Multi-Horizon Volatility Forecasting using Ordinary Differential Equation and Cross-stitch Networks

William Jallot[*1] Matthias Wyss[*1] Thierry Sokhn[*1]

Abstract

This paper introduces a novel architecture for predicting the volatility of limit order books (LOBs) using the FI-2020 dataset. Our approach combines the strengths of Ordinary Differential Equation (ODE) networks and Cross-Stitch networks to forecast the volatility on multi-horizon in high-frequency trading environments. The ODE network captures the continuous-time evolution of market dynamics. Cross-stitch network layers are used to improve forecasting performance by allowing one time horizon to influence others.

To evaluate our approach, we compare it against the Temporal Fusion Transformers (TFT) architecture, a state-of-the-art method for temporal forecasting tasks, on the same FI-2010 dataset. Our architecture outperforms the TFT architecture in Mean Absolute Relative Error.

1. Introduction

Predicting the volatility of financial markets is crucial for applications such as risk management, portfolio optimization, and algorithmic trading.

A limit order book (LOB) contains all buy and sell orders for a financial asset, organized by price levels. It consists of bid prices (highest prices buyers are willing to pay) and ask prices (lowest prices sellers are willing to accept), along with corresponding volumes. The order book reflects market supply and demand, determines the asset's market price, and indicates liquidity by showing the depth of orders across price levels. In high-frequency trading, the rapid evolution of the order book provides crucial insights into market dynamics.

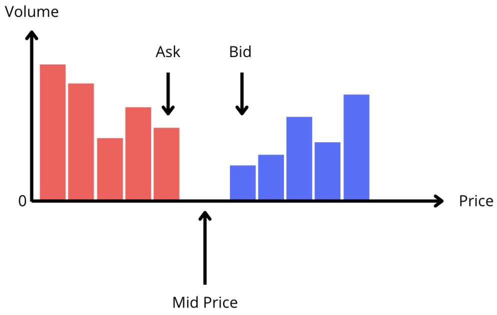

Figure 1. Example of a Limit Order Book

However, one faces a lot of challenges as LOB data is complex, high-dimensional, and non-stationary, especially when it comes to precise and efficient forecasting over a range of time horizons.

Traditional statistical models and basic machine learning approaches often fall short in capturing the intricate temporal dependencies and nonlinear patterns within LOBs. Recent advancements in deep learning have demonstrated potential, but existing architectures typically fail to unify continuous-time modeling with multi-horizon forecasting, leaving room for improvement in capturing long-term market behavior while maintaining short-term precision.

This research presents a novel approach that blends stitch networks with Ordinary Differential Equation (ODE) networks in order to overcome these challenges. The ODE network offers a strong basis for modeling volatility since it reflects the continuous-time evolution of market dynamics.

We trained the TFT architecture on this dataset to effectively compare the results of our proposed architecture in outperforming baseline methods for volatility prediction in multi-horizon tasks.

The remainder of this paper is organized as follows: Section 2 provides the necessary preliminaries, ethical risks and discusses related work. Section 3 presents our proposed model, detailing its architecture and key components. Section 4 describes the experimental setup and evaluates the model's performance. Finally, Section 5 concludes the paper and outlines potential directions for future research.

[*]Equal contribution [1]Department of Data Science, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. Correspondence to: William Jallot <william.jallot@epfl.ch>, Matthias Wyss <matthias.wyss@epfl.ch>, Thierry Sokhn <thierry.sokhn@epfl.ch>.

2. Preliminaries

The predictability of stock markets has been extensively scrutated since their creation, with evidence suggesting that those financial markets are partially predictable even though they may apparently and inherently seem complex. Traditional statistical methods such as simple moving averages, often fail to capture these dynamics. Machine learning approaches, by contrast, excel at modeling such nonlinearity without requiring prior assumptions, making them well-suited for financial data analysis.

Recent work has focused on applying machine learning techniques to predict limit order book (LOB) data. Early studies utilized linear statistical methods. While effective to some extent, these approaches were surpassed by end-to-end nonlinear learning frameworks, which integrate feature extraction into neural layers and achieves superior results on datasets like FI-2010 (Ntakaris et al., 2018). These findings emphasize the importance of data-driven feature extraction.

Deep learning has further advanced the field. In fact, Convolutional Neural Networks (CNNs) automatically tune features to optimize performance and have been widely applied in different fields, ranging from protein analysis to computer vision. However, their use in financial data remains limited, with existing implementations often gatekeeped by private. Meanwhile, Long Short-Term Memory (LSTM) networks have gained traction in financial time-series analysis due to their ability to address vanishing gradients. Studies monitoring LSTMs on large-scale LOB datasets have demonstrated robust out-of-sample prediction accuracy.

A recent contribution in this area is DeepLOB: Deep Convolutional Neural Networks for Limit Order Books (Zhang et al., 2019), which introduced a hybrid CNN-LSTM model for LOB prediction. This work incorporates the Inception model, representing a significant advancement in using deep learning for financial data analysis.

Concerning multi-horizon forecasting, a significant and recent contribution by Google Cloud AI is Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting (Lim et al., 2020), which introduced a new architecture combining attention mechanisms with gating layers to achieve both high performance and interpretability. This Temporal Fusion Transformer effectively models complex temporal dependencies and contextual information, making it particularly suited for multi-horizon forecasting tasks across diverse domains.

2.1. Ethical Risks

Neural networks used for financial volatility prediction present lots of ethical risks, particularly regarding sustainability, market fairness, and model explainability and accountability. One critical concern is the high electrical energy consumption associated with training and deploying these neural network models. This indeed contributes to increase various carbon emissions, impacting climate change, and governments and countries tasked with regulating emissions from the different COPs. Because of the computational intensity of financial forecasting and the constant and intense demand for accurate predictions and solutions, the environmental impact of these models is both severe and likely to continue without proper intervention with ethical reasoning.

Furthermore, due to their opacity, neural networks increase the possibility of market manipulation. Rich market players might take use of our open-source volatility forecasts and models to take preventative action, which would worsen market volatility and put investors in an unfair situation. Such abuse has the capacity to exacerbate financial system injustices and destabilize markets. Furthermore, the ”black-box” structure of neural networks makes interpretation more difficult and encourages an excessive dependence on forecasts without a clear grasp of their limitations. Systemic hazards may be increased by this opacity when uncertainty is high. Although these ethical hazards were recognized throughout the project, mitigation strategies could not be implemented due to time, resource, and skill restrictions. In order to overcome these obstacles, further study and cooperation are needed to advance ethical AI methods in financial forecasting that prioritize justice, sustainability, and transparency.

2.2. Data set overview

One of the most popular and often used benchmarks in financial machine learning research is the FI-2010 (Ntakaris et al., 2018) dataset. The limit order book (LOB) data from a Nordic stock exchange is provided at the millisecond level, representing the complex microstructural dynamics of financial markets. The dataset is especially well-suited for high-frequency trading and volatility forecasting research since it contains comprehensive information about bid and ask prices, volumes, and order placements across numerous LOB layers.

In this work, the decimal-normalized dataset derived from FI-2010 (Ntakaris et al., 2018) was used to standardize feature scales and ensure numerical stability during training. The features selected for the model include the volume, bid price, and ask price across all LOB layers. This selection captures the core dynamics of order flow and liquidity, providing a rich representation of market behavior.

By focusing on these features, the method minimizes feature selection redundancy while efficiently utilizing the dataset's granularity to depict the market's current situation. This guarantees that the model inputs are highly informative and computationally efficient, which is crucial for tasks like volatility prediction and multi-horizon forecasting.

The FI-2010 dataset's goal of improving predictive modeling in financial environments, especially for tasks requiring precise comprehension of high-frequency market dynamics, is in line with the dataset's extensive temporal and structural information and emphasis on critical traits.

2.3. Data pre-processing

The FI-2010 dataset is composed of 5 stocks from the NASDAQ Nordic stock market for a time period of ten consecutive days. We started by computing the midprice for each event (which is a variation of the midprice whether it be an increase or a decrease) on the dataset using the following formula:

$$\text{Midprice} = \frac{\text{Ask} + \text{Bid}}{2}$$

To achieve better results, we decided to split these 5 stocks to get 5 training sets and 5 testing sets independent to each other, to feed to the model individually. When there was a big drop in the mid price, this meant it was a new stock, by doing so we obtained the following separation for the training. The Figure 2 shows the training of the 5 split stocks, the testing looks similar. With this method, we had to train and test our model 5 times in total, one time for each stock. Since our model is forecasting volatility, we had to derive this from the LOB dataset. It is widely believed that the returns of the mid-price follow a geometric Brownian motion. Under this assumption, we computed volatility as the standard deviation of the log returns over a specified time window, the log return is defined as :

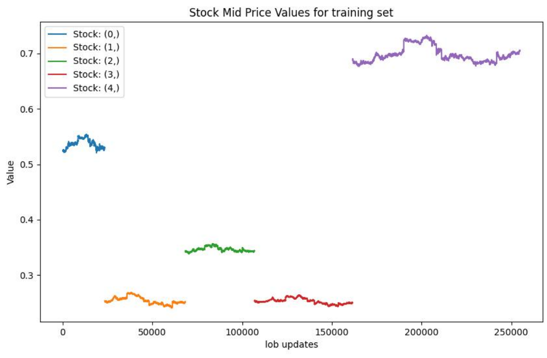

Figure 2. Separated 5 training stocks

$$\log(P_{t+1}) - \log(P_t)$$

In order to scale the values around mean with a unit standard deviation, we standardize each stock individually, and for each horizon, which have been chosen to be k=20, 50 and 100 like in (Zhang et al., 2019).

Using this method, we computed the volatility, using a rolling window adapted to each of our forecasted horizon (k=20, 50 and 100).

As we are working with high-frequency data, the short time intervals between updates of the limit order book can result in very low volatility values, with our target sometimes approaching zero. This behavior can lead the model to predict values of zero instead of providing more precise approximations. To address this issue, we scaled our volatility predictions by a factor of 1000. This scaling was reverted when computing performance metrics after training to ensure accurate evaluation of the model.

As the researchers did in the paper we mainly inspire ourselves from (Zhang et al., 2019), we use the $T = 100$ most recent states of the LOB as an input to our model for our dataset. Specifically, a single input is defined as:

$$X = [x_1, x_2, \cdots, x_t, \cdots, x_{100}]^T \in \mathbb{R}^{100\times 40},$$

where

$$x_t = \left[p_a^{(i)}(t), v_a^{(i)}(t), p_b^{(i)}(t), v_b^{(i)}(t)\right]_{i=1}^{n=10}.$$

Here, $p^{(i)}$ and $v^{(i)}$ denote the price and volume size at the i-th level of a limit order book.

3. Model

Our Cross-Stitch ODE architecture is composed of 2 Cross-Stitch units. A cross-stitch block is a mechanism used to learn shared representations between two or more neural network branches. It adaptively combines the feature maps from different branches, allowing the network to decide how much information to share.

To explain this architecture, we will use an example with a simple case where we have 2 tasks. Given two feature maps $\mathbf{x}_1$ and $\mathbf{x}_2$ from two network branches, a cross-stitch unit produces new feature maps $\mathbf{y}_1$ and $\mathbf{y}_2$ as follows:

$$\mathbf{y}_{task1} = \alpha_{11}\mathbf{x}_{task1} + \alpha_{12}\mathbf{x}_{task2},$$
$$\mathbf{y}_{task2} = \alpha_{21}\mathbf{x}_{task1} + \alpha_{22}\mathbf{x}_{task2},$$

where α_{ij} are learnable parameters that control the contribution of each input feature map to the output feature maps.

The parameters α_{ij} are initialized such that:

$$\alpha_{11} = \alpha_{22} = 1 \quad \text{and} \quad \alpha_{12} = \alpha_{21} = 0,$$

which allows the initial feature maps to remain unchanged. During training, the values of α_{ij} are updated to optimize the network's performance, enabling flexible sharing of features between branches. As we are working with several horizon, we decided to turn this cross stitch into a triangular one, as we thought shorter horizon should be able to influence shorter horizons, but not the other way around.

Our Cross-Stitch ODE architecture takes the 40 features as input for each of the three chosen tasks (with horizons $k = 20, 50, 100$). The architecture includes two Cross-stitch blocks (as shown in Figure 3, built with cross-stitch units as inspired by (Misra et al., 2016) with the possibility of experimenting with a different number of blocks). Following this, we apply an ODE layer using a standard ODE function (with plans to explore alternative ODE functions in future experiments). Finally, the output from the ODE layer goes through a Linear layer to produce the multi-horizon volatility forecasting with as activation function a SoftPlus, a smooth approximation to the ReLU.

To optimize the multi-task learning framework, we employed a Multi-Task Dynamic Loss function inspired by uncertainty weighting (Kendall et al., 2018). This approach dynamically balances multiple tasks, accommodating differences in task difficulty and scale during training. Each task is assigned an uncertainty parameter, represented as a learnable log variance which is updated during back-propagation. This method improves multi-task learning by adapting task importance dynamically, without manual tuning of task weights, handling tasks with different output scales (e.g., regression vs. classification), and enhancing overall model performance through efficient balancing of task contributions. By leveraging this dynamic weighting mechanism, the model achieves better convergence and more robust predictions across tasks with varying complexities and objectives.

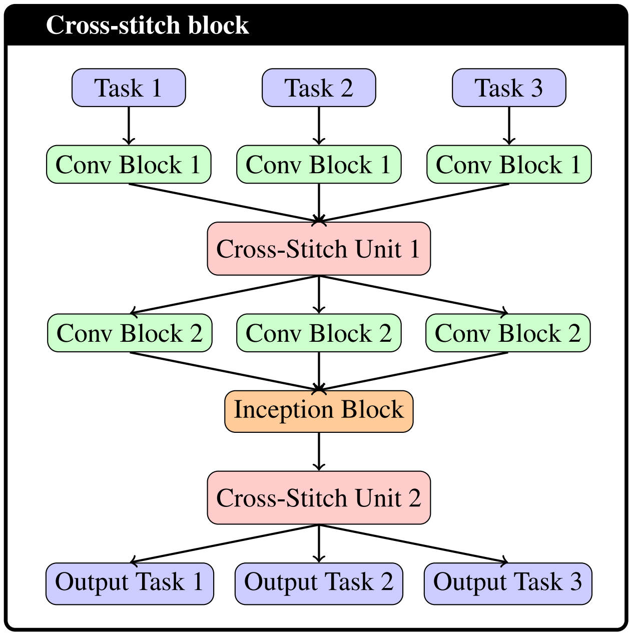

Figure 3. Cross-stitch block structure with an added Inception block.

To compare our Cross-Stitch ODE network, we adapt the Temporal Fusion Transformer (TFT) architecture from (Lim et al., 2020) on the same dataset (Ntakaris et al., 2018). The TFT is designed to provide a robust quantile prediction of volatility as a range, capturing the inherent uncertainty and variability in the data. It predicts three quantiles for each time step:

- **Low Quantile (e.g., 10th percentile):** Represents the minimum expected volatility, providing a lower bound.
- **Median Quantile (e.g., 50th percentile):** Represents the most likely volatility, serving as the central prediction.
- **High Quantile (e.g., 90th percentile):** Represents the maximum expected volatility, offering an upper bound.

We train this architecture by adding as static input the information of which stock it is predicting on, and selected as features the first layer of the order book, along with the spread, defined as:

$$\text{Spread} = P_{\text{ask}}^{(1)} - P_{\text{bid}}^{(1)}$$

where $P_{\text{ask}}^{(1)}$ is the price of the best ask and $P_{\text{bid}}^{(1)}$ is the price of the best bid in the first layer of the book. And we used the median quantile (50th percentile) as the output for comparison with our proposed Cross-Stitch ODE network. This allows for a direct evaluation of the central tendencies of the predictions.

4. Experiments

To evaluate the different models, we used the Mean Absolute Relative Error (MARE) metric that makes it suitable for comparing predictions across different horizons with different scales and provides a more interpretable measure for low-value targets.

Table 1 presents MARE of our Cross-Stitch ODE model for each stocks on the different chosen horizons (k = 20, 50, 100) for the 5^{th} epoch. Table 2 presents the median quantile MARE of the TFT model for all stocks on the different chosen horizons for the 10^{th} epoch.

k	**Stocks**				
	Stock 0	Stock 1	Stock 2	Stock 3	Stock 4
20	0.5766	0.6055	0.7765	1.0000	1.0000
50	0.3656	0.6291	0.8665	0.8379	0.8465
100	0.2211	0.4993	1.2328	0.9783	0.9134

Table 1. MARE for Cross-Stitch ODE for different horizons k

k	Stocks
20	4.1120
50	2.1209
100	2.3096

Table 2. MARE for median quantile TFT for different horizons k

5. Conclusion

On the FI-2010 dataset, the TFT model performs below expectations and in poor terms. Our Cross-stitch ODE architecture, on the other hand, continuously produces better outcomes throughout every prediction horizon. This increase may be ascribed to the model's improved capacity for feature integration and its ability to better capture complex temporal dynamics, both of which are very important for the structure and properties of this dataset.

The ODE function itself could be changed as a possible improvement. The model's performance could be further enhanced, for example, by using a GARCH function, which has shown promise in forecasting volatility. Furthermore, we could improve the model's representational capability by adding and mixing other and several Cross-Stich ODE blocks, with different used convolutions.

However, the long and time-consuming training periods and inadequate computing power restricted our capacity to test and improve further. These limitations restrained our ability to test more complex design configurations and better investigate the parameter space.

References

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. URL https://arxiv.org/abs/1705.07115.

Lim, B., Arik, S. O., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting, 2020. URL https://arxiv.org/abs/1912.09363.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch networks for multi-task learning, 2016. URL https://arxiv.org/abs/1604.03539.

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., and Iosifidis, A. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37 (8):852–866, 2018. doi: https://doi.org/10.1002/for.2543. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2543.

Zhang, Z., Zohren, S., and Roberts, S. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, June 2019. ISSN 1941-0476. doi: 10.1109/tsp.2019.2907260. URL http://dx.doi.org/10.1109/TSP.2019.2907260.