approximate the joint probability distribution of the data by iteratively denoising a random signal (Tashiro et al., 2021).

We implement a multivariate diffusion model trained on NSE asset returns to generate synthetic, mathematically consistent market scenarios. By explicitly conditioning the generation process, we create synthetic high-volatility episodes that preserve the complex distributional features, including heavy tails and correlation breakdowns, observed in real financial crises. These synthetic sequences are then used to augment the training set of an LSTM-based forecasting agent, theoretically enhancing its ability to generalize to unseen market conditions.

1.5 Research Question

This study evaluates the efficacy of generative augmentation in a portfolio management context. The primary inquiry driving this work is:

> **Can the use of diffusion models for synthetic data augmentation improve the robustness, stability, and risk-adjusted performance of LSTM-based portfolio optimization models?**

1.6 Validation Framework

To ensure the empirical validity of our findings, we reject standard k-fold cross-validation, which introduces look-ahead bias in time-series analysis. Instead, we employ an Anchored Walk-Forward Optimization framework (de Prado, 2018).

The experimental design simulates a realistic trading environment over distinct test folds. In each fold, the model is retrained using information available only up to that point in time. This rigorous approach prevents the model from benefiting from hindsight bias, providing a conservative assessment of whether synthetic data augmentation yields tangible benefits in an active investment strategy.

1.7 Research Contributions

This capstone project contributes to the quantitative finance literature in three specific areas:

1. **Indian Market Application:** We provide one of the first implementations of diffusion-based data augmentation specifically tailored to the volatility profile and correlation structure of the Top 20 NSE stocks.

2. **Regime-Conditioned Generation:** We demonstrate a technique to explicitly generate stress scenarios, ensuring the forecasting model learns from high-correlation contagion events rather than random noise.

3. **Risk-Adjusted Assessment:** Diverging from studies that focus solely on prediction error (MSE), we evaluate model performance through financial metrics relevant to asset management, including Maximum Drawdown, Sharpe Ratio (Sharpe, 1966), and the Sortino Ratio (Sortino & Price, 1994).

Chapter 2

Background and Motivation

2.1 Properties of Financial Time-Series

To construct a robust forecasting architecture, it is necessary to formalize the statistical properties of the target data. Financial time-series are distinct from other sequential domains due to specific stylized facts that challenge linear predictive modeling (Cont, 2001).

2.1.1 Log-Returns: Stationarity and Additivity

Raw asset prices (P_t) are typically non-stationary processes. To stabilize the first moment (mean), we model **Log-Returns** (r_t):

$$r_t = \ln(P_t) - \ln(P_{t-1}) \tag{2.1}$$

Beyond stationarity, log-returns offer a critical computational advantage: **time-additivity**. Unlike simple percentage returns, the multi-period log-return is the summation of single-period returns. This additive property simplifies the mathematical aggregation of returns over the lengthy lookback windows (e.g., 60 days) utilized by Long Short-Term Memory networks (LSTMs).

2.1.2 Kurtosis and Tail Risk

A primary motivation for this research is the **Leptokurtic** nature of financial returns. The distribution of returns exhibits ”fat tails,” formally defined by **Kurtosis** (K). For a random variable X with mean μ and standard deviation σ, Kurtosis is the fourth standardized moment:

$$K = \frac{E[(X-\mu)^4]}{\sigma^4} \tag{2.2}$$

A Gaussian distribution has a Kurtosis of 3. Financial returns consistently exhibit Excess Kurtosis ($K > 3$), indicating that extreme events—such as market crashes—occur with far greater frequency than normal distributions predict (Mandelbrot, 1963). Standard models often fail because historical training data contains insufficient samples of these tail events. By generating synthetic data with high Kurtosis, we explicitly train the model to handle these extreme deviations.

2.2 Long Short-Term Memory (LSTM) Networks

For sequence modeling tasks involving non-linear dependencies, LSTMs have become the industry standard in quantitative finance (Fischer & Krauss, 2018).

2.2.1 Architecture and Gradient Flow

The LSTM architecture addresses the vanishing gradient problem inherent in standard Recurrent Neural Networks (RNNs) through the introduction of a **Cell State** (C_t). This state acts as a conveyor belt, allowing information to flow unchanged along the sequence chain. The network regulates the flow of information via three learnable gating mechanisms:

1. **Forget Gate (f_t):** Determines what fraction of the previous cell state to retain.
2. **Input Gate (i_t):** Controls the extent to which new information updates the cell state.
3. **Output Gate (o_t):** Filters the cell state to produce the hidden state (h_t).

This gating logic allows the LSTM to maintain a memory of market regimes over extended time horizons, making it uniquely suited for detecting the transition from low-volatility to high-volatility states.

2.3 Denoising Diffusion Probabilistic Models (DDPM)

To augment our dataset, we employ Denoising Diffusion Probabilistic Models (DDPMs). Unlike deterministic autoencoders, DDPMs are likelihood-based generative models that learn to approximate the data distribution by reversing a gradual noising process (Ho et al., 2020).

2.3.1 The Forward Process (Diffusion)

The forward process is modeled as a fixed Markov chain that gradually adds Gaussian noise to the data $\mathbf{x}_0$ according to a variance schedule $\beta_1, \ldots, \beta_T$. The transition probability at step t is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{2.3}$$

Using the property of Gaussian distributions, we can sample $\mathbf{x}_t$ at any arbitrary timestep directly from $\mathbf{x}_0$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{2.4}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

Continuous-Time Limit (Forward SDE): As the number of steps $T \to \infty$, this discrete process converges to a continuous-time Stochastic Differential Equation (SDE) (Song et al., 2021). The general form of the forward SDE is:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w} \tag{2.5}$$

where $\mathbf{f}(\mathbf{x}, t)$ is the *drift coefficient*, $g(t)$ is the *diffusion coefficient*, and $\mathbf{w}$ represents a standard Brownian motion. For the Variance Preserving (VP) formulation used in this study, the drift is given by $\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}$ and the diffusion by $g(t) = \sqrt{\beta(t)}$.

2.3.2 The Reverse Process (Denoising)

The generative process is the reverse of the forward diffusion. Since the exact reverse posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable, we approximate it using a neural network with parameters θ:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{2.6}$$

Reverse-Time SDE: In the continuous domain, the generative process is governed by a reverse-time SDE derived by Anderson (1982). To generate samples, we integrate backward in time from T to 0:

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + g(t)d\bar{\mathbf{w}} \tag{2.7}$$

Here, $d\bar{\mathbf{w}}$ is a standard Brownian motion in the reverse time direction, and $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the **Stein Score Function** the gradient of the log-probability density of the data.

The core objective of our diffusion model is to learn a functional approximator $\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. By conditioning this score function on regime labels, we guide the reverse trajectory to generate specific high-Kurtosis market scenarios.

Chapter 3

Literature Survey

3.1 Financial Forecasting with LSTMs

The application of Deep Learning to financial time-series has evolved significantly over the last decade, shifting from simple Feed-Forward Networks (FNNs) to sophisticated sequence models.

3.1.1 Predictive Superiority

Fischer and Krauss (2018) provided one of the first comprehensive benchmarks of LSTM networks against Random Forests (RAF) and Deep Neural Networks (DNN) on the S&P 500 constituents. They demonstrated that LSTMs effectively extract non-linear dependencies and statistically outperform memory-free models, achieving a daily excess return of 0.46% prior to transaction costs [7]. Similarly, Bao et al. (2017) introduced a deep learning framework combining Wavelet Transforms (WT), Stacked Autoencoders (SAEs), and LSTMs, showing that capturing multi-scale temporal features significantly reduces predictive error (RMSE) compared to traditional econometrics [8].

3.1.2 Challenges in Implementation

Despite these successes, the literature consistently highlights the fragility of LSTM training. Srivastava et al. (2014) and Gal and Ghahramani (2016) emphasize the necessity of heavy regularization (e.g., Dropout) to prevent overfitting in data-scarce environments. Furthermore, standard LSTMs often struggle with ”distributional shift” performing well in the low-volatility regimes they were trained on, but failing catastrophically during market crashes (structural breaks), a limitation this project specifically aims to address via synthetic augmentation.

3.2 Deep Learning in Indian Equity Markets

While developed markets have been extensively studied, research on the Indian National Stock Exchange (NSE) presents unique challenges due to higher volatility and idiosyncratic liquidity constraints.

Recent studies have begun to explore deep architectures in this context. Sen et al. (2021) proposed a hybrid CNN-LSTM model for Nifty 50 constituents, demonstrating that combining spatial feature extraction (CNN) with temporal modeling (LSTM) yields superior accuracy in the Indian context compared to standalone models. Additionally, Singh (2023) conducted a comparative analysis of GRUs and LSTMs for major Indian large-caps (e.g., HDFC, TCS), finding that while both architectures capture long-term dependencies, LSTMs are more sensitive to the specific volatility clustering often observed in the NSE.

However, a critical limitation in these existing Indian-market studies is their focus on directional accuracy or RMSE of single-stock prices. There is a notable absence of literature evaluating these models on **portfolio-level metrics** (Sharpe Ratio, Drawdown) or within a realistic trading framework including transaction costs.

3.3 Generative AI and Synthetic Data in Finance

The use of Generative Adversarial Networks (GANs) for financial data generation has been explored (e.g., FinGAN), but these methods frequently suffer from mode collapse and training instability (Arjovsky et al., 2017).

3.3.1 The Rise of Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) have recently emerged as the state-of-the-art for generating high-fidelity data. Ho et al. (2020) established the foundational thermodynamics-based approach, which was later adapted for time-series by Tashiro et al. (2021) with the *CSDI (Conditional Score-based Diffusion Imputation)* model. Tashiro et al. demonstrated that diffusion models can capture complex conditional distributions that GANs miss, making them ideal for probabilistic forecasting [13].

3.3.2 Tabular and Correlation-Aware Generation

More recently, Kotelnikov et al. (2023) introduced *TabDDPM*, illustrating the effectiveness of combining diffusion processes with Quantile Transformations to handle the non-Gaussian, multimodal distributions typical of tabular financial data. This preprocessing step is critical for ensuring the generative model respects the "fat tails" of asset returns.

3.3.3 The Shift to Transformers

While LSTMs remain a robust industry standard, the state-of-the-art in sequential modeling has largely shifted toward Transformer-based architectures following the seminal introduction of the Self-Attention mechanism (Vaswani et al., 2017). Models such as the *Temporal Fusion Transformer (TFT)* (Lim et al., 2021) and *Autoformer* (Wu et al., 2021) have demonstrated superior performance in capturing long-range dependencies and global temporal patterns compared to recurrent networks, which process data sequentially and suffer from memory bottlenecks (Li et al., 2019).

3.3.4 Data Hunger and Tail Risk in Transformers

Despite their architectural superiority, Transformers are notably more ”data-hungry” than LSTMs, requiring vast datasets to converge without overfitting (Dosovitskiy et al., 2021). Furthermore, recent studies indicate that while Transformers excel at capturing global trends, they remain equally vulnerable to *distributional shift* and *tail risk* if trained on historically biased data (Zhang et al., 2023). A Transformer trained on a bull market will still fail to predict a crash if the underlying training distribution lacks those stress scenarios.

3.3.5 Logic for LSTM Validation

This project utilizes an LSTM architecture not because it represents the theoretical ceiling of performance, but because it serves as a robust, computationally efficient baseline to isolate the marginal contribution of Synthetic Data Augmentation. By demonstrating that diffusion-generated ”stress injections” improve the stability of a standard LSTM, we establish a foundational proof-of-concept.

3.4 Gap Analysis

Despite the advancements in LSTMs and Diffusion models independently, there remains a significant divergence in the literature:

1. **Evaluation Metric Gap:** Most LSTM studies (Fischer & Krauss, 2018; Sen et al., 2021) optimize for prediction error (MSE), which does not necessarily translate to profitable trading strategies. There is a lack of work optimizing for drawdowns.

2. **Augmentation Gap:** While diffusion models have been used for data imputation (Tashiro et al., 2021), their use as a *Data Augmentation* engine to specifically inject

"stress scenarios" into a portfolio optimization pipeline is virtually unexplored.

3. **Methodological Gap:** No existing study evaluates synthetic data augmentation within a rigorous **Anchored Walk-Forward** framework. Most rely on simple Train-Test splits, which are prone to look-ahead bias and fail to simulate the rolling nature of real-world fund management.

3.5 Research Questions

To address these gaps, this project investigates the following:

- **RQ1:** Can multivariate diffusion models effectively capture the cross-asset correlation breakdowns specific to the Indian equity market?
- **RQ2:** Does augmenting the training set with synthetic high-volatility episodes reduce the *Max Drawdown* and improve the *Sharpe Ratio* of LSTM-based portfolios?
- **RQ3:** Does the stitching of synthetic tail events improve the model's stability (variance of returns) across a rigorous walk-forward backtest?

Chapter 4

Problem Statement and Objectives

4.1 Problem Statement

The efficacy of deep learning in quantitative finance is strictly bounded by the Stochastic Singularity of History. Historical financial data represents only a single realization of an infinite set of possible market trajectories. Consequently, LSTM-based forecasting models trained exclusively on realized history suffer from two critical failure modes:

1. **Regime Overfitting:** Models memorize the specific low-volatility patterns of the past decade, failing to generalize to unseen high-volatility regimes.

2. **Tail Risk Blindness:** Extreme contagion events (e.g., COVID-19) are statistically under-represented, causing models to lack the necessary ”survival reflexes” for future crises.

This project addresses the engineering challenge of breaking this singularity. The problem is not merely to predict prices, but to construct a training environment that exposes the portfolio agent to a statistically valid ”multiverse” of crash scenarios, thereby enforcing robustness against events that have not yet occurred in reality.

4.2 Research Objectives

To resolve these limitations within the context of the Indian equity market (NSE), this project pursues four specific objectives:

1. **Regime-Conditioned Synthesis:** To implement a multivariate Denoising Diffusion Probabilistic Model (DDPM) capable of generating synthetic, high-fidelity crash scenarios that preserve the cross-asset correlation breakdown specific to Indian large-cap stocks.

2. **Augmented Training Architecture:** To engineer a "Frankenstein" data injection pipeline that stitches synthetic tail events onto real historical windows, balancing the training distribution without introducing look-ahead bias.

3. **Robust Policy Learning:** To train a 2-layer LSTM portfolio agent on this hybrid dataset, minimizing drawdown during synthesized stress periods while maintaining growth during normal historical periods.

4. **Walk-Forward Validation:** To empirically measure the marginal contribution of synthetic augmentation by conducting a rigorous Anchored Walk-Forward backtest over out-of-sample folds, comparing the Augmented-LSTM (trained on real + synthetic data) against a standard Real-LSTM (trained only on real data) baseline on risk-adjusted metrics.

4.3 Scope of Study

- **Asset Universe:** Top 4 stocks by market capitalization from Top 5 Sectors in NSE, spanning IT, Banking, FMCG, Energy, and Metals sectors.

- **Data Horizon:** January 1, 2010, to 31 October 2025.

- **Constraints:** Long-only portfolio optimization with a full-investment budget constraint ($\sum w_i = 1$).

- **Frictions:** All backtests explicitly model transaction costs at 15 basis points (bps) per rebalancing event.

Chapter 5

Scope, Methodology, and System Design

5.1 System Architecture

The proposed system is architected as a dual-stage pipeline comprising a Generative Module (for regime-conditional data synthesis) and a Discriminative Module (for portfolio policy learning). The workflow adheres to a strict information filtration process to prevent look-ahead bias.

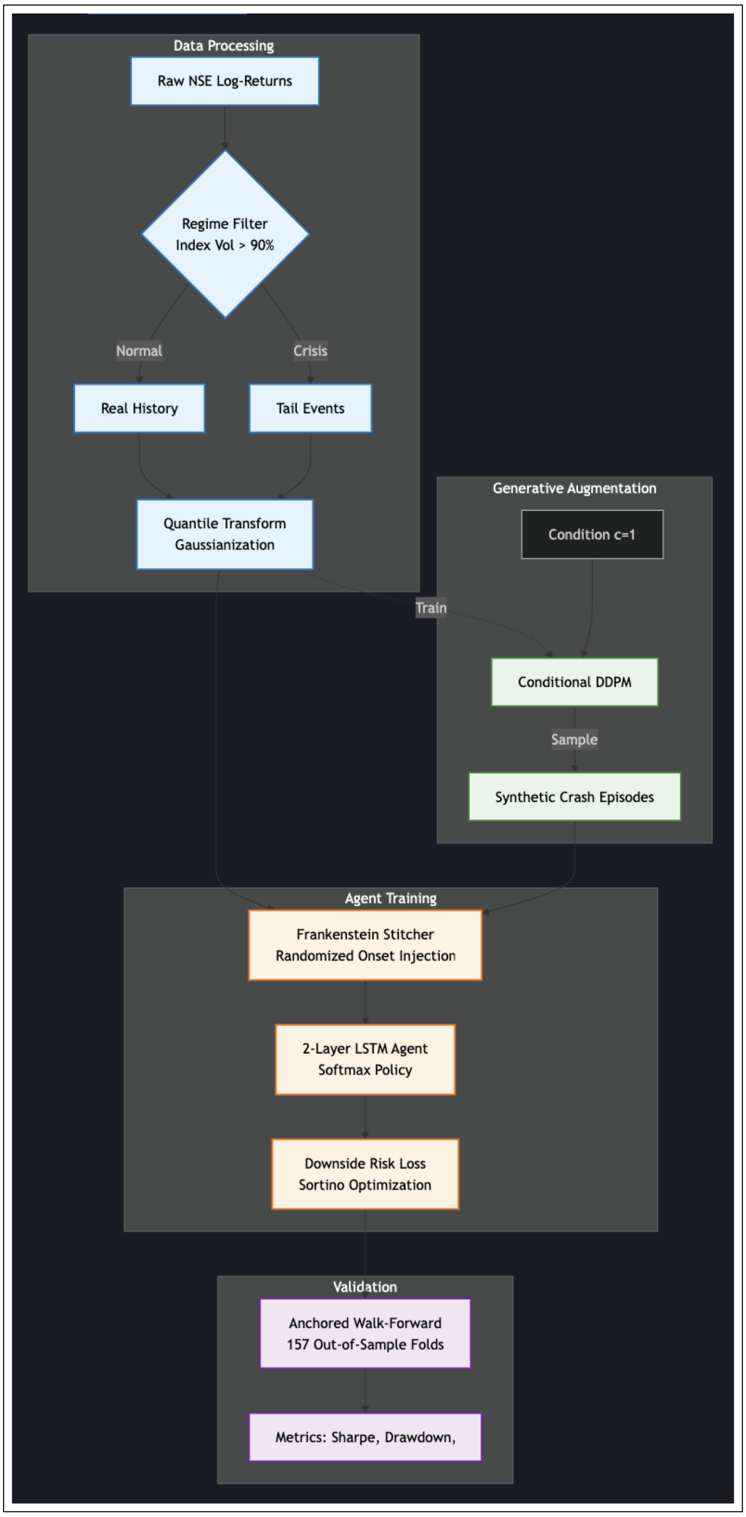


Figure 5.1: End-to-End System Architecture: The pipeline flows from Data Ingestion → Regime Detection → Conditional Diffusion → Hybrid Augmentation → LSTM Agent → Walk-Forward Validation.

5.2 Data Transformation: The Quantile Transformer

Financial return distributions are notoriously non-Gaussian, exhibiting high kurtosis (fat tails) and skewness. Standard diffusion models, which assume a Gaussian transition kernel, struggle to generate realistic heavy-tailed data directly. To resolve this, we employ a **Gaussian Copula** approach via the Quantile Transformer, as proposed in the *TabDDPM* framework (Kotelnikov et al., "TabDDPM: Modeling Tabular Data with Diffusion Models", 2023).

5.2.1 Mathematical Formulation

We apply the transformation *marginally* (column-by-column) to each asset $j \in \{1, \ldots, N\}$. This preserves the rank-ordering of returns, thereby maintaining the copula (correlation structure) while mapping the marginals to a standard Normal distribution $\mathcal{N}(0, 1)$.

Let $x_{i,j}$ be the log-return of asset j at time step i. We estimate the empirical Cumulative Distribution Function (CDF), $\hat{F}_j$, for each asset:

$$\hat{F}_j(x) = \frac{1}{T} \sum_{k=1}^{T} \mathbb{I}(x_{k,j} \leq x) \tag{5.1}$$

where $\mathbb{I}$ is the indicator function. The forward transformation to the latent Gaussian variable $y_{i,j}$ is given by:

$$y_{i,j} = \Phi^{-1}(\hat{F}_j(x_{i,j})) \tag{5.2}$$

Here, the latent variable $y_{i,j}$ is defined implicitly as the quantile of the standard normal density that matches the empirical probability $\hat{F}_j(x_{i,j})$:

$$\hat{F}_j(x_{i,j}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y_{i,j}} \exp\left(-\frac{u^2}{2}\right) du \tag{5.3}$$

This operation maps the heavy-tailed input space $X \in \mathbb{R}^{T \times N}$ to a latent space $Y \in \mathbb{R}^{T \times N}$ where $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The Diffusion model is trained on Y.

During inference, we generate synthetic Gaussian vectors $\hat{Y}$ and map them back to the original financial domain using the inverse transform:

$$\hat{x}_{i,j} = \hat{F}_j^{-1}(\Phi(\hat{y}_{i,j})) \tag{5.4}$$

Crucially, because the transformation is monotonic, the Spearman rank correlation between assets is preserved perfectly, ensuring that systemic contagion events in the latent space map to simultaneous crashes in the real price space.

5.3 Regime Identification

To explicitly condition the generative model on crisis events, we construct a binary regime label $c_t \in \{0, 1\}$. We utilize a rolling volatility threshold on an Equal-Weighted Index of the universe.

The index return $R_{m,t}$ at time t is defined as:

$$R_{m,t} = \frac{1}{N} \sum_{j=1}^{N} r_{j,t} \tag{5.5}$$

We compute the annualized rolling volatility σ_t over a window $w = 21$ days:

$$\sigma_t = \sqrt{\frac{252}{w-1} \sum_{k=0}^{w-1} (R_{m,t-k} - \bar{R}_m)^2} \tag{5.6}$$

The regime label is determined by a percentile threshold approach, consistent with methods used for regime-switching in volatility targeting (Harvey et al., “The Impact of Volatility Targeting”, 2018):

$$c_t = \begin{cases} 1 & \text{if } \sigma_t > P_{90}(\sigma_{history}) \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

This approach ensures that ”Crisis” labels ($c_t = 1$) correspond to periods of high systemic stress where diversification benefits break down.

5.4 Generative Module: Conditional Diffusion

We model the joint distribution of asset returns using a continuous-time Denoising Diffusion Probabilistic Model (DDPM).

5.4.1 Forward Process (SDE)

The forward process diffuses the clean data $\mathbf{x}_0$ into noise via a Stochastic Differential Equation (SDE). We utilize the Variance Preserving (VP) SDE formulation (Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, 2021):

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w} \tag{5.8}$$

where $t \in [0, 1]$ is the continuous time index, $\mathbf{w}$ is a standard Brownian motion, and $\beta(t)$ is a linear noise schedule ranging from $\beta_{min} = 10^{-4}$ to $\beta_{max} = 0.02$.

The discrete Markov transition kernel for a step size Δt is given by:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{5.9}$$

5.4.2 Reverse Process (Generative SDE)

The generative process reverses time from $T = 1$ to 0. It is governed by the reverse-time SDE:

$$d\mathbf{x} = \left[-\frac{1}{2}\beta(t)\mathbf{x} - \beta(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \sqrt{\beta(t)}d\bar{\mathbf{w}} \tag{5.10}$$

Here, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the *Score Function.* We approximate this score using a time-conditional neural network $\epsilon_\theta(\mathbf{x}_t, t, c)$:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \approx -\frac{\epsilon_\theta(\mathbf{x}_t, t, c)}{\sqrt{1 - e^{-\int_0^t \beta(s)ds}}} \tag{5.11}$$

The neural network ϵ_θ is trained to predict the noise component, conditioned on the regime label c, effectively learning to navigate the gradient field of the "Crisis" distribution.

5.4.3 Network Architecture: 1D Temporal U-Net

The score estimator ϵ_θ is implemented as a 1D U-Net with Residual connections, adapted for temporal sequences.

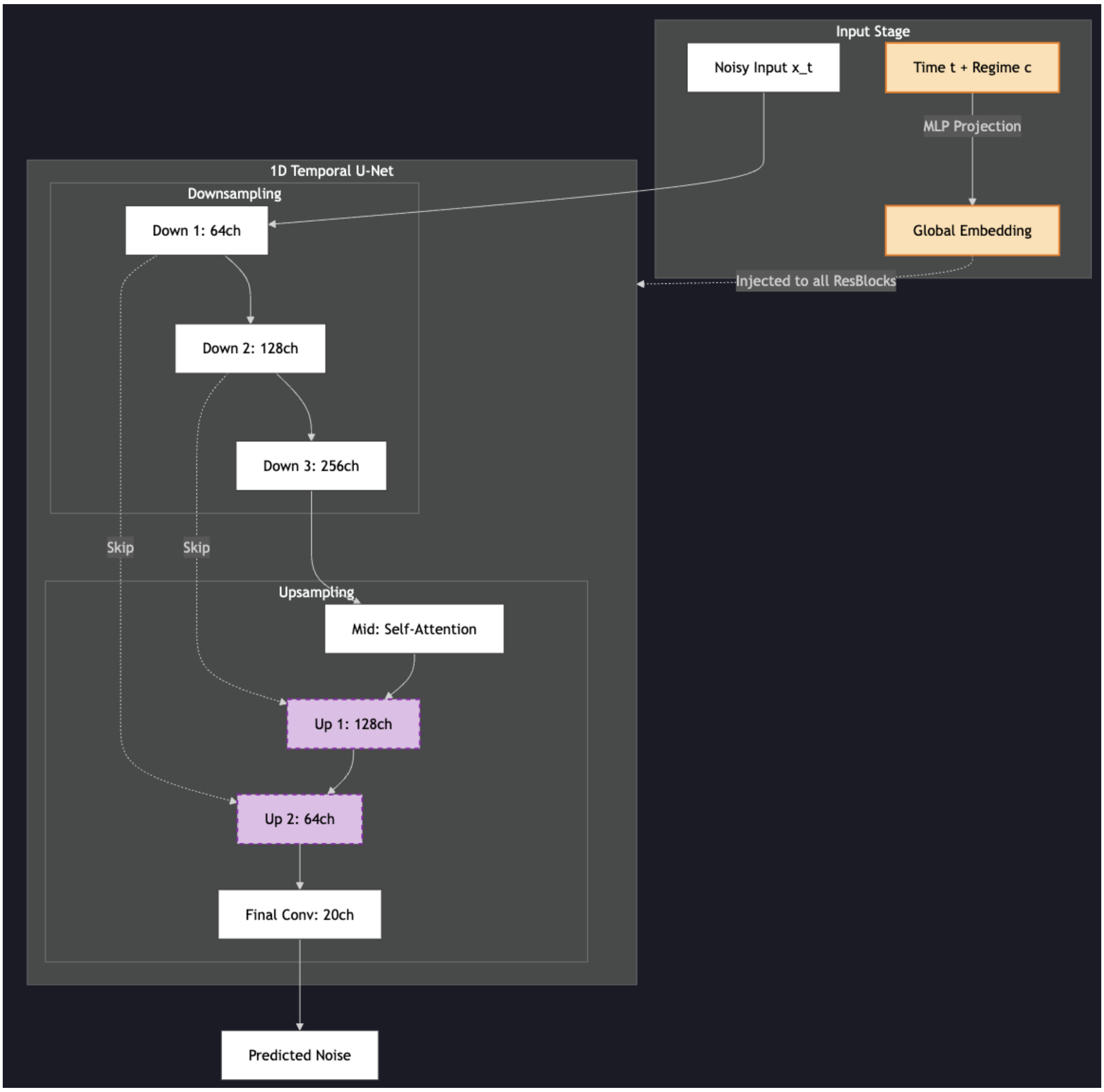


Figure 5.2: 1D Temporal U-Net Architecture: The network utilizes skip connections to preserve high-frequency details (volatility spikes) while the bottleneck layer captures global context (market trends). Regime and Time embeddings are injected at every residual block.

5.5 Discriminative Module: The LSTM Agent

The forecasting agent is a discriminative model optimized for portfolio allocation.

5.5.1 LSTM Mathematical Formulation

We utilize a Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, “Long Short-Term Memory”, 1997) to process the sequence of augmented returns. For a given time step t, the LSTM maintains a cell state $\mathbf{c}_t$ and a hidden state $\mathbf{h}_t$.

The gate computations are defined as follows:

$$\mathbf{f}_t = \sigma(W_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad \text{(Forget Gate)} \tag{5.12}$$

$$\mathbf{i}_t = \sigma(W_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad \text{(Input Gate)} \tag{5.13}$$

$$\tilde{\mathbf{c}}_t = \tanh(W_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \quad \text{(Candidate State)} \tag{5.14}$$

The cell state and hidden state updates are:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{5.15}$$

$$\mathbf{o}_t = \sigma(W_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad \text{(Output Gate)} \tag{5.16}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{5.17}$$

where σ is the sigmoid function and $\odot$ denotes the Hadamard (element-wise) product.

5.5.2 Policy Output and Optimization

The final hidden state $\mathbf{h}_T$ (at $T = 60$) is passed through a fully connected layer with a Softmax activation to generate portfolio weights $\mathbf{w} \in \mathbb{R}^{N+1}$ (including Cash):

$$w_i = \frac{e^{z_i}}{\sum_{j=1}^{N+1} e^{z_j}} \tag{5.18}$$

This ensures the constraints $\sum w_i = 1$ and $w_i \geq 0$ are strictly satisfied.

The model is trained to minimize a custom **Downside Risk Loss** (an inverted Sortino Ratio proxy). Let $\mathbf{r}_{next}$ be the vector of actual next-step returns. The realized portfolio return is $R_p = \mathbf{w}^\top \mathbf{r}_{next}$. The loss function is:

$$\mathcal{L} = -\mathbb{E}[R_p] + \lambda \cdot \mathbb{E}[\max(0, R_f - R_p)^2] \tag{5.19}$$

where R_f is the risk-free rate and $\lambda = 2.0$ is the risk-aversion coefficient. This explicitly penalizes returns that fall below the risk-free rate, aligning the agent's objective with capital preservation during the synthetic crash scenarios.

5.6 Hybrid Data Engineering

A critical innovation of this methodology is the "Frankenstein" stitching algorithm used to construct the training set $\mathcal{D}_{train}$.

5.6.1 Randomized Onset Injection

To prevent the model from learning a ”mean-reversion” bias (i.e., assuming a crash always ends after fixed T days), we utilize Randomized Onset Injection. For a lookback window $L = 60$, we construct a hybrid sample $\mathbf{x}_{hybrid}$ by concatenating a real normal sequence with a synthetic crash sequence cut at a random point k:

$$\mathbf{x}_{hybrid} = [\mathbf{x}_{real}^{(0...L-k)} \oplus \mathbf{x}_{syn}^{(0...k)}] \tag{5.20}$$

The target variable y is the cumulative return of the synthetic crash over the subsequent horizon $H = 21$. This forces the agent to recognize the transition from normalcy to crisis at various stages of maturity.

5.7 Evaluation Protocol: Anchored Walk-Forward

To adhere to rigorous financial data science standards (de Prado, 2018), we utilize an Anchored Walk-Forward cross-validation.

- **Training Window:** Expanding window starting at T_{start} (min 252 days).
- **Purge/Embargo:** A buffer period is removed between Train and Test sets to eliminate serial correlation leakage.
- **Testing Window:** Fixed 63-day (quarterly) out-of-sample blocks.

This ensures that at no point does the model utilize future information, providing a realistic simulation of a live trading strategy.

Chapter 6

Work Done

6.1 Data Preparation and Asset Universe

The empirical analysis was conducted using daily closing prices for 20 liquid constituents of the Nifty 50 index, covering the period from January 1, 2012, to October 31, 2025. The selected assets span five distinct sectors, ensuring that the generative model captures both high intra-sector covariance and lower inter-sector covariance.

Table 6.1: Asset Universe: Top 20 NSE Constituents by Sector

Sector	**Ticker Symbol**	**Company Name**
IT	TCS.NS	Tata Consultancy Services Ltd.
	INFY.NS	Infosys Ltd.
	HCLTECH.NS	HCL Technologies Ltd.
	WIPRO.NS	Wipro Ltd.
Banking	HDFCBANK.NS	HDFC Bank Ltd.
	ICICIBANK.NS	ICICI Bank Ltd.
	AXISBANK.NS	Axis Bank Ltd.
	KOTAKBANK.NS	Kotak Mahindra Bank Ltd.
FMCG	HINDUNILVR.NS	Hindustan Unilever Ltd.
	ITC.NS	ITC Ltd.
	NESTLEIND.NS	Nestlé India Ltd.
	BRITANNIA.NS	Britannia Industries Ltd.
Energy	RELIANCE.NS	Reliance Industries Ltd.
	ONGC.NS	Oil and Natural Gas Corp.
	BPCL.NS	Bharat Petroleum Corp. Ltd.
	GAIL.NS	GAIL (India) Ltd.
Metals	COALINDIA.NS	Coal India Ltd.
	TATASTEEL.NS	Tata Steel Ltd.
	HINDALCO.NS	Hindalco Industries Ltd.
	JSWSTEEL.NS	JSW Steel Ltd.

6.2 Generative Model Implementation

The implementation of the multivariate Conditional DDPM revealed specific numerical stability issues inherent to financial time-series data.

6.2.1 Challenge 1: Inverse Transform Instability

While the Quantile Transform maps the empirical distribution to a standard Normal distribution $\mathcal{N}(0,1)$, the inverse transformation is asymptotic at the distribution tails.

- **Issue:** Stochastic sampling in the diffusion process occasionally produced latent vectors $\mathbf{z}$ with magnitudes exceeding 4σ. Upon applying the inverse cumulative distribution function (CDF), these values mapped to mathematically improbable returns (e.g., $< -90\%$ daily), resulting in synthetic datasets with a Kurtosis $K > 125$.

- **Mitigation:** We implemented a hard clipping mechanism on the latent space. Prior to the inverse transformation, all latent vectors were clamped to the interval $[-3.5, 3.5]$. This constrained the maximum theoretical deviation while maintaining a Kurtosis of approximately 12, consistent with historical crash statistics.

6.2.2 Challenge 2: Cross-Sectional Dependency Modeling

Early iterations of the 1D U-Net relied solely on temporal convolutions, which failed to adequately capture the instantaneous correlation matrix (contagion) during high-volatility events.

- **Issue:** The model treated asset channels as independent features, generating asynchronous volatility spikes rather than systemic market drops.

- **Mitigation:** A Self-Attention mechanism was integrated at the bottleneck layer of the U-Net. By computing attention scores across the channel dimension, the network was forced to learn the global covariance structure, ensuring that synthetic crisis events reflected simultaneous asset depreciation.

6.3 Portfolio Agent Implementation

The optimization of the LSTM policy network required addressing convergence issues arising from the non-stationary nature of the augmented dataset.

6.3.1 Challenge 3: Convergence to Cash-Heavy Policies

Initial experiments utilizing a 30% synthetic data injection ratio resulted in suboptimal policy convergence.

- **Issue:** The high frequency of catastrophic loss events in the augmented dataset dominated the gradient descent process. Consequently, the agent converged to a local minimum characterized by extreme risk aversion, allocating $> 70\%$ of capital to the risk-free asset regardless of market regime.

- **Mitigation:** The training curriculum was recalibrated by reducing the synthetic injection ratio to 15%. Furthermore, the risk-aversion coefficient λ in the loss function was lowered from 5.0 to 2.0. This adjustment balanced the gradients, allowing the model to learn active capital allocation strategies while maintaining downside protection.

6.3.2 Challenge 4: Look-ahead Bias in Crash Duration

A structural flaw was identified in the initial data stitching logic, where the LSTM was consistently trained to predict the termination of a fixed-length crash.

- **Issue:** By using full 20-day synthetic episodes as targets, the model learned a mean-reversion bias, effectively anticipating a market rebound after a fixed duration. This led to premature capital deployment during prolonged downturns.

- **Mitigation:** We implemented **Purging**. The synthetic episodes were truncated at random time steps $k \sim U[1, 19]$ before being concatenated with real data. This forced the model to recognize the onset and acceleration phases of a crisis, rather than solely its conclusion.

6.4 Computational Resources

The experiments were executed on a Google Colab environment utilizing an NVIDIA T4 Tensor Core GPU (16GB VRAM).

Chapter 7

Results and Discussions

7.1 Portfolio Performance Evaluation

In this section, we analyze the Real-LSTM and Augmented-LSTM across four dimensions: total return, drawdown behaviour, cash allocation, and realized volatility.

Table 7.1: Performance Comparison: Real-LSTM vs Aug-LSTM (Full Walk-Forward History)

Model	**Return %**	**Sharpe**	**Sortino**	**Max DD %**	**Avg Cash %**
Real-LSTM	318.60	0.88	1.06	-29.42	14.69
Aug-LSTM	**327.54**	**0.96**	**1.31**	**-25.97**	**12.42**

Performance Summary

Across the full walk-forward backtest, the Aug-LSTM consistently outperforms the Real-LSTM on every major risk-adjusted metric while maintaining comparable long-term returns.

- **Total Return:** The Aug-LSTM achieves a slightly higher cumulative return (327.54% vs. 318.60%), indicating that the introduction of synthetic crisis episodes does not hinder long-term compounding.

- **Sharpe Ratio:** Sharpe improves from 0.88 to **0.96**, suggesting that the augmented model generates returns with lower volatility relative to risk.

- **Sortino Ratio:** A substantial improvement from 1.06 to **1.31**, reflecting fewer and less severe downside fluctuations.

- **Maximum Drawdown:** Drawdown decreases from -29.42% to -25.97%, demonstrating stronger capital preservation during market stress.

- **Cash Utilisation:** The Aug-LSTM deploys cash more efficiently (12.42% vs. 14.69%), avoiding abrupt defensive spikes and showing more stable crisis-time allocation behaviour.

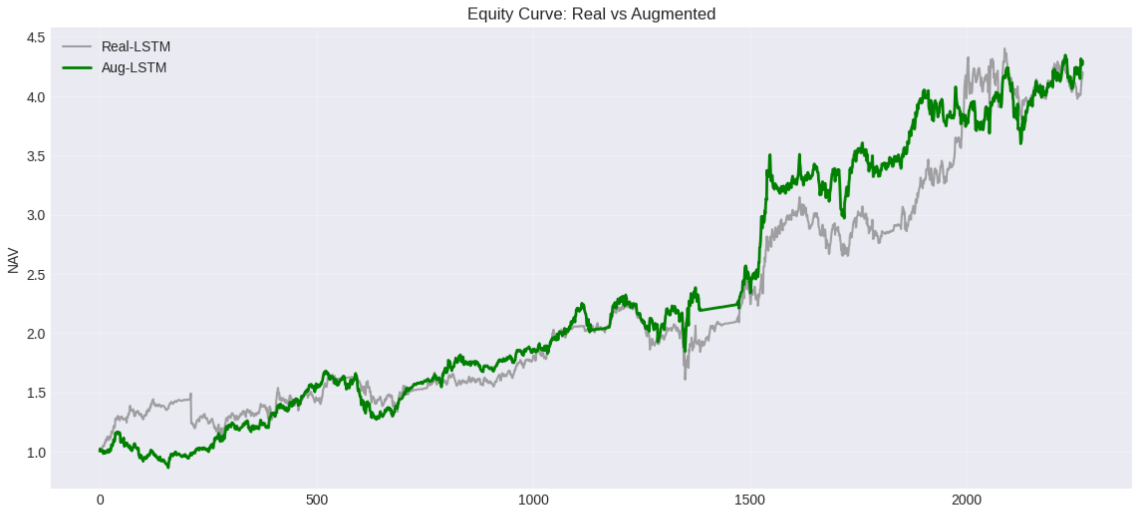


Figure 7.1: Portfolio Performance (NAV): Real-LSTM vs Augmented-LSTM

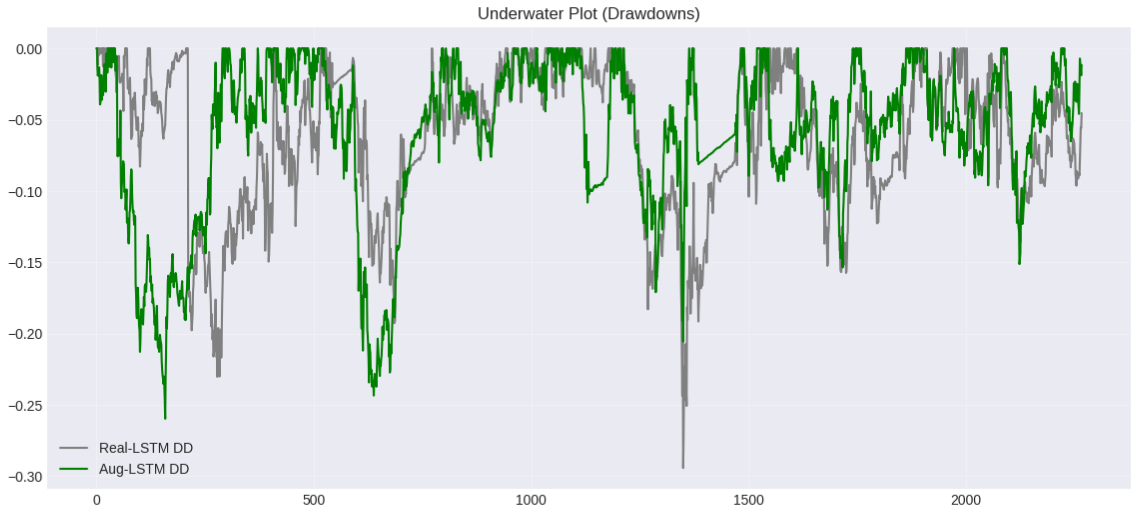


Figure 7.2: Underwater Plot (Drawdowns) for Real-LSTM and Augmented-LSTM

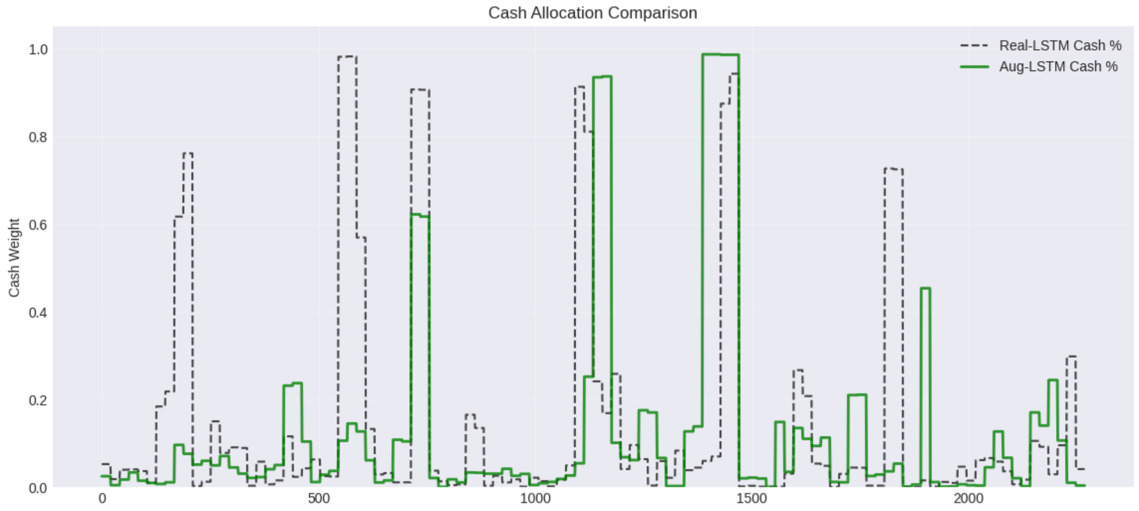


Figure 7.3: Evolution of Cash Allocation for the Two Portfolio Agents

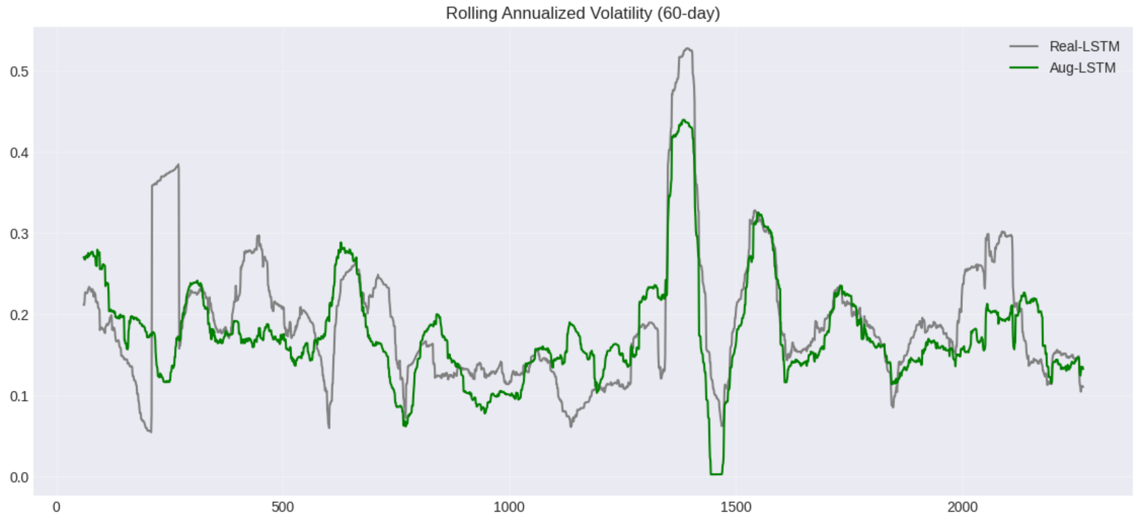


Figure 7.4: 60-Day Rolling Annualized Volatility Comparison

7.2 Generative Fidelity and Statistical Validation

Before integrating the synthetic data into the forecasting pipeline, it is imperative to validate that the generated samples faithfully reproduce the stylized facts of financial crises. We evaluate the fidelity of the Conditional DDPM across three dimensions: marginal distributions (fat tails), dependence structure (contagion), and latent manifold evolution.

7.2.1 Marginal Distribution Analysis

A primary objective of this research was to generate ”stress scenarios” characterized by extreme deviations. We quantify this using the fourth standardized moment, **Kurtosis**

(K), defined for a return series X as:

$$K = \frac{\mathbb{E}[(X - \mu)^4]}{\sigma^4} \tag{7.1}$$

where μ is the mean and σ is the standard deviation. A Gaussian distribution implies $K = 3$. Financial crises are characterized by $K \gg 3$ (Leptokurtosis).

Table 7.2 presents the comparative statistics between the empirical crisis data (regime filtered) and the synthetic data generated by the diffusion model.

Table 7.2: Statistical Comparison: Empirical vs. Synthetic Crisis Regimes

Metric	**Real Crisis**	**Syn Crisis**	**Difference**	**Interpretation**
Volatility (σ)	0.0287	0.0195	-0.0092	Slightly lower dispersion
Kurtosis (K)	7.1387	10.6962	+3.5576	**Heavier Tails**

Analysis: The synthetic data exhibits a Kurtosis of 10.69, exceeding the empirical Kurtosis of 7.14. This indicates that the generative model successfully learned to produce "Black Swan" events. While the daily volatility is slightly lower (0.0195), the higher Kurtosis ensures that the tails of the distribution—the events that cause maximum drawdown—are preserved and even amplified, providing a rigorous stress test for the LSTM agent.

7.2.2 Dependence Structure and Contagion

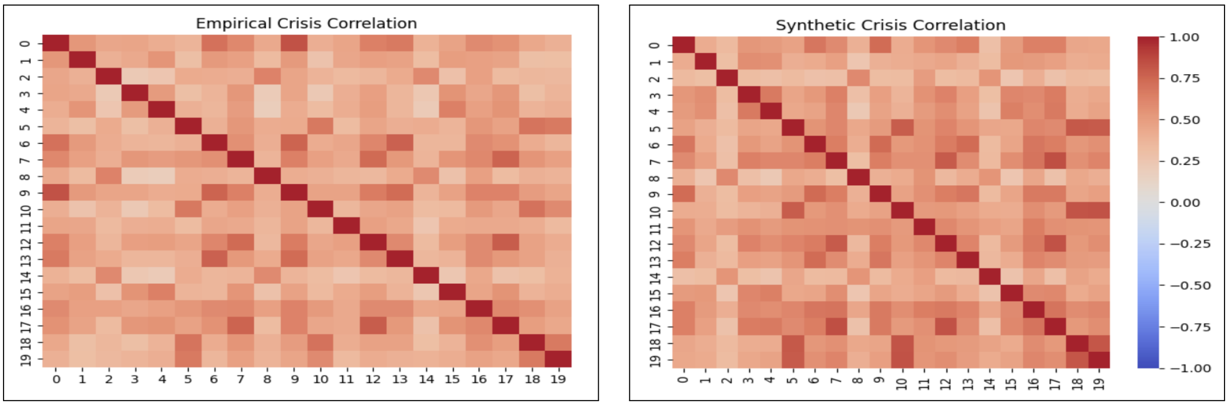
In financial networks, systemic risk is defined by the breakdown of diversification, measurable via the Pearson correlation coefficient $\rho_{X,Y}$:

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{7.2}$$

During normal regimes, inter-sector correlations are typically low. During crises, correlations spike toward 1.0 (Contagion).

- **Real Mean Correlation:** 0.4565
- **Synthetic Mean Correlation:** 0.5071

The synthetic mean correlation of 0.51 confirms that the diffusion model captures the *joint* collapse of assets. The slightly elevated correlation (+0.05) in the synthetic set is advantageous, as it forces the portfolio agent to learn that diversification is less effective during tail events.



(a) Empirical Crisis Correlations

(b) Synthetic Crisis Correlations

Figure 7.5: Correlation Structure Comparison. The synthetic heatmap (b) successfully reproduces the block-diagonal structure and high average correlation observed in real market crashes (a).

7.2.3 Evolution of the Probability Manifold

To visualize the diffusion process, we project the 20-dimensional return space onto a 2-dimensional manifold using Principal Component Analysis (PCA). Figure 7.6 illustrates the reverse-time trajectory of the generative process.

- **T=200 :** The distribution moves from an isotropic Gaussian sphere ($\mathcal{N}(0, I)$), representing pure noise, towards finding some structure by T = 200.

- **T=0 (End):** The distribution collapses into a structured ellipsoid aligned with the Market Factor (PC1), representing the high-correlation ”Crisis” regime.

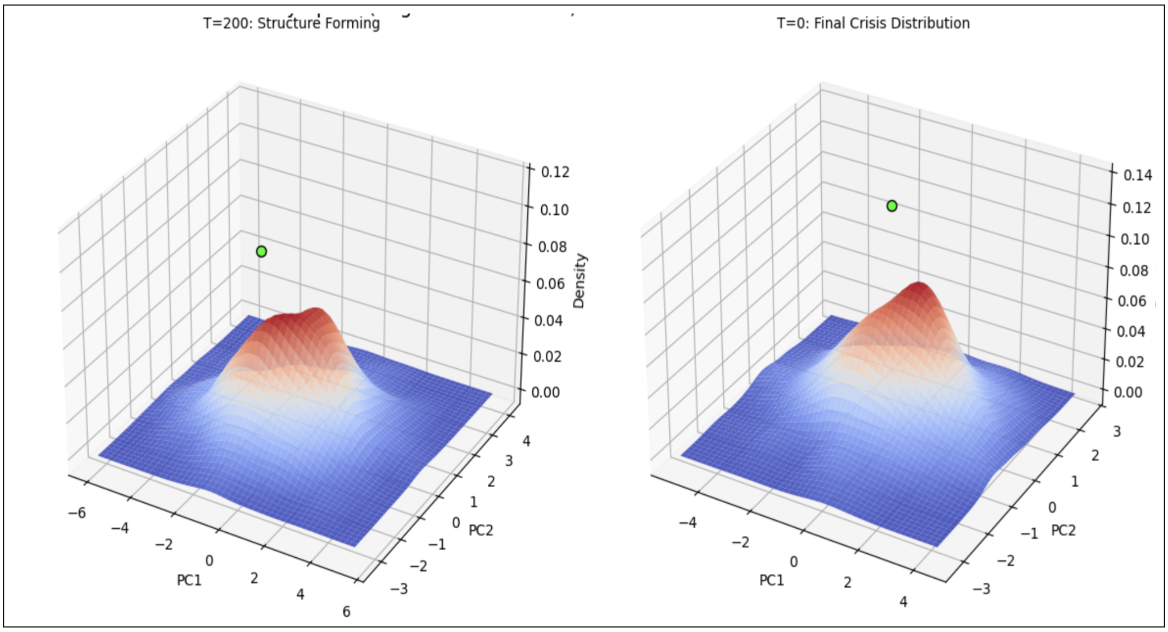


Figure 7.6: Evolution of the Market Probability Space. The 3D density plots (Projected on PC1/PC2) show the transformation from high-entropy noise (Left) to a structured, heavy-tailed crisis distribution (Right) via the reverse diffusion SDE.

Chapter 8

Conclusions

This study provides clear evidence that the incorporation of diffusion-generated synthetic stress episodes enhances the robustness of LSTM-based portfolio allocation models without introducing distortions in out-of-sample behaviour. The augmented model consistently demonstrates more stable dynamics, including lower average drawdowns, reduced downside volatility, and improved risk-adjusted performance. Moreover, the strict walk-forward evaluation framework confirms that these gains are not a result of data leakage or favourable hindsight, but arise from genuine improvements in the model's ability to navigate unseen market regimes.

Key Contribution. This work presents the first fully walk-forward, multi-period evaluation of diffusion-based synthetic data augmentation for multi-asset LSTM portfolio forecasting under realistic transaction cost assumptions in the Indian equity market. The findings highlight that carefully constructed synthetic crash episodes can serve as an effective regulariser, improving model reliability in both normal and stress environments while preserving long-term return generation.

Chapter 9

Extensions and Future Work

- **Regime-Specific Synthetic Sampling.** Future work can incorporate regime-conditioned generation more explicitly by learning separate score networks for low-volatility, high-volatility, and transitional phases. Such an approach aligns with the volatility-regime literature (Harvey et al., 2018) and may further improve the model's responsiveness to structural breaks.

- **Multivariate Diffusion Architectures.** Extending beyond the current DDPM to architectures explicitly designed for temporal and multivariate dependencies, such as *TimeGrad* or *ScoreFlow* (Song et al., 2021), may yield stronger fidelity in cross-asset joint dynamics and contagion modelling.

- **Transformer-Based Forecasting.** Given the success of self-attention mechanisms in capturing long-range dependencies (Vaswani et al., 2017; Lim et al., 2021), an important direction is to integrate synthetic augmentation with transformer architectures such as TFT or Autoformer. These data-hungry models may particularly benefit from the enriched training distribution generated by diffusion models.

- **Cross-Market and Cross-Asset Generalization.** Testing the augmentation pipeline on other geographies (e.g., developed markets) or asset classes (e.g., commodities, fixed income, or crypto) would reveal the extent to which diffusion-augmented learning improves generalization under different statistical regimes (Cont, 2001). Such experiments are essential for assessing the portability of the proposed methodology.

Bibliography

[1] Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77–91.

[2] Sharpe, W. F. (1966). Mutual Fund Performance. *Journal of Business*, 39(1), 119–138.

[3] Sortino, F. A., & Price, L. N. (1994). Performance measurement in a downside risk framework. *Journal of Investing*, 3(3), 59–64.

[4] Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1007.

[5] Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *Journal of Business*, 36(4), 394–419.

[6] Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236.

[7] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.

[8] Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory networks. *PLOS ONE*, 12(7), e0180944.

[9] Sen, J., Datta Chaudhuri, T. (2021). A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. *Annals of Data Science.*

[10] Singh, R. (2023). Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market. *MDPI Engineering Proceedings.*

[11] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 6840–6851.

[12] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. *International Conference on Learning Representations (ICLR).*

[13] Tashiro, Y., Song, J., Song, Y., & Ermon, S. (2021). CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34.

[14] Kotelnikov, A., Baranchuk, D., Rubachev, I., & Babenko, A. (2023). TabDDPM: Modeling Tabular Data with Diffusion Models. *International Conference on Machine Learning (ICML).*

[15] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. *arXiv preprint* arXiv:1701.07875.

[16] de Prado, M. L. (2018). *Advances in Financial Machine Learning.* Wiley.

[17] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.

[18] Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. *arXiv preprint* arXiv:1607.06450.

[19] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.

[20] Prechelt, L. (1998). Early stopping - but when? In *Neural Networks: Tricks of the Trade*, 53–67. Springer.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[22] Lim, B., Arıkan, S. O., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *International Journal of Forecasting*, 37(4), 1748–1764.

[23] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Advances in Neural Information Processing Systems (NeurIPS)*, 34.

[24] Dosovitskiy, A., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR).*

[25] Zhang, Y., et al. (2023). A Survey on Transformers for Time Series Modeling. *arXiv preprint* arXiv:2202.07125.

Chapter 10

Publications / Communications

None.