Technical report

Bao Dien Quoc Nguyen
baondq.ai.research.engineer@gmail.com
AIC - FPT Software
Ho Chi Minh city, Vietnam

ABSTRACT

This report details my techniques using in the Market Scenario Generator Hackathon: From Stability to Storms.

ACM Reference Format:
Bao Dien Quoc Nguyen. 2018. Technical report. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Market Scenario Generator Hackathon: From Stability to Storms).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The Hackathon requires to generate a high quality synthetic time series to approximate the financial time series in the U.S. market under regular and crisis conditions. Besides the conditional generation problem, I notice that time series has multiple variables and these variables may be correlated, as shown in figure 1 and figure 2. Another issue is the extreme imbalance between the number of samples in regular and in crisis condition (with ratio of roughly 9:1). My solution is to use classifier free diffusion model with some modifications to handle these problems.

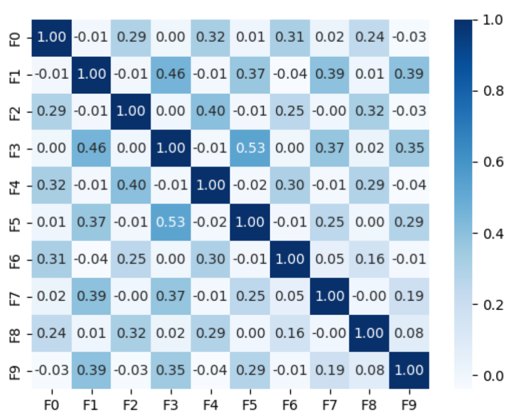

Figure 1: Correlation in regular condition

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

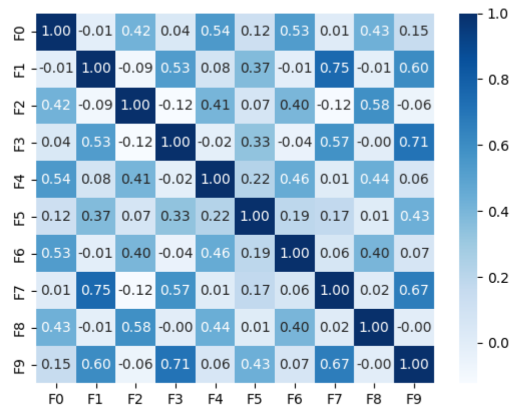

Figure 2: Correlation in crisis condition

2 SOLUTION DETAILS

I follow the two references [1] and [2] to overcome the imbalanced problem of conditional generation. To preserve the correlation between features in multivariate time series, I replace standard normal noise by a multivariate Gaussian noise with covariance matrix empirically computed from training data. The training algorithm is given as in the following. I use cosine noise schedule and bi-directional LSTM as model base. The model was trained by AdamW optimizer. Learning rate was warmed up in the first 500 steps, before decaying by a cosine function.

Algorithm 1 Training algorithm

```
for Every batch of size N do
    for each sample in this batch do
        sample ϵ ∼ N(0, Σ), t ∼ U(0..T)
        compute x_t = sqrt(α̂_t) x_0 + sqrt(1 − α̂_t) ϵ
        compute L_DM = ||ϵ − ϵ_θ(x_t, y)||^2
        sample y′ from class distribution
        compute L_r = tτ||ϵ_θ(x_t, y) − sg(ϵ_θ(x_t, y′))||^2
        compute L_rc = tτ||sg(ϵ_θ(x_t, y)) − ϵ_θ(x_t, y′)||^2
        update with loss L_DM + L_r + 0.25 L_rc
    end for
end for
```

3 OFFLINE RESULTS

The offline evaluation result is in the following table.

Table 1: Detail of training

Parameters	value
Batch size	64
T	500
Learning rate	0.0005
τ	1
number of layers in LSTM	1
hidden dim	64
embedding dim	64

Table 2: Offline evaluation result

metric	mean	std
hist loss	15.6	0.243
acf loss	0.270	0.024
cov loss	8.04e-05	4.74e-06
cross corr loss	2.97	0.234
var	0.019	0.009
es	0.008	0.001
auc	0.964	

REFERENCES

[1] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
[2] Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. 2023. Class-balancing diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18434–18443.