**Project Report: Deep Quantile Regression and GAN-based Forecasting for VaR and ES**

**📊 Objective**

The primary objective of this project is to build a robust forecasting system for **Value at Risk (VaR)** and **Expected Shortfall (ES)** using deep learning methods. These two measures are critical for assessing downside risk in financial markets and play a vital role in risk capital allocation, regulatory compliance, and investment decision-making.

Traditional models such as **Historical Simulation**, **Variance-Covariance**, or **GARCH** often fall short in capturing the **long-memory**, **non-linearity**, and **tail dependency** present in financial time series.

**🎓 Models Used**

**1. Deep Quantile Regression using Mogrifier-LSTM for VaR**

* **Architecture**:
  + Mogrifier-LSTM Layer
  + Fully Connected Output Layer
* **Key Parameters**:
  + hidden\_dim=32, num\_layers=1, alpha=0.05
* **Reason for Choice**:
  + The **Mogrifier-LSTM** architecture efficiently captures **temporal dependencies**, **clustering of volatility**, and **long memory** effects by augmenting the input and hidden state interactions.
  + It addresses the shortcomings of plain LSTM or GRU models in complex time series.
  + Quantile regression enables **direct estimation** of the α-quantile (VaR level), eliminating the need for assumptions like normality or linearity.

**2. Generative Adversarial Network (GAN) for ES**

* **Architecture**:
  + Generator: Dense layers generating return scenarios.
  + Discriminator: Dense layers classifying real vs synthetic returns.
* **Key Parameters**:
  + z\_dim=100, hidden\_dim=64, n\_samples=1000
* **Reason for Choice**:
  + ES requires understanding the **tail risk**, which traditional methods can't simulate accurately.
  + GANs can model **high-dimensional, non-Gaussian** distributions and produce realistic synthetic return paths.
  + By generating plausible future returns, the GAN allows us to simulate losses beyond VaR and calculate ES directly from the left tail of the synthetic distribution.

**🔬 Dataset**

* **Asset**: BlackRock (Ticker: BLK)
* **Source**: [Stooq](https://stooq.com/)
* **Time Frame**: 2020-05-04 to 2023-12-31
* **Features Used**: Daily closing prices
* **Preprocessing**:
  + Resampling to daily frequency
  + Forward-filling missing values
  + Normalization to zero mean, unit variance for training

**🎮 Evaluation Metrics**

**For VaR:**

* **Hit Ratio**: Measures how often actual losses exceed VaR. Should be close to α (e.g., 0.05).
* **Kupiec Test**: Statistical test for verifying if the number of violations is consistent with the confidence level.

**For ES:**

* **Empirical ES**: Mean of returns that fall below VaR.
* **Naive Historical ES**: Used as a baseline to compare against GAN-based ES.

**📊 Results Summary**

* **VaR Hit Ratio**: 0.0513 (very close to 0.05 target)
* **Kupiec Test p-value**: 0.8097 (supports correct coverage)
* **Estimated ES**: -0.336591
* **Naive Historical ES**: 0.503660
  + Indicates the GAN generated more conservative (safer) tail losses.

**🔄 Comparison to Traditional Approaches**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Historical** | **Parametric** | **GARCH** | **DeepQR + GAN (This Project)** |
| Non-linearity | ❌ | ❌ | ✅ | ✅ |
| Tail Risk Modeling | ❌ | ❌ | ✅ (limited) | ✅ (via GAN) |
| Memory Handling | ❌ | ❌ | ✅ | ✅ (LSTM-based) |
| Flexibility | ❌ | ❌ | ❌ | ✅ |

**📅 How to Run**

1. Clone the repo and install dependencies

pip install -r requirements.txt

1. Download data:

python data/download\_data.py

1. Preprocess:

python data/preprocess\_data.py

1. Train models:

python scripts/train\_var.py

python scripts/train\_es.py

1. Evaluate:

python scripts/evaluate\_models.py

**💅 Hyperparameter Tuning Tips**

* Try different values for:
  + hidden\_dim (16, 32, 64)
  + sequence\_length (10, 20, 30)
  + alpha (0.01, 0.05, 0.1)
* Enable backtesting by using a rolling window on the dataset.
* Test different architectures: Transformer-based QR, WGANs, CVaR loss variants.

**🌟 Conclusion**

This project demonstrates a **state-of-the-art approach** to financial risk modeling, leveraging deep learning to overcome traditional shortcomings. The use of **Mogrifier-LSTM** and **GANs** enables accurate estimation of both VaR and ES, capturing the complex dynamics of financial returns.

This pipeline offers an end-to-end system that can be adapted to other assets and extended for multivariate or portfolio risk estimation in future work.

**🌐 Future Work**

* Implement **multivariate VaR/ES** for portfolios.
* Include **macroeconomic indicators** as features.
* Test with **extreme value theory (EVT)** enhancements.
* Integrate with **real-time dashboards** for trading or investment risk monitoring.

Let me know if you want a PDF version of this report or integration with LaTeX for submission!