Financial Forecasting in the Data-Centric Era

Chen Liu

Discipline of Business Analytics
The University of Sydney
2025

Supervisor: Associate Prof. Minh-Ngoc Tran
Co-Supervisor: Prof. Robert Kohn

This dissertation is submitted for the degree of
Doctor of Philosophy

Statement of Originality

This is to certify that the content of this thesis is my own work. This thesis has not been submitted for any other degree or purpose. I certify that the intellectual content of this thesis is the product of my own work, and that all assistance received in preparing this thesis and all sources have been acknowledged.

October 7, 2025

Student: Chen Liu

Date

Acknowledgements

I would like to thank the many people who greatly helped with the papers comprising this thesis. I am particularly grateful to my collaborators (alphabetically by last name): Richard Gerlach, Robert Kohn, Richard Philip, Minh-Ngoc Tran, and Chao Wang, and to the many reviewers for their helpful comments and discussions. I extend special thanks to my supervisor, Minh-Ngoc Tran, for his unconditional support and guidance throughout my PhD studies. Finally, I want to thank my parents for their unwavering encouragement in every aspect of my life. It has been a rewarding journey.

Author Attribution Statement

The chapters 2-4 of this thesis are derived from the following three papers. In each case, I designed the study, analysed the data, and drafted the manuscript.

1. Chapter 2: **Liu, C.***, M.-N. Tran, C. Wang, R. Gerlach, R. Kohn. *Financial Forecasting in the Data-Centric Era* (Under Review).

2. Chapter 3: **Liu, C.***, P. Richard, M.-N. Tran. *Price Discovery in the Data-Centric Era* (Under Review).

3. Chapter 4: **Liu, C.***, W. Chao, M.-N. Tran, R. Kohn. (2024). *A Long Short-Term Memory Enhanced Realized Conditional Heteroskedasticity Model. Economic Modelling.*

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

October 7, 2025

Student: Chen Liu — **Date**

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

October 7, 2025

Supervisor: Minh-Ngoc Tran — **Date**

Artificial Intelligence Statement

During the preparation of the thesis the author used OpenAI ChatGPT for the purposes of grammar checking of the introduction and conclusion chapters. The author confirms that where text was modified by generative AI, the content was reviewed for possible errors, inaccuracies, and bias. The author takes full responsibility for the submitted thesis and ensures the work is their own and has used generative AI within the parameters of use.

October 7, 2025

Student: Chen Liu

Date

Australian Government Support

The research reported in this thesis was supported by the PhD Scholarship in Deep Learning-Based Modelling and Forecasting (SC4345), awarded to the PhD candidate.

Abstract

Deep learning (DL) has become an important tool in finance research, with applications ranging from asset pricing and portfolio allocation to risk management. However, empirical findings on its performance are often mixed, with DL frequently underperforming econometric or traditional machine learning methods. This work argues that a primary source of this inconsistency is the mismatch between the data-intensive nature of neural networks (NNs) and the data-scarce environments typical of many financial tasks. When trained at sufficient scale, NNs can deliver competitive or superior forecasts and can also inform theoretical research.

The first part of the thesis examines how data volume, model size, and architectural choice affect NN performance in financial forecasting. Using a dataset of more than 10,000 equities for volatility prediction, we find that performance gains from data scaling substantially exceed those from model scaling or architectural variation. We further demonstrate that a global training approach, which trains a single model across all assets, should be adopted as standard practice. The second part investigates the use of globally trained NNs as a tool for testing price discovery theories. We find that theory-driven variables effectively summarize the static state of the market but fail to capture temporal information in order flow. By training NNs on sequences of raw order book data, we provide empirical evidence against the Markovian assumption in traditional theory models. Finally, we show that NN performance can be further enhanced by hybridizing econometric and deep learning structures. We introduce the Realized Recurrent Conditional Heteroskedasticity (RealRECH) model, which augments the RealGARCH framework with a Long Short-Term Memory (LSTM) network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Financial forecasting is a central objective in economics and finance. Accurate predictions of returns and volatility underpin almost every aspect of the finance industry ranging from asset pricing to derivative valuation. For decades, parsimonious econometric models have dominated this domain. For example, in volatility forecasting, GARCH-family models such as GARCH (Bollerslev, 1986), EGARCH (Nelson, 1991), GJR (Glosten et al., 1993), Realized GARCH (Hansen et al., 2012), and HAR-family models, such as HAR (Corsi, 2009) or the measurement-error-robust HARQ (Bollerslev et al., 2016), remain standard. These models embed economic theory and stylized facts, such as volatility clustering and leverage asymmetry, yielding forecasts that are both accurate and interpretable.

On the other hand, recent advances in computational power and data availability have enabled deep learning (DL) to transform many empirical fields such as computer vision, natural language processing, and scientific discovery. This progress has stimulated substantial interest in applying powerful DL models to financial prediction across asset pricing (Gu et al., 2020; Kelly et al., 2024), risk management (Christensen et al., 2023; C. Zhang et al., 2023), option pricing (Almeida et al., 2023; Bali et al., 2023), market microstructure (Bogousslavsky et al., 2024; Kwan et al., 2024), and portfolio management (Chevalier, 2022; Ni et al., 2024). Relative to econometric approaches, DL can capture complex high-dimensional, nonlinear dependencies and exploit unstructured information, enabling it to achieve superior predictive accuracy.

However emperical results on neural networks (NNs) for financial forecasting remains mixed. While some studies report superior performance, others find that NNs underperform econometric or traditional machine learning benchmarks (Almeida et al., 2023; Bali et al., 2023; Capponi & Yu, n.d.; Gu et al., 2020; Makridakis et al., 2018). Limited interpretability and higher computational cost further impedes NNs' adoption in finance indsutry. The field possesses powerful new tools but lacks consensus on when they deliver genuine gains and which conditions govern their performance. This thesis addresses this gap and investigates three questions:

- Do deep learning models consistently outperform econometric and machine learning benchmarks in financial forecasting?
- What factors determine NNs performance in financial forecasting?
- Can NNs uncover insights that traditional theoretical frameworks overlook, thereby informing theory development?

1.1 The Three Pillar Studies

This work investigates these questions through three complementary empirical studies, each forming a self-contained chapter. Collectively, these studies provide a comprehensive analysis of deep learning's role in finance by examining it through the lenses of model hybridisation, data scaling, and economic interpretability.

Chapter 2: The Data Scaling Effect

We first directly addresses the thesis's "data-centric" theme, investigating the critical role of data scale and diversity in neural network performance. This chapter posits that the inconsistent results in prior literature may stem from the conventional practice of local training which fits a separate model to each individual time series. This approach, while standard in econometrics, severely limits the data available for highly-parameterized NNs, leading to poor generalization.

Using a dataset of over 10,000 stocks for volatility forecasting, we investigate how data volume, model size, and architectural choice affect the performance of NNs. We first identify that performance gains from data scaling decisively outweigh those from model scaling and architectural choice. We further demonstrate that when applying NNs to financial data, the global training approach, which trains a single model on all time series, should be applied as standard.

Chapter 3: Interpreting Price Discovery

The third chapter delves into high-frequency market to explore the potential of DL for economic discovery and addresses the challenge of model interpretation. We use a unique, highly granular dataset of every order book event to benchmark traditional, theory-driven microstructure variables against the raw data stream of the limit order book (LOB).

The investigation proceeds in two stages. First, we compare linear models with non-linear ML models trained on theory-driven microstructure variables (e.g., depth imbalance, directional trade size) to quantify the importance of nonlinearity and interaction effects. Second, and more critically, we benchmark these models against state-of-the-art DL models that learn directly from raw Level 3 (L3) order book data. This comparison directly tests the efficiency of existing theory-driven variables. Another important contribution of this chapter is its challenge to the pervasive Markovian assumption in price formation—that future price changes depend only on the current state. By training a Transformer model on the history of LOB snapshots, this study empirically demonstrates that the temporal evolution of order flow contains significant predictive power, providing strong support for emerging theories of path dependency.

Chapter 4: Hybridizing Econometrics and NN Models

The forth chapter explores synergistic integration rather than outright replacement. We investigate whether combining classical econometrics and deep learning can yield a superior hybrid framework. This study introduces the Realized Recurrent Conditional Heteroskedasticity (RealRECH) model, which enhances the RealGARCH framework of

Hansen et al. (2012) with a Long Short-Term Memory (LSTM) network.

This investigation confronts the limitations of both parent methodologies. GARCH-type models, while interpretable and reliable, are less effective for capturing nonlinear dynamics and long-range dependencies in volatility. Pure "black-box" DL models, conversely, are difficult to interpret and may not properly incorporate known stylized facts. The RealRECH model uses the GARCH component to anchor the model to established theory and the LSTM component to flexibly capture additional complex patterns from data. By incorporating high-frequency realized volatility measures, this hybrid approach aims for state-of-the-art forecasting performance while retaining structural interpretability. This chapter thus examines whether a carefully constructed synthesis can outperform its individual components, providing a different answer to how DL can be productively integrated into the econometrician's toolkit.

1.2 Contribution

This thesis makes a twofold contribution to the financial econometrics literature. First, it identifies when and how neural network models can be applied to financial forecasting to maximise their effectiveness. Second, it interprets the behaviour of NNs in these settings and demonstrates how they can be used to test and guide theoretical development.

Conditions for NN Performance

Across the settings studied, NNs can outperform econometric and machine learning benchmarks, but only under specific conditions regarding the training regime, data availability, and input representation. We do not find support for the view that deep learning is a universal improvement over econometric and machine-learning models.

- **Local vs. Global**: When trained locally on a single time series, NNs rarely surpass simple econometric or machine learning baselines. By contrast, global training, which jointly estimates a single model across many related assets, yields materially better generalization and significantly improves the practicality of NNs in real-

world financial applications. We advocate adopting the global training approach as standard practice when applying NNs to financial forecasting.

- **The Power of Data Scale**: In our experiments, increases in the volume and cross-sectional diversity of training data deliver the largest and most systematic gains in out-of-sample accuracy. Improvements from data scaling dominate those from expanding model size or altering architecture.

- **Model Architecture and Size**: Holding data fixed, larger or more complex architectures provide limited incremental benefit for financial time series forecasting. The binding constraint appears to be the low signal-to-noise ratio of financial data rather than model capacity; excessive complexity can increase estimation variance without commensurate gains.

- **Input Format**: The choice of input format is decisive. Econometric and conventional ML models often perform best with theory-driven, engineered features. NNs exhibit their greatest comparative advantage when applied directly to high-dimensional raw data streams, where representation learning can extract structure that engineered variables may omit.

- **Hybrid Models**: Embedding NN components within established econometric frameworks often yields hybrids that improve both in-sample fit and out-of-sample forecasting under standard statistical and economic loss functions, while retaining elements of structural interpretability from the econometric core.

Interpretation and Economic Insights

NNs constitute a modelling paradigm distinct from traditional econometrics: rather than imposing a prespecified structure derived from theory, they learn stylized facts directly from rich data environments. We use interpretability analyses and targeted experiments to demonstrate what these models learn and how such evidence can inform theory.

1. **Learning Stylised Facts**: Feature-attribution, ablation, and related diagnostics

indicate that globally trained NNs are able to recover well-documented stylzed facts. In volatility forecasting, they correctly capture volatility clustering and leverage effects. In high-frequency return forecasting, the models assign sustained importance to order-flow proxies (e.g., trade direction and volume) and order-book imbalance, consistent with the dominant role of these variables in short-horizon price discovery.

2. **Guiding Theoretical Research**: Using price discovery as a case study, we benchmark theory-motivated econometric models against NNs trained on raw order flow to highlight their complementary strengths. Existing theory-driven variables capture static book states effectively, while NNs extract additional predictive information from the temporal dynamics of order flow. These findings have two implications for theoretical research: the need to place greater emphasis on modelling order-flow dynamics and to reassess common Markovian assumptions in traditional microstructure model.

Together, these findings clarify when NNs are likely to add value in financial research and illustrate how they can be used as empirical instruments to test and refine finance theory.

1.3 Thesis Structure

The thesis is organised as follows.

- **Chapter 2** analyses the factors affecting NN performance in financial forecasting and compares local and global training to identify the conditions under which NNs perform well.
- **Chapter 3** examines high-frequency market microstructure, using deep learning to test price-formation theories and to document path dependence in order flow.
- **Chapter 4** evaluates hybrid models that combine econometric structures with NNs to improve volatility forecasting.

- **Chapter 5** presents the key findings, implications, limitations, and directions for future research.

The appendices provide supplementary tables, figures, and technical details. Code for the experiments is available at https://github.com/VBayesLab and https://github.com/cqlc94.

Chapter 2

The Data Scaling Effect

2.1 Introduction

Neural networks have become an important tool in empirical finance and financial econometrics. However, a fundamental divide persists in how these models should be trained. Some studies adopt a *global* approach, training a single model on all available time series (e.g., Almeida et al., 2023; Gu et al., 2020; C. Zhang et al., 2023). Many others adhere to the standard econometric practice of *local* estimation, fitting models to each series individually (e.g., Bogousslavsky et al., 2024; Chevalier, 2022; Easley et al., 2021). This methodological divergence is not trivial. We argue that the local approach is fundamentally misaligned with the data-intensive nature of NNs, which require large, diverse datasets to realize their full potential (Das et al., 2024; Kaplan et al., 2020; Zhai et al., 2022). This chapter, therefore, investigates a core question: Should a global training approach be the standard when applying NNs to financial forecasting?

To answer this question, we conducted a comprehensive empirical study forecasting the daily volatility of over 10,000 individual stocks across global markets from 2014 to 2024. Volatility forecasting is a cornerstone of risk management, and its higher predictability than first-moment return makes it an ideal testing ground (e.g. Bollerslev, 1986; Bucci, 2020; Corsi, 2009; Hansen & Huang, 2016; Zhao et al., 2024). We systematically examined the effectiveness of traditional econometric models and various NN architectures under

both local and global training regimes. By incrementally increasing the number of training series and the model size, we documented the distinct scaling properties of these models and established the mechanisms driving their performance.

We establish four main results. First, we ask whether homogeneity among financial time series guarantees the effectiveness of the global approach. We find the answer depends critically on model complexity. Simple, low-parameter econometric models perform poorly in a global setting, with accuracy deteriorating as we add more series to the training pool. In contrast, the performance of highly parameterized NNs improves significantly as the training pool grows. We test this further by comparing models trained on homogeneous stock groups (e.g., by industry or country) against models trained on diversified pools of the same size. The results show that diversity among financial time series is also a key driver of NNs' performance in mitigating overfitting, consistent with the bias-variance trade-off literature (e.g. Belkin et al., 2019; Geman et al., 1992). Our findings also clarify the mixed view in the literature regarding the effectiveness of NNs compared to their econometric counterparts. We show that locally trained NNs do not reliably outperform simple econometric benchmarks, whereas globally trained NNs consistently outperform locally trained NNs and econometric baselines in various statistical and economic metrics.

Second, we show that for financial forecasting, the effect of data scaling dominates the effect of model scaling and model architecture. We find that forecast accuracy improves substantially as the number of time series in the training set increases. However, beyond a relatively small model size (approximately 200 parameters), increasing the size of the NN yields no improvement. We further tested four different foundational NN model architectures, and we find that the specific choice of NN architecture has a minimal impact on model performance. These findings present a notable departure from other domains in computer science, where performance scales predictably with model complexity (Das et al., 2024; Kaplan et al., 2020; Zhai et al., 2022), and suggest that for many financial tasks, the marginal value of data scaling far exceeds that of architectural choice.

Third, we demonstrate that global training enhances the practicality of NNs in the

finance industry. A single, universal NN can effectively generalize to all stocks, including unseen stocks and portfolios, while still offering superior forecasting accuracy. This significantly reduces the computational and managing cost of training thousands of NNs, especially considering the complex tuning process of NNs. Furthermore, once trained, these models require as little as twelve months of data to accurately forecast unseen stocks or portfolios, whereas econometric models typically require several years of historical observations for reliable estimation. This is particularly useful for newly listed stocks, which often lack sufficient data to fit a local model.

Finally, we extrapolate the key properties of global NNs. Through interpretation studies, we find that a global NN is able to learn the key stylized facts of stock volatility, such as clustering and leverage effects, directly from the data without any pre-specified structure. Trained on a diverse dataset, these models are more robust to outliers and adapt more effectively to changing market conditions.

Our results provide strong evidence that researchers and practitioners should adopt a global approach as the standard when applying NNs to financial forecasting tasks and focus more on data scaling over architectural choice. Ignoring the data-intensive nature of NNs can yield misleading conclusions about their performance.

This chapter is structured as follows: Section 3.1.1 reviews the related literature. Section 3.2 describes the local and global training approaches and introduces the various NN architectures used in this study. Section 3.4 analyzes how data size and model complexity affect model performance. Section 3.5 compares the performance of global NNs against local NNs and econometric baselines for Value-at-Risk and Expected Shortfall forecasting and offers an interpretation of the global models' characteristics. The appendix provides further technical details and training configurations.

We provide a Jupyter notebook with a pre-trained global volatility model that allows readers to forecast stock volatility, examine leverage effects, analyze responses to outliers, and compare results with stock-specific local models. The notebook is available at https://github.com/cqlc94/DeepVol.

2.1.1 Related Literature

This chapter contributes to several areas of recent literature. First, we contribute to the growing body of studies applying NNs to empirical finance. The seminal work of Gu et al. (2020) demonstrates that NNs can outperform econometric methods in a large cross-section of stock returns. Subsequent research has applied NNs to various domains, such as option pricing (Almeida et al., 2023; Bali et al., 2023), market microstructure (Bogousslavsky et al., 2024; Easley et al., 2021), and portfolio management (Chevalier, 2022; Ni et al., 2024). Our primary contribution to this literature is methodological. Through explicit comparison, we show that when comparing NN models and econometric benchmarks, econometric models should be trained locally and NNs should be trained globally to avoid misleading conclusions about their relative performance. Furthermore, we show that when applying NNs in financial forecasting, researchers should focus on data diversity, which plays a far more important role than architectural choices.

We also contribute to the extensive literature on volatility modeling and forecasting. Since the foundational ARCH/GARCH models (Bollerslev, 1986; R. F. Engle, 1982), a vast body of research has sought to improve forecast accuracy, notably by incorporating leverage effects (Nelson, 1991) and by utilizing realized measures (Corsi, 2009; Hansen & Huang, 2016; Hansen et al., 2012). More recently, researchers have applied NNs due to their ability to capture nonlinearity (e.g. Bucci, 2020; Nguyen et al., 2023; C. Zhang et al., 2023; Zhao et al., 2024). While prior studies have often focused on model architecture, this study focuses on the training approach that can be applied to any architectural research. We show that adopting a global training approach not only enhances volatility forecast accuracy but also produces a single, universal volatility model that improves its practicality through enhanced robustness to outliers, applicability in limited history scenarios, and reduced computational and tuning costs.

Our study also relates to the scaling law literature in computer science. Kaplan et al. (2020) first presented data and model scaling laws in natural language processing, and Zhai et al. (2022) studied these laws in computer vision applications. More recently, there has been a growing interest in universal time series models such as the billion-parameter

models proposed by Das et al. (2024) and Shi et al. (2025), which aim to model any time series. However, financial time series possess different dynamics, characterized by a low signal-to-noise ratio and smaller dataset sizes, necessitating a dedicated study. We show that the data scaling effect dominates the performance of NNs in finance, but at a much smaller scale than in other domains. Furthermore, due to limited data size and a low signal-to-noise ratio, the model scaling effect plays a minimal role in financial forecasting.

Finally, the global training approach also relates to panel data models in econometrics literature (e.g. Chamberlain, 1982). Panel data models often come in two forms: characteristics-based or time series-based. The former is, by its nature, using a global training approach. The latter often utilizes a multivariate regression framework designed to capture interdependence among multiple series. The global models assume that the time series share common patterns but remain independent. While global training utilizes a large pool of time series during the training phase, it is still a univariate model that forecasts each series independently.

2.2 Data

Our dataset consists of daily closing prices for all stocks with a market capitalization exceeding ten billion USD, listed on ten major exchanges across North America, Europe, Asia, and Oceania. The data, obtained from Reuters Refinitiv Workspace, span the period from January 1, 2014, to January 1, 2024. These criteria yield an extensive dataset comprising over 12,000 individual stock price series. To ensure sufficient data for reliably estimating the econometric baselines, we exclude stocks with fewer than 1,260 trading days (approximately five years) of data within the designated training period. This filtering yields a final dataset of 11,771 distinct stock series. Summary statistics for this sample are reported in Table 2.1.

Following standard partition practices in the machine learning literature for model development and out-of-sample evaluation, we partition the time series chronologically into training (the first 60%), validation (the next 20%), and testing (the final 20%)

sets. Accordingly, neural network models are trained on the training data, and their parameters are held fixed throughout the validation and testing periods. In contrast, and following standard econometric practice for time series forecasting, we evaluate the econometric benchmark using an expanding window. Their parameters are re-estimated daily throughout the test period, a procedure designed to maximize their predictive performance and provide a more stringent benchmark for the NN models.

This partitioning scheme results in a training period covering approximately 2014–2019, a validation period for tuning and early stopping of the neural networks from 2020–2021, and a final out-of-sample testing period for all models from 2022–2023.

Table 2.1: Data summary statistics by Exchange.

	NStocks	Avg Length	Avg Std	Avg Skew	Avg Kurt
Tokyo Stock Exchange	1987	2399	2.12	0.15	11.28
Shenzhen Stock Exchange	1954	2179	3.12	0.05	6.15
Shanghai Stock Exchange	1314	2165	2.81	0.01	6.48
New York Stock Exchange	1243	2441	2.46	-0.62	23.88
NASDAQ Stock Exchange	1123	2375	2.89	-0.44	23.13
London Stock Exchange	1077	2206	2.22	-0.46	23.73
National Stock Exchange of India	832	2360	2.83	0.48	9.14
Taiwan Stock Exchange	743	2397	2.01	0.14	10.20
Hong Kong Stock Exchange	509	2288	2.89	0.42	17.20
Toronto Stock Exchange	249	2387	2.77	-0.29	21.53
Euronext Paris	249	2499	2.09	-0.09	23.01
Frankfurt Stock Exchange	198	2410	3.31	0.04	17.39
Saudi Stock Exchange	149	2428	2.13	0.05	10.76
Australian Securities Exchange	114	2206	3.04	-0.28	18.82
Johannesburg Stock Exchange	30	2253	2.63	-0.00	32.74

2.3 Methodology

2.3.1 Training schemes

In this chapter, we systematically compare two distinct paradigms for model estimation: local training and global training.

Local training. Let $\boldsymbol{y} = \{y_t : t = 1, \ldots, T\}$ represent a daily time series of demeaned returns. In volatility modeling, the key quantity of interest is the conditional return

variance, $\sigma_t^2 = \text{var}(y_t \mid \mathcal{F}_{t-1})$, where $\mathcal{F}_{t-1}$ contains a set of available information up to time $t-1$, which, in our case, is the past returns. A local volatility model, assuming Gaussian errors, is specified as

$$y_t = \sigma_t \epsilon_t, \quad \epsilon_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1), \tag{2.1a}$$

$$\sigma_t = f\left(y_{1:t-1}\right), \quad t = 1, 2, \ldots, T_{in}, \tag{2.1b}$$

$$\ell(\boldsymbol{y} \mid \boldsymbol{\theta}) = -\frac{1}{T_{in}} \sum_{t=1}^{T_{in}} \log\left(p\left(y_t \mid \sigma_t\right)\right). \tag{2.1c}$$

Here, T_{in} represents the in-sample data size. The function $f(\cdot)$ forecasts the conditional volatility σ_t, and it could be an econometric model (e.g., GARCH) or a NN model. The negative log-likelihood $\ell(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the objective to minimize, where $\boldsymbol{\theta}$ denotes all the model parameters, and $p(\cdot \mid \sigma_t)$ is a Gaussian density function with mean zero and variance σ_t^2.

Local training means estimating the model parameters using a single time series $\boldsymbol{y}$. The resulting model is called a *local model*, as it is trained independently for each asset. For example, if $f(\cdot)$ is based on a GARCH(1,1) model, i.e. $f\left(y_{1:t-1}\right) = \left(\omega + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2\right)^{1/2}$, then the parameters $\boldsymbol{\theta} = (\omega, \alpha, \beta)$ are optimized specifically for each series $\boldsymbol{y}$.

Global training. Global training uses several time series to estimate a shared set of parameters, forming a global model. We write this approach as

$$y_t^n = \sigma_t^n \epsilon_t^n, \quad \epsilon_t^n \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1), \quad t = 1, 2, \ldots, T_{in}, \quad n = 1, 2, \ldots, N, \tag{2.2a}$$

$$\sigma_t^n = f^*(y_{1:t-1}^n), \tag{2.2b}$$

$$\ell(\boldsymbol{Y} \mid \boldsymbol{\theta}^*) = -\frac{1}{NT_{in}} \sum_{n=1}^{N} \sum_{t=1}^{T_{in}} \log\left(p(y_t^n \mid \sigma_t^n)\right). \tag{2.2c}$$

Here, a single model $f^*(\cdot)$ is trained across all series, optimizing parameters $\boldsymbol{\theta}^*$ collectively. For example, using global training to fit a GARCH(1,1) model on 10,000 stock series results in a single parameter $\boldsymbol{\theta}^* = (\omega^*, \alpha^*, \beta^*)$, optimized over the entire 10,000 time series dataset.

2.3.2 Model Selection

Our objective is to investigate the general characteristics of econometric and NN models under local and global training regimes rather than to conduct a model comparison exercise. Therefore, following Kaplan et al. (2020) and Zhai et al. (2022), our model selection focuses on representative foundational architectures in financial time series analysis.

For econometric benchmarks, we employ the GARCH (Bollerslev, 1986), GJR (Glosten et al., 1993), and EGARCH (Nelson, 1991) models. Despite their simplicity, these models are canonical for modeling conditional volatility from daily returns and provide a robust baseline for capturing key stylized facts, such as volatility clustering and the leverage effect. We exclude HAR-type models (Corsi, 2009) because our study requires a large dataset while high-frequency data are not uniformly available at such a scale.

For the NNs, we select four of the most widely used models for sequential data analysis: Recurrent Neural Network (RNN) of Elman (1990), the Long Short-Term Memory (LSTM) of Hochreiter and Schmidhuber (1997), the Gated Recurrent Unit (GRU) of Chung et al. (2014), and the attention-based Transformer of Vaswani et al. (2017). We focus on establishing broad principles from these foundational models and are confident that the insights will apply to their architectural extensions. A detailed technical exposition of each NN architecture, especially the Transformer, is beyond the scope of this study. In what follows, we provide a brief overview of the RNN-based models and refer readers to Vaswani et al. (2017) for a full treatment of the Transformer.

RNN

RNNs process sequential information by maintaining a latent state vector that acts as a memory, capturing temporal dependencies. This state is updated recursively at each time step, incorporating both the current input and the previous state. In our context of volatility forecasting using stock returns (y_t) as input, a basic RNN structure updates

its hidden state h_t and predicts the conditional volatility σ_{t+1} as follows:

$$h_t = \psi\left(W_y y_t + W_h h_{t-1} + b_h\right) \tag{2.3a}$$

$$\sigma_{t+1} = \mathrm{ReLU}\left(W_\sigma h_t + b_\sigma\right) + 10^{-8}, \tag{2.3b}$$

where $h_t \in \mathbb{R}^{d_h}$ is the hidden state vector of dimension d_h at time t, and y_t is the input return. The matrices W_y, W_h, W_σ and bias vectors b_h, b_σ contain the model parameters, estimated during training. The function ψ represents a non-linear activation function, often the hyperbolic tangent (tanh) or sigmoid function, enabling the model to capture non-linear dynamics. The Rectified Linear Unit activation, $\mathrm{ReLU}(x) = \max\{x, 0\}$, in the output layer ensures non-negative volatility predictions, with a small constant (10^{-8}) added for numerical stability. Figure 2.1 provides a graphical illustration.

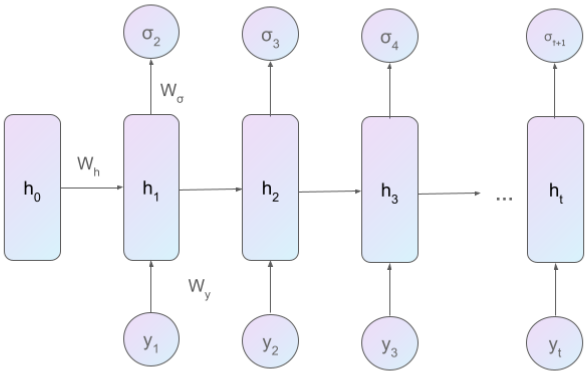


Figure 2.1: Graphical representation of the basic RNN.

A significant limitation of this basic RNN formulation is the potential for vanishing or exploding gradients during the backpropagation phase of training. As gradients are propagated backward through time, they can diminish exponentially (vanish) or grow uncontrollably (explode), hindering the network's ability to learn long-range dependencies effectively. This issue is particularly relevant for financial time series, which often exhibit persistent dynamics.

LSTM

To address the gradient flow problems in basic RNNs, Hochreiter and Schmidhuber (1997) introduced the Long Short-Term Memory (LSTM) network. LSTMs incorporate a more sophisticated recurrent unit structure, featuring a dedicated memory cell (C_t) and gating mechanisms—input (i_t), forget (f_t), and output (o_t) gates. These gates regulate the flow of information into, out of, and within the memory cell, allowing the network to selectively retain or discard information over long periods. The core LSTM equations are:

$$f_t = \psi\left(W_{fy} y_t + W_{fh} h_{t-1} + b_f\right) \tag{2.4a}$$

$$i_t = \psi\left(W_{iy} y_t + W_{ih} h_{t-1} + b_i\right) \tag{2.4b}$$

$$o_t = \psi\left(W_{oy} y_t + W_{oh} h_{t-1} + b_o\right) \tag{2.4c}$$

$$\tilde{C}_t = \tanh\left(W_{Cy} y_t + W_{Ch} h_{t-1} + b_C\right) \tag{2.4d}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.4e}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{2.4f}$$

$$\sigma_{t+1} = \mathrm{ReLU}\left(W_{\sigma h} h_t + b_\sigma\right) + 10^{-8}. \tag{2.4g}$$

Here, ψ is typically the sigmoid function, $\odot$ denotes element-wise multiplication, and $\tilde{C}_t$ represents a candidate cell state. The additive interaction in Equation 2.4e provides a more stable path for gradient propagation compared to the basic RNN structure. The gating mechanisms allow LSTMs to capture complex temporal patterns, including long-range dependencies and non-stationarities often observed in financial markets, making them suitable candidates for our investigation. We also consider the GRU model (Chung et al., 2014), a computationally simpler variant of the LSTM, detailed in Appendix A.1.

Transformer

Beyond recurrent architectures, the Transformer model utilizes attention mechanisms, eliminating recurrence entirely. By processing input sequences in parallel and employing self-attention to weigh the importance of different time steps, Transformers can po-

tentially model long-range dependencies more effectively than RNNs. Its non-recurrent nature offers a distinct approach to capturing temporal patterns, and its demonstrated success in various domains, including time series forecasting (Wu et al., 2021; Zhou et al., 2021), warrants its consideration, particularly when analyzing large datasets where parallel processing capabilities are advantageous. Due to the extensive technical details of the Transformer's internal components such as multi-head self-attention, positional encodings, and layer normalization, covering those falls outside the scope of this chapter. We refer the interested reader to Vaswani et al. (2017) for a detailed technical treatment of the Transformer architecture.

2.3.3 Evaluation metrics

This section introduces the evaluation metrics employed in our study. A comprehensive evaluation framework is essential to assess the performance of volatility models from both statistical and economic perspectives. To achieve this, we employ the metrics that measure the accuracy of volatility forecasts, the effectiveness in risk management applications, and the statistical significance of model performance at the individual stock level. By combining these metrics, we aim to provide a robust assessment of the models' predictive abilities, economic utility, and practical relevance in financial applications.

To assess model fit and the quality of the full predictive distribution, we employ the Negative Log-Likelihood (NLL), also known as the Partial Predictive Score. The NLL is a proper scoring rule that directly corresponds to the likelihood objective of parametric volatility models and thus facilitates fair comparison across different model classes. Although alternative distributional scoring rules—such as the Continuous Ranked Probability Score (CRPS) or the Brier score—are available in other domains, NLL is particularly well-established in financial econometrics for volatility forecasting and easily accommodates both Gaussian and heavy-tailed error specifications. The NLL is defined as:

$$\mathrm{NLL} \;:=\; -\frac{1}{T_{\text{test}}} \sum_{t \in D_{\text{test}}} \log p\big(y_t \mid y_{1:t-1}, \widehat{\theta}\big), \tag{2.5}$$

where $\widehat{\theta}$ denotes the estimated model parameters and $p(\cdot)$ the assumed return density.

Economic utility: In addition to statistical fitness, we assess the economic utility of volatility models in risk management, a crucial application of volatility modeling. Specifically, we focus on forecasting VaR and ES. These metrics are mandated by regulatory frameworks such as the Basel Accord and capture tail-risk characteristics beyond the central moments of the distribution. The α-level VaR represents the α quantile of the return distribution, while the α-level ES corresponds to the conditional expectation of returns exceeding the corresponding VaR. To evaluate the accuracy of VaR forecasts, we use the quantile loss function (Koenker & Bassett, 1978)

$$\text{Qloss}_\alpha := \frac{1}{T_{\text{test}}} \sum_{y_t \in D_{test}} \left(\alpha - I\left(y_t < Q_t^\alpha\right)\right)\left(y_t - Q_t^\alpha\right), \tag{2.6}$$

where Q_t^α is the forecast α-level VaR at time t. The quantile loss function is strictly consistent (Fissler & Ziegel, 2016), meaning the expected loss is minimized when the forecast accurately predicts the true quantile. The model with the lowest quantile loss is therefore preferred for VaR forecasting.

While no strictly consistent loss function exists for ES in isolation, Fissler and Ziegel (2016) demonstrates that ES and VaR are jointly elicitable, allowing for their joint evaluation using specific loss functions. One such function, based on the Asymmetric Laplace (AL) distribution, is strictly consistent for jointly assessing VaR and ES (Taylor, 2019). The AL-based joint loss function is defined as:

$$\text{JointLoss}_\alpha := \frac{1}{T_{\text{test}}} \sum_{y_t \in D_{test}} \left(-\log\left(\frac{\alpha - 1}{\text{ES}_t^\alpha}\right) - \frac{\left(y_t - Q_t^\alpha\right)\left(\alpha - I\left(y_t \leq Q_t^\alpha\right)\right)}{\alpha \text{ES}_t^\alpha}\right), \tag{2.7}$$

where ES_t^α is the forecast α-level ES at time t. Consistent with econometric practice in risk management, we report both quantile loss and joint loss at the 1% and 2.5% levels.

Statistical significance: Beyond assessing aggregate performance across stocks, we also evaluate the global model performance at the individual stock level and to determine

whether observed differences in predictive performance are robust and not driven by sampling variability, we implement the Model Confidence Set (MCS) by Hansen et al. (2011). Unlike pairwise comparison tests, the MCS simultaneously evaluates all candidate models while controlling for the multiplicity of tests, thereby maintaining the family-wise error rate without relying on conservative Bonferroni corrections. The MCS identifies a Superior Set of Models (SSM), defined as the subset of models that demonstrate equal predictive accuracy based on sequential hypothesis testing. Let $\mathcal{M}$ denote the set of competing models. For models i and j in $\mathcal{M}$, the relative loss is defined as $d_{i,j,t} = L_{i,t} - L_{j,t}$, where $L_{i,t}$ represents the performance loss of model i at time t. The MCS tests the null hypothesis:

$$H_0 : \mu_{i,j} = 0, \quad \text{for all } i, j \in \mathcal{M}, \tag{2.8}$$

where $\mu_{i,j} = \mathbb{E}(d_{i,j,t})$ is the expected relative loss. A model is excluded from the SSM if the null hypothesis of equal predictive accuracy is rejected. The remaining models in the SSM are those for which H_0 cannot be rejected. The MCS assigns a p-value p_i to each model $i \in \mathcal{M}$, with higher p-values indicating a higher likelihood of inclusion in the SSM. The interested reader is referred to Hansen et al. (2011) for more details.

2.4 Scaling Effect

This section investigates how data size, model complexity, and model architecture choice affect the performance of global models. To this end, we train multiple global GARCH and NNs by systematically varying the number of stock series in the training set from 10 to 10,240, doubling at each increment, while keeping all other model hyperparameters and training configurations constant (see Appendix A.4 for details of training hyperparameters). For each stock series, only its in-sample period contributes to model training. All reported results, unless specified otherwise, pertain to one-day-ahead forecasts evaluated on the out-of-sample testing period. We use out-of-sample NLL as the primary predictive score in this section; other predictive scores more relevant to risk management are examined in Section 2.5.

We evaluate predictive accuracy under two distinct scenarios:

1. **Supervised Forecasting:** The model predicts the out-of-sample volatility for stock series whose in-sample data were part of its training set.

2. **Zero-Shot Forecasting:** The model predicts out-of-sample volatility for unseen stock series, meaning their in-sample data were entirely excluded from the training process.

Our dataset comprises 11,771 stock series. From these, 10,240 series (the training series) are used for training the global models and for supervised evaluation. The remaining 1,531 series (the unseen series) are exclusively reserved for evaluating zero-shot forecasting performance. To ensure a consistent benchmark for supervised forecasting across different data scales, performance for this scenario is evaluated on the same initial 10 stock series that constitute the smallest training set. This approach avoids conflating supervised performance with zero-shot capabilities, which would occur if a model trained on a small subset of series were evaluated across all 10,240 training series.

2.4.1 Local GARCH versus Local NNs

We first study the performance of NNs and the econometrics models when the local training approach is used. Our aim is to address the ongoing debate within the econometrics community regarding the effectiveness of NN models in financial time series forecasting (Makridakis et al., 2018). For NN models, we choose basic RNN, LSTM and GRU. Inspired by Hansen and Lunde (2005) who decisively confirm the performance of GARCH(1,1), we choose it to represent the econometrics models; other econometric volatility models are considered in Section 2.5.

Table 2.2 reports the average out-of-sample NLL for local NNs, along with their respective win rates against the GARCH(1,1) model across 11,771 stocks. Additional results for risk management metrics and other econometric baseline models are presented in Table 2.5 and 2.6. The results indicate that even the best-performing NN architecture, the LSTM, outperforms the GARCH(1,1) model in only 32% of the stocks and on

average underperform compared to the GARCH(1,1) model. This confirms the finding in Makridakis et al. (2018) that ML models often do not outperform simple statistical models in financial time series forecasting.

Table 2.2: Performance comparison of local models

This table presents a performance comparison between various neural network (NN) models and the GARCH(1,1) baseline across 11,771 individual stock series in the local training setting. The NLL column reports the out-of-sample average negative log-likelihood (NLL) across all stocks. The Win Rate column shows the percentage of stocks for which a given NN model outperforms the GARCH(1,1) baseline.

	NLL	Win Rate
GARCH	2.261	-
RNN	2.273	17%
GRU	2.269	29%
LSTM	2.266	32%
Transformer	2.267	30%

These results stand in stark contrast to the widespread success of NNs in other fields. Here, we attempt to explain this discrepancy and, in subsequent sections, propose to correct this perception through the use of global training. Financial time series are traditionally treated as heterogeneous, leading to the use of local training approaches where models are fitted individually for each series. As a result, this approach only makes use of limited data— typically around several thousand observations per stock — which is insufficient for effectively training NNs, as they generally require large datasets to learn generalized patterns. In contrast, econometric models excel in financial time series forecasting even with limited data because they incorporate theoretical patterns into their structure. For instance, the GARCH model is designed to capture the clustering effect, while GJR (Glosten et al., 1993) and EGARCH (Nelson, 1991) models are additionally designed to capture the leverage effect. These structural assumptions make econometric models particularly well-suited for data-scarce environments, offering a level of robustness that is challenging for black-box neural networks to achieve under similar conditions.

2.4.2 Global GARCH

However, when econometric models are applied as global models, their performance characteristics change considerably. Figure 2.2 plots the data-size scaling effect of the global GARCH model and the local baseline. Even when the global GARCH model is trained on only 10 stock series, it performs substantially worse than its local counterparts. This decline in performance arises from the global GARCH model's inability to capture the increased heterogeneity across different stocks.

When applied to a single stock, a GARCH-type model is well-suited to capture its unique volatility dynamics, since parameter estimates can specialize to the features of that individual series. But when one attempts a global GARCH, that is, one model with a common parameter set for many stocks, serious problems arise: namely, stocks differ substantially in volatility behaviour owing to differences in industry, market conditions, and idiosyncratic features, yet a global GARCH imposes that all series share parameters governing unconditional variance (e.g.), which is implausibly rigid. This shared-parameter constraint severely limits the model's flexibility, and likely accounts for its degraded performance as the number of series increases.

One potential approach to address this limitation could involve coupling a random-effects framework with GARCH, where the common fixed-effect structure is retained across stocks, but random-effect parameters allow for stock-specific variations. However, we do not pursue this idea further in this chapter.

As the number of stock series used for training increases, the global GARCH model shifts from capturing stock-specific idiosyncrasies to representing the aggregate characteristics of the pooled dataset. In supervised settings, this transition leads to a decline in forecast accuracy for the 10 training stocks, as the model's ability to reflect unique stock-level features diminishes. Conversely, in zero-shot settings—where the model forecasts stock series not included in the training set—the global model achieves improving performance by leveraging the aggregate characteristics of the broader dataset.

Table 2.3 presents the estimated parameters for 11 global GARCH models with varying data sizes, demonstrating the data scaling effect. The parameter estimates stabilize

once the model is trained on more than 160 stock series, marking a clear transition from modeling stock-specific features to capturing general patterns shared across the dataset. Beyond this point, the global GARCH model predominantly reflects the aggregate dynamics of all the stock series rather than the unique characteristics of individual stocks.

Table 2.3: Estimated parameters of global GARCH models by number of training series

NSeries	ω	α	β
10	0.109	0.054	0.931
20	0.041	0.050	0.945
40	0.051	0.053	0.943
80	0.019	0.034	0.964
160	0.014	0.029	0.968
320	0.015	0.030	0.969
640	0.014	0.029	0.970
1280	0.013	0.029	0.970
2560	0.015	0.030	0.968
5120	0.014	0.030	0.968
10240	0.015	0.030	0.969

In conclusion, while the structured restrictions imposed by econometric models are beneficial in local training settings, these restrictions and the lack of model complexity limit their performance in global settings. This result underscores the limitations of traditional econometric models in generalizing across diverse time series and highlights the need for more complex models capable of effectively handling the heterogeneity present in financial data. The following section examines the performance of global NNs.

2.4.3 Global NNs

Data Scaling Effect and Model Arichitecture

Unlike econometric models, NNs perform particularly well in data-rich environments due to their flexible structure and expressive power, which allows them to capture complex relationships in large and diverse datasets. In this section, we demonstrate the performance of global NNs and examine their data scaling effect.

To assess the relative importance of data scaling effect and model architectures, we

study four NN architectures: RNN, GRU, LSTM, Transformer. Figure 2.3 shows the performance of global NNs and local NNs for both supervised and zero-shot forecasting. Unlike global econometric models, global NN models quickly outperform their local counterparts once the number of training series reaches 40. The contrasting performance of global GARCH and global NNs highlights a fundamental difference in their nature: econometric models rely on patterns imposed through prior knowledge, whereas NNs leverage their flexible structure and expressive power to uncover such patterns directly from the data. However, the flexibility of NNs comes with a requirement for substantial data to avoid overfitting noise or idiosyncratic patterns that do not generalize to out-of-sample periods. When trained on individual stock series, local NNs are particularly prone to this overfitting, which degrades their out-of-sample performance, making them inferior to simpler, well-constrained econometric models like GARCH. By pooling data from multiple stocks, global NNs overcome this limitation by learning shared temporal dynamics and capturing broader market trends that local models often miss. The increased diversity and volume of pooled data enable global models to form a more robust representation of market behavior, accommodating volatility fluctuations and other stochastic factors. Consequently, predictive accuracy improves significantly as the number of series included in training increases.

Among the evaluated NN architectures, basic RNNs demonstrated notably weaker performance compared to their more advanced counterparts. This underperformance is likely attributable to their susceptibility to the vanishing gradient problem and limited capacity to capture long-range dependencies effectively. Interestingly, we did not observe state-of-the-art Transformer models exhibiting a clear performance advantage over LSTMs in our specific application. This finding may be explained by several factors. First, the primary strength of Transformers often lies in their computational efficiency for parallel processing, which allows them to scale to extremely large datasets and model sizes, such as those with billions of parameters seen in modern large language models. While our dataset is substantial, comprising approximately 12,000 stock series, it still should be considered relatively small for leveraging the full scaling benefits of Transform-

ers, and computational throughput was not the bottleneck for RNN-type architectures in this study. Second, in the context of univariate time series forecasting, where the input is one-dimensional, the sophisticated embedding layers and self-attention mechanisms integral to Transformers might offer less distinct advantages than they do in higher-dimensional sequence processing tasks.

While the variation in NN architectures results in differences in forecast accuracy, the data scaling effect consistently holds. Notably, in terms of NN performance, the scale of the data plays a far more critical role than the choice of model architecture.

The results also demonstrate a critical benefit of global NNs in a zero-shot setting, where the models predict stock series not included in the training set. The global models consistently outperform local models trained directly on the target stocks. This result underscores two important insights: first, that stock series exhibit significant similarities in volatility and temporal patterns, which supports the view that they can be treated as homogeneous for modeling purposes; and second, that NNs can effectively capture these shared patterns, enabling better generalization to unseen data.

These data scaling effects are not unique to financial applications. Similar effects are well-documented in other domains, such as natural language processing (Kaplan et al., 2020; Zhai et al., 2022), where NNs trained on large and diverse text corpora demonstrate superior performance and generalization compared to those trained on smaller, domain-specific corpora. This phenomenon is evident in the success of large language models like ChatGPT. The same principles should apply to financial time series. Consider stocks A and B. Suppose a specific pattern appears in the in-sample period of stock A and the out-of-sample period of stock B. If separate local models are trained for stocks A and B, the model for stock B would fail to recognize this pattern during out-of-sample forecasting. By contrast, a global model trained on pooled data from both stocks can learn the pattern from stock A and leverage it to improve predictions for stock B.

Despite this, stock series in finance are often modeled individually as heterogeneous, which limits the data available for NNs to identify robust and generalizable patterns. This practice is fundamentally at odds with the nature of NN models and underutilizes

their strengths. We argue that, if one intends to treat stocks as heterogeneous series and train models locally, NN models may not be a suitable approach. Conversely, if NN is to be applied to financial time series, a global approach—pooling data from multiple stocks—should be considered as the standard.

Model Scaling Effect

This section examines the effect of model complexity on predictive performance. To do so, we train a series of global LSTM models on a fixed dataset of 10,240 stocks while incrementally increasing the number of hidden units from one to fifteen. We vary complexity by adding hidden units rather than scaling the parameter count by a factor of 10 such as in (e.g. Kaplan et al., 2020; Zhai et al., 2022). This granular approach is necessary because, as we demonstrate below, the optimal model size for financial forecasting is relatively small, and aggressive scaling would obscure these effects. We then evaluate the predictive accuracy of each model on both the training series (supervised forecasts) and a holdout set of 1,531 unseen series (zero-shot forecasts).

Figure 2.4 plots the out-of-sample performance against model size. Initially, predictive accuracy improves with complexity. Smaller models, akin to a global GARCH, lack the capacity to capture the heterogeneity within the pooled data, leading to suboptimal performance. However, this improvement quickly plateaus. We find that performance peaks at a modest model size of only five hidden units (approximately 200 parameters), a point we term the "bottleneck model size." Beyond this threshold, additional complexity provides no marginal benefit. This finding contrasts with the scaling laws observed in other domains, which is likely attributed to the smaller data size and low signal-to-noise ratio inherent in financial time series.

Data Diversity Effect

Stocks from the same country or industry sector are expected to exhibit higher levels of homogeneity, raising the question of whether grouping such stocks for modeling enhances forecast accuracy. To investigate this, the 11,771 stocks are divided into groups based

on their country of origin and industry classification. Separate group-specific models are trained for each group, meaning a global model is trained using only stocks from that group. These group-specific models are then compared to global models trained on more diverse datasets. To ensure a fair comparison and eliminate the effect of data size, the global models trained on diverse datasets are matched to the size of the corresponding stock group. For example, for the China stock group containing 3,469 series, the corresponding global model is trained on 3,469 diversified stocks randomly selected from the 11,771 stocks.

Table 2.4 reports the performance comparison. The results indicate that the global model, trained on a similarly sized but more diverse dataset, generally outperforms the group-specific models. This superior performance can be attributed to the global model's exposure to a more diverse dataset, enabling it to capture patterns shared across individual segments. These findings underscore the importance of data diversity as a solution to the data size bottleneck. Future research could be conducted by combining financial time series across different asset classes. Moreover, the results again highlight that, when employing flexible models such as neural networks, stock series should be effectively treated as homogeneous and modeled together for optimal performance.

Table 2.4: LSTM Model performance by country and industry

	Group Model	Global Model	NStocks
Country			
China (mainland)	2.291	**2.287**	3469
United States	2.319	**2.310**	2146
Japan	**1.950**	1.951	1988
India	2.310	**2.301**	834
United Kingdom	2.157	**2.146**	573
Industry			
Industrials	2.181	**2.179**	2459
Consumer Discretionary	2.237	**2.229**	1633
Information Technology	2.337	**2.321**	1550
Materials	2.207	**2.201**	1396
Financials	2.215	**2.169**	1202

2.5 Universal Volatility Model

The previous section examined the data scaling effects of NNs, and confirmed the superior performance of global NNs for financial volatility modelling. This section explores the economic utility and statistical significance of a globally trained stock volatility model for real world financial applications, along with a detailed interpretation of its characteristics. This global model employs an LSTM architecture trained on a dataset of 10,240 stock series. We refer to this trained global LSTM model as the *universal volatility model*.

2.5.1 Data Scarcity and Temporal Importance

Data scarcity is a persistent challenge in time series modeling, especially in business and economic applications. For example, data sets such as annual GDP series or daily stock returns often contain at most only a few thousand observations, which is considered small in successful machine learning applications. A related challenge arises with newly listed stocks which have insufficient historical data for a meaningful modeling. Traditional econometric models, such as GARCH, typically require at least two thousand observations for reliable parameter estimation (Nguyen et al., 2022), making them less effective in such scenarios. In contrast, the ability to produce zero-shot forecasts of global models allows them to overcome data scarcity, enabling accurate volatility forecasts for newly listed stocks with fewer observations.

To investigate the performance of universal volatility model in such data scarce scenarios, we analyze the impact of input series length on its forecast accuracy. We adopt the remove-and-predict method, originally developed for computer vision tasks (Samek et al., 2017), in which pixels or regions are systematically removed from input images to identify the features that contribute the most to the prediction accuracy of a trained model. We extend this idea to assess the importance of the input length in stock series data. First, we evaluate the universal model using a rolling window forecasting scheme with a fixed window size of 504 observations (equivalent to two trading year). That is, the model uses the past 504 return observations to make one day ahead volatility forecasts,

and we compute the negative log-likelihood NLL for all the unseen stocks. Next, to study the effect of input length, we systematically reduce the input window size and recalculate the NLL to observe how changes in input length influence model performance. We define the temporal importance (TI) of size k as

$$\mathrm{TI}_k = 100\frac{\mathrm{NLL}_{504} - \mathrm{NLL}_k}{\mathrm{NLL}_{504}}, \tag{2.9}$$

where NLL_k is the NLL of the universal model evaluated with a fixed window size of k on the unseen stocks. The value of TI_k indicates the relative performance of the input window of size k compared to the 504 observation window. This approach allows us to quantify the contribution of various past observations to the prediction accuracy of the universal model. Figure 2.5 plots the temporal importance of the universal LSTM model for different window sizes k. The results indicate that the most critical observations are typically concentrated within the past four months, with the importance of historical observations decreasing almost exponentially as they become more distant. Observations from more than twelve months prior have negligible importance and almost no impact on forecast accuracy.

These findings underscore a practical advantage of universal volatility models in real-time applications for rapidly evolving financial markets. Unlike local econometric models, which generally require eight years of daily observations for reliable parameter estimation and forecasting, once trained, global NNs can directly provide accurate volatility forecasts using as little as twelve months of data, even for stocks not included in the training set. This ability is particularly valuable for newly listed stocks, where historical data is inherently limited and insufficient to fit local models effectively.

2.5.2 Financial Risk Forecasts

We now assess the economic utility of the universal LSTM model in risk management, using VaR and ES as the risk metrics. All local baselines—including local GARCH, local GJR, local EGARCH, and local LSTM—are evaluated using an expanding window

forecasting scheme, leveraging the full historical observations starting from 01/01/2014. In contrast, drawing on insights from the temporal importance analysis in Section 2.5.1, the universal LSTM is evaluated using a rolling window forecasting scheme, which relies on only the past twelve months (252 days) of observations.

Table 2.5 reports the average risk metrics for the models under consideration. Consistent with the earlier findings, the local LSTM model underperforms the econometric baselines on average. However, the Model Confidence Set results in Table 2.6 reveal a more nuanced picture: the local LSTM is more frequently included in the Superior Set of Models (SSM) compared to the GARCH and GJR models. This indicates that the performance of the local LSTM model is highly stock dependent: while it can achieve strong results for certain stocks, it may perform poorly for others. In contrast, econometric models exhibit more consistent performance across stocks.

The dual behavior of the local LSTM—lower average predictive accuracy but higher inclusion in the SSM when it performs well—helps explain why, despite skepticism within the econometrics community, neural network models are often reported to be superior in some studies. The existing literature tends to focus primarily on cases where machine learning models perform well, potentially overlooking their inconsistent results across broader contexts.

Table 2.5: Risk metrics of the universal LSTM model and local baselines

This table reports the risk metrics for the universal LSTM model and local baseline models, averaged across all 11,771 stocks. For all metrics, lower values indicate better performance.

	GARCH	GJR	EGARCH	LSTM	Universal LSTM
NLL	2.259	2.253	2.247	2.278	**2.229**
QLoss 1%	0.102	0.096	0.094	0.113	**0.082**
QLoss 2.5%	0.193	0.185	0.182	0.201	**0.157**
JointLoss 1%	3.291	3.278	3.269	3.385	**3.060**
JointLoss 2.5%	3.002	2.987	2.973	3.083	**2.795**

The universal LSTM, even using only the past twelve months of observations, demonstrates consistent and statistically significant improvements in out-of-sample performance for risk management applications compared to the local LSTM and econometric baselines.

Table 2.6: Model Confidence Set (MCS) of the universal LSTM model and local baselines

This table reports the frequency with which each model is included in the set of superior models (SSM) at the 5% significance level. The numbers in parentheses are the average p-values across all 11,771 stocks. Higher p-values indicate a greater likelihood of inclusion in the SSM.

	GARCH	GJR	EGARCH	LSTM	Universal LSTM
NLL	2023	1868	3105	2773	**8794**
	(0.106)	(0.095)	(0.187)	(0.170)	**(0.685)**
QLoss 1%	2493	2087	3489	2621	**8541**
	(0.134)	(0.109)	(0.210)	(0.153)	**(0.661)**
QLoss 2.5%	2398	1989	3373	2842	**8443**
	(0.133)	(0.103)	(0.209)	(0.171)	**(0.654)**
JointLoss 1%	2534	2190	3440	2419	**8636**
	(0.140)	(0.113)	(0.208)	(0.139)	**(0.670)**
JointLoss 2.5%	2395	2046	3311	2546	**8887**
	(0.131)	(0.104)	(0.196)	(0.145)	**(0.693)**

At the individual stock level, as reported in Table 2.6, the universal model also performs robustly, being included in the SSM for the majority of stocks and achieving significantly higher p-values. Notably, in our experiments, all econometric models are re-estimated daily during out-of-sample periods, while the global LSTM does not undergo any retraining, demonstrating its robustness to distribution shifts.

Training 10,240 local LSTM models requires 1,382 minutes, compared with five minutes for a single global model on the same series and hyperparameters. For local models, computation scales linearly with the number of series, and the operational burden is amplified by the need to manage and tune thousands of models. In large-scale financial applications requiring frequent recalibration, this renders local neural networks impractical. A single global model offers a scalable alternative, delivering consistent volatility forecasts with minimal training time.

2.5.3 Model Interpretation

This section evaluates the universal LSTM model in more detail, providing additional interpretation and insights to better understand its behavior and performance.

Leverage Effect

The leverage effect is a key stylized fact of stock volatility, describing the phenomenon where negative shocks to a stock's return increase its volatility more than positive shocks of the same magnitude. This asymmetry is explicitly incorporated into the structure of GJR and EGARCH models. This section investigates whether global NNs can also capture the leverage effect.

The leverage effect is commonly measured by the news impact curve (NIC) (R. F. Engle & Ng, 1993), which illustrates how unexpected returns influence the volatility of an asset. For the GARCH family models such as the GJR, the NIC can be derived analytically due to their simple functional forms, enabling a direct representation of the relationship between shocks and volatility changes. In contrast, NNs lack such predefined functional forms, making it impossible to directly derive the NIC. To overcome this challenge, we adopt a simulation-based approach to approximate the NIC for NNs. Specifically, we provide the model with an input sequence of length 252 (representing the past 12 months of inputs) consisting entirely of zeros, except for the last observation, which is varied from -5 to 5. This allows us to examine how the model's output, which is the conditional volatility, responds to these changes.

Figure 2.6 plots the NIC for the universal LSTM model. Despite lacking a predefined structure, the model effectively captures the leverage effect. It produces an asymmetric response of volatility to positive and negative shocks, with negative shocks causing a more pronounced increase in volatility compared to positive shocks of the same magnitude. The NIC derived from the model exhibits a sharp, nonlinear rise in volatility in response to adverse news, closely aligning with empirical patterns observed in financial markets, where declines in asset prices amplify financial risk due to higher debt-equity ratios. The result again highlight the fundamental distinction between econometric and NN models: the former rely on structured restrictions derived from theoretical patterns, while NN models uncover such patterns directly from rich data environments.

Characteristics of the Universal Model

We now closely examine the forecasts made by the universal model for several individual stocks, comparing their characteristics to those of the forecasts produced by econometric models. Figure 2.7 compares the 1% ES forecasts generated by the GARCH model and the universal LSTM model for the top three market-cap companies—Nvidia, Apple, and Microsoft—during the out-of-sample period, along with their respective joint losses for the ES forecasts. The results demonstrate that the universal model achieves improved joint loss compared to local GARCH models, while exhibiting markedly different forecast characteristics. The universal model generally produces more conservative forecasts, i.e., more negative ES values, during relatively stable periods when stock returns experience minimal sudden jumps (e.g., for Apple from 01-01-2023 to 01-04-2023). In contrast, during periods of volatile market conditions with sudden jumps in returns, the universal model produces forecasts that are much less extreme and recovers from outliers much faster (e.g., the high shock in Nvidia stock around 2023-06).

To further illustrate this difference, we artificially added a single high shock in Apple stock returns and analyzed the reaction of GARCH and universal LSTM models. Figure 2.8 plots the ES forecasts during the out-of-sample period, with and without the added outlier. The results show that the universal model responds much more moderately to the outlier, a behavior attributed to its training on a large, diverse dataset of stocks, which significantly enhances its robustness to extreme values. Additionally, the universal model recovers from outliers significantly faster than GARCH models. In this example, the underlying volatility dynamics did not change after the shock. However, the GARCH model generated an extended period of high-volatility forecasts due to the shock, a limitation inherent to the structure of GARCH-family models, where shocks can only have an additive effect. Even if stock returns immediately return to zero after a high shock, the GARCH-family model can only reduce volatility forecasts incrementally at a rate determined by the β parameter, leading to slower recovery. In contrast, the universal NN model, unconstrained by such structural limitations, can adjust variance instantaneously while still capturing volatility clustering, allowing them to return to normal volatility

forecasts much faster and adapt quickly to rapidly shifting market conditions.

We further examine the performance of the universal LSTM model during high and low volatility years. Table 2.7 presents the percentage improvement in NLL of the universal model relative to GARCH models for each year, along with the average standard deviations of stock returns in that year (higher average standard deviations indicating more volatile years). The results demonstrate that the universal model outperforms the local GARCH models more during periods of pronounced market turbulence.

Table 2.7: Model performance by year

This table reports the annual performance of the universal LSTM model relative to local GARCH models. The middle column shows the percentage improvement in negative log-likelihood (NLL) of the universal LSTM over the local GARCH models. The last column presents the average standard deviation of returns for all stocks in each year, with higher values indicating more volatile years. Both metrics are calculated across all 11,771 stocks. Note: 2020 and 2021 are validation periods, while 2022 and 2023 are testing periods.

Years	Average NLL Improvement (%)	Average Std of Returns
2020	2.362	3.266
2021	1.546	2.452
2022	1.698	2.578
2023	1.030	2.149

This robustness of the universal model ensures consistent performance across diverse market conditions. By dynamically adapting and recovering quickly from market shocks, the universal model enhances reliability for risk management and decision-making, offering an attractive alternative to econometric models.

We also include a residual analysis in Appendix A.3, showing that the standardized residuals of the universal model display characteristics consistent with a well-specified GARCH model, including uncorrelated but non-normally distributed series.

2.5.4 Portfolio Risk Forecasts

Portfolio risk forecasting is a critical function in the financial industry. This section evaluates the performance of the universal model in this context. To conduct a large-scale test and avoid the potential biases of specific portfolio construction methods, we

generated 10,000 fully randomized portfolios using the following procedure:

- Randomly select the size of the portfolio M, where $M \sim \text{Uniform}(10, 50)$.
- Randomly select M stocks from the pool of 11,771 available stocks.
- Assign random positive weights summing to 1 to the M selected stocks.
- Compute the weighted sum of the M stocks to construct an artificial portfolio.

The universal LSTM model is then evaluated on these 10,000 portfolios. We note that the universal model is not retrained for these portfolios, i.e. it provides zero-shot forecasts for these portfolios.

Table 2.8: Risk metrics for the local and universal models

This table reports risk metrics for the local and universal models, averaged over all 10,000 portfolios. For all metrics, lower values indicate better performance.

	GARCH	GJR	EGARCH	LSTM	Universal LSTM
NLL	1.631	1.625	1.622	1.645	**1.614**
QLoss 1%	0.068	0.064	0.062	0.074	**0.054**
QLoss 2.5%	0.128	0.123	0.121	0.133	**0.105**
JointLoss 1%	2.307	2.298	2.289	2.351	**2.117**
JointLoss 2.5%	2.102	2.087	2.075	2.143	**1.960**

Table 2.9: Model Confidence Set (MCS) results for portfolio models

This table reports the number of times each model is included in the set of superior models (SSM) at the 5% significance level, along with the average p-value for each model across portfolios. Higher p-values indicate a greater likelihood of inclusion in the SSM.

	GARCH	GJR	EGARCH	LSTM	Universal LSTM
NLL	1647	1449	1893	1748	**9151**
	(0.123)	(0.102)	(0.133)	(0.118)	**(0.807)**
QLoss 1%	1723	1521	1972	1821	**9328**
	(0.128)	(0.107)	(0.137)	(0.122)	**(0.819)**
QLoss 2.5%	1686	1492	1927	1782	**9411**
	(0.126)	(0.105)	(0.135)	(0.120)	**(0.832)**
JointLoss 1%	1746	1573	1996	1847	**9233**
	(0.130)	(0.112)	(0.139)	(0.125)	**(0.814)**
JointLoss 2.5%	1703	1534	1952	1802	**9383**
	(0.127)	(0.108)	(0.137)	(0.122)	**(0.825)**

Tables 2.8 and 2.9 report the average one-day-ahead portfolio risk metrics and MCS results, respectively. The findings from the individual stock analysis remain consistent in the portfolio setting. The universal LSTM demonstrates significant improvements in out-of-sample performance compared to the local baselines. Moreover, the global LSTM is almost always included in the MCS for all portfolios—exceeding its forecasting performance for stocks. This superior performance compared to stock volatility forecasts can be attributed to the characteristics of portfolios, which typically align more closely with broader market movements. By leveraging its exposure to a diverse set of stocks during training, the universal LSTM model has learned those broader market patterns extensively resulting in more accurate and robust portfolio volatility forecasts.

2.6 Conclusion

Using volatility prediction as an empirical proving ground, we conduct a comparative analysis of local and global training approaches for neural networks in financial forecasting. At the highest level, our findings demonstrate that neural networks represent a data centric paradigm, distinct from econometric models, and that a global training approach should be the standard for their application to financial tasks. Applying inconsistent training methods to econometric and neural network models can lead to misleading comparisons of their relative performance.

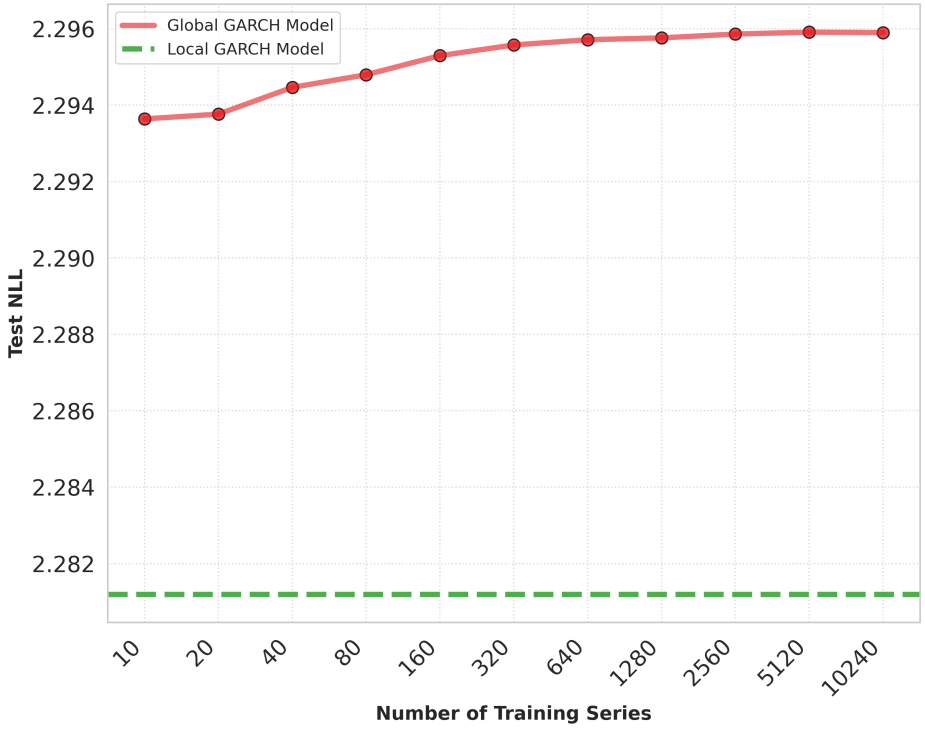
We also show that for financial data, the effect of data scaling far dominates that of model scaling or architectural choice, suggesting the fintech industry would benefit more from improving data volume and diversity than from engineering more complex models. Our interpretation reveals key properties of global neural networks, such as their ability to discern temporal importance, their robustness to outliers, and their adaptive response to rapidly changing market conditions.

Finally, through a universal volatility model, we demonstrate that the global approach enhances not only the prediction accuracy but also the practicality of neural networks in applied settings. It drastically reduces the computational and management overhead of

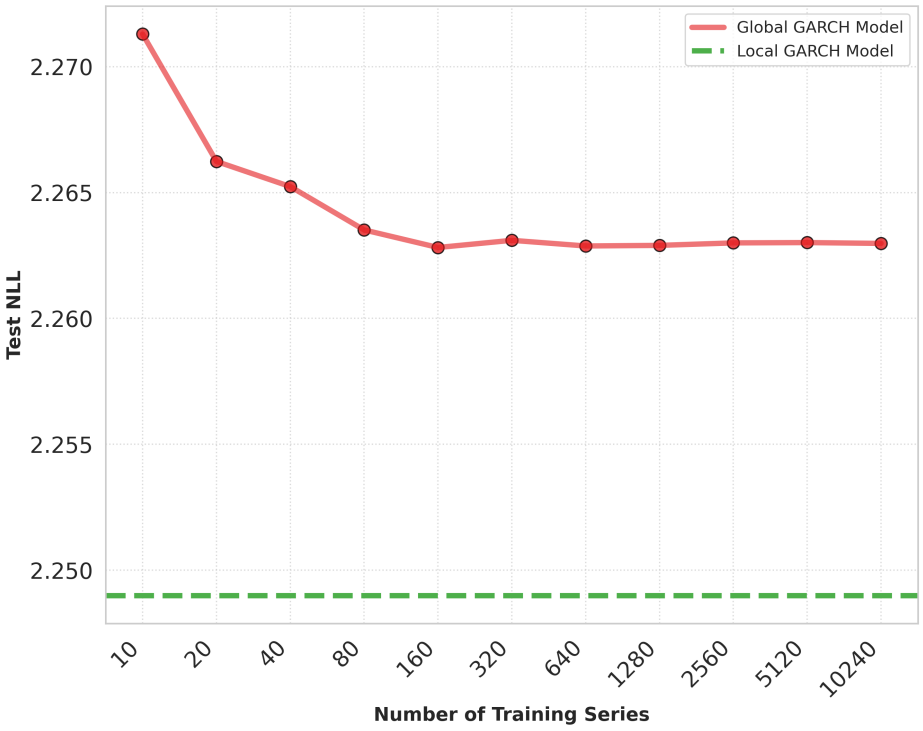
maintaining thousands of individual models and provides effective zero-shot forecasting for stocks with limited history.

Figure 2.2: GARCH data scaling effect

This figure plots the impact of data scaling on GARCH model performance. For supervised forecasts, the models are evaluated on 10 training stocks. For zero-shot forecasts, the models are evaluated on 1,531 previously unseen stocks.



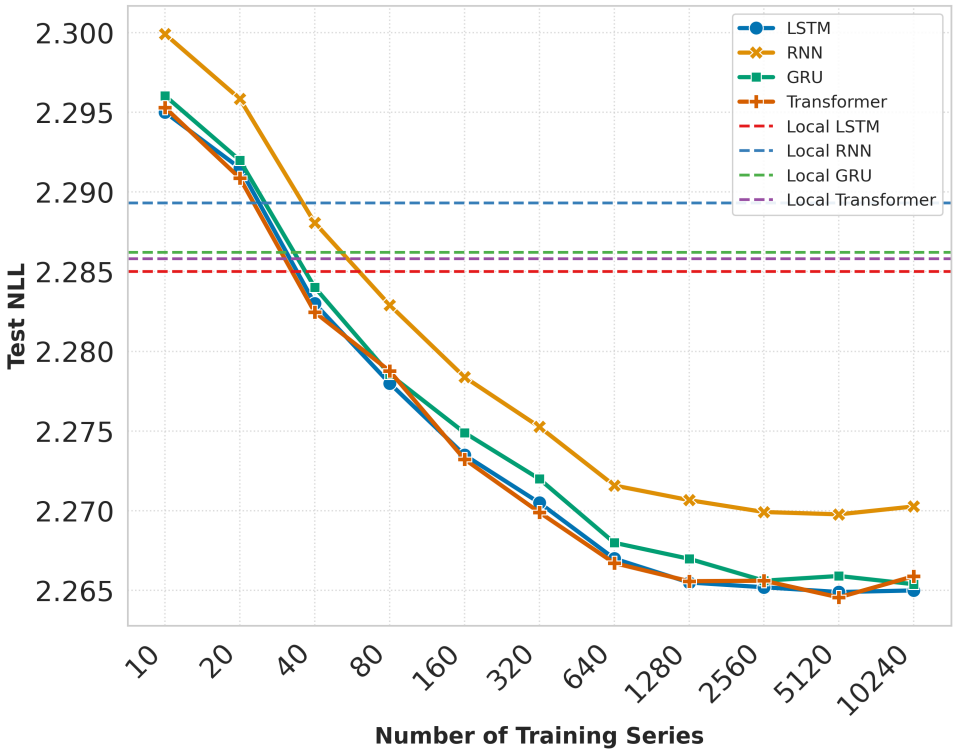
(a) Supervised



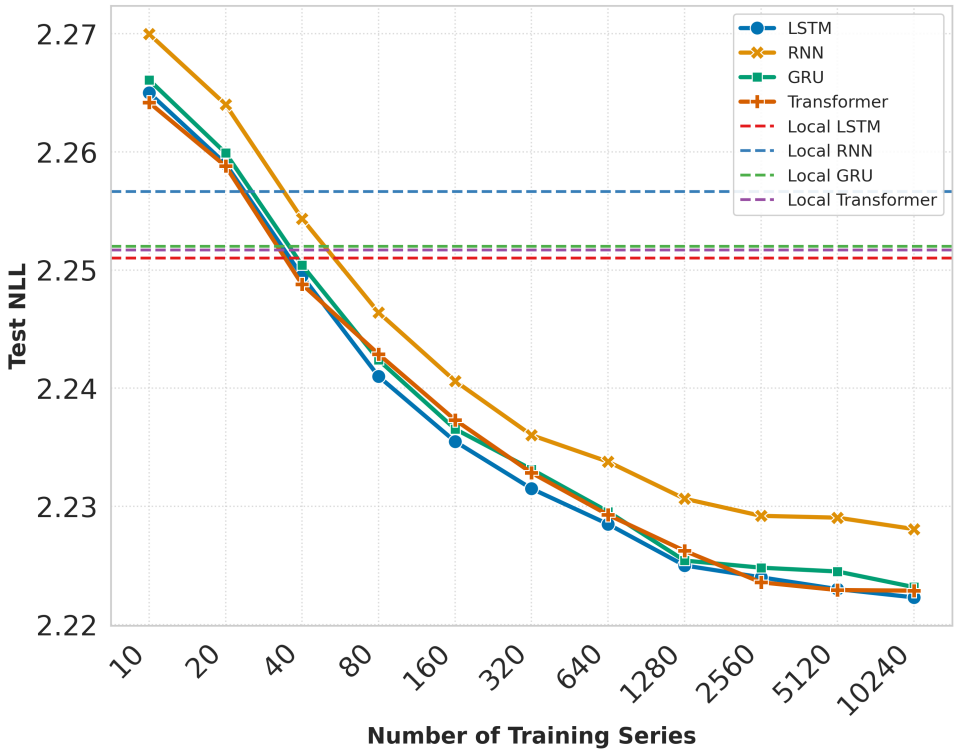
(b) Zero-shot

Figure 2.3: Neural network data scaling effect

This figure plots the impact of data scaling on neural network (NN) model performance. For supervised forecasts, the models are evaluated on 10 training stocks. For zero-shot forecasts, the models are evaluated on 1,531 previously unseen stocks.



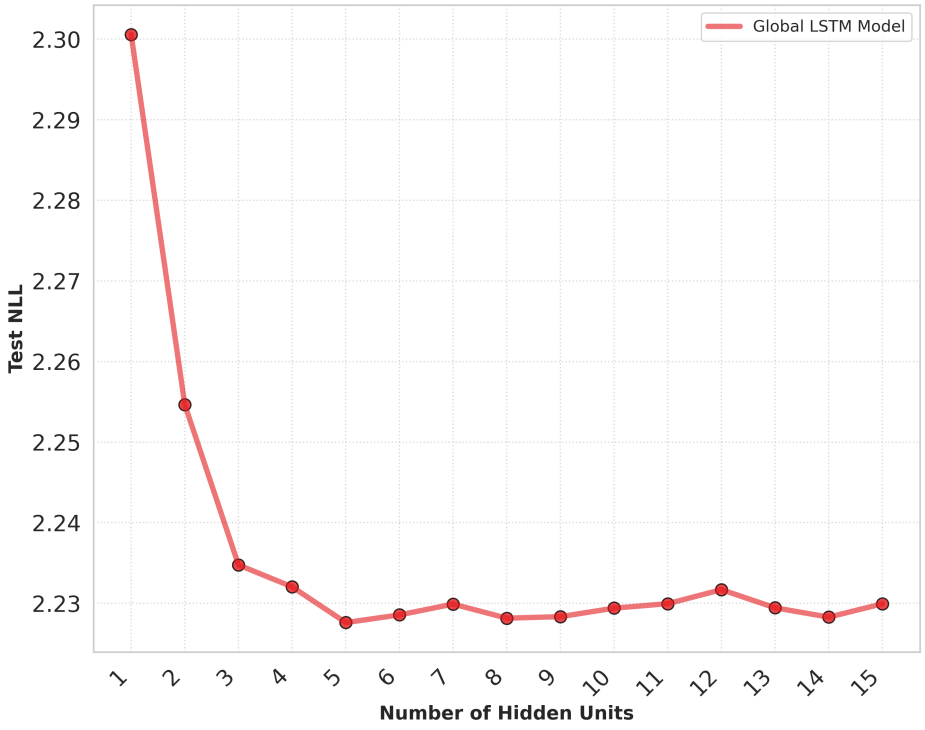
(a) Supervised



(b) Zero-shot

Figure 2.4: Model size versus performance for global LSTM

This figure plots the relationship between model size and performance for the global LSTM. The models are trained on 10,240 stock series and evaluated on all 11,771 stocks, covering both supervised and zero-shot forecasting scenarios.



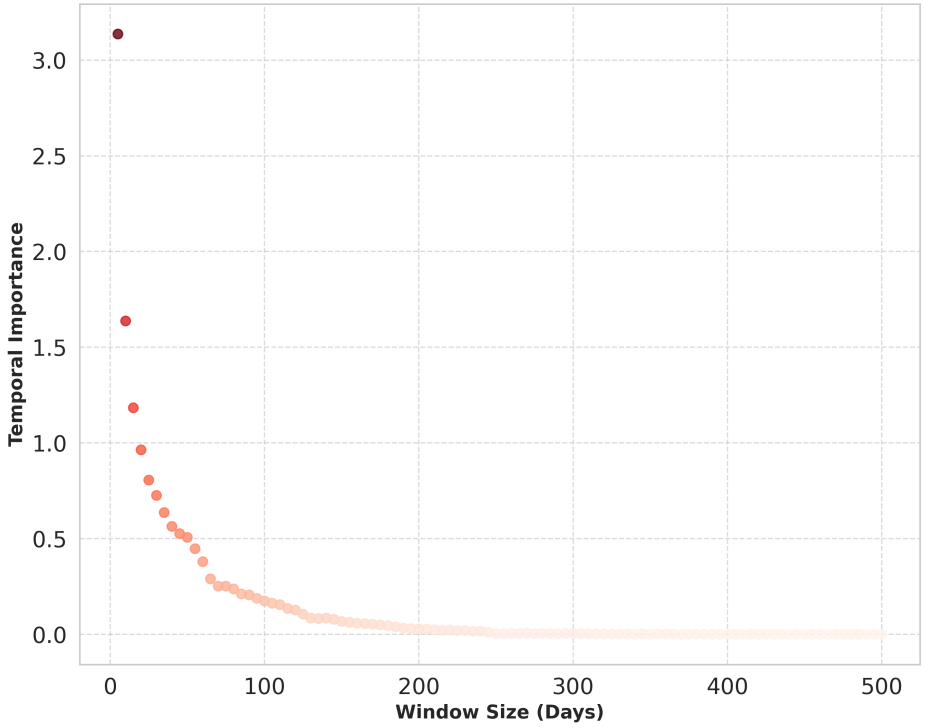


Figure 2.5: Temporal importance for the universal LSTM model.

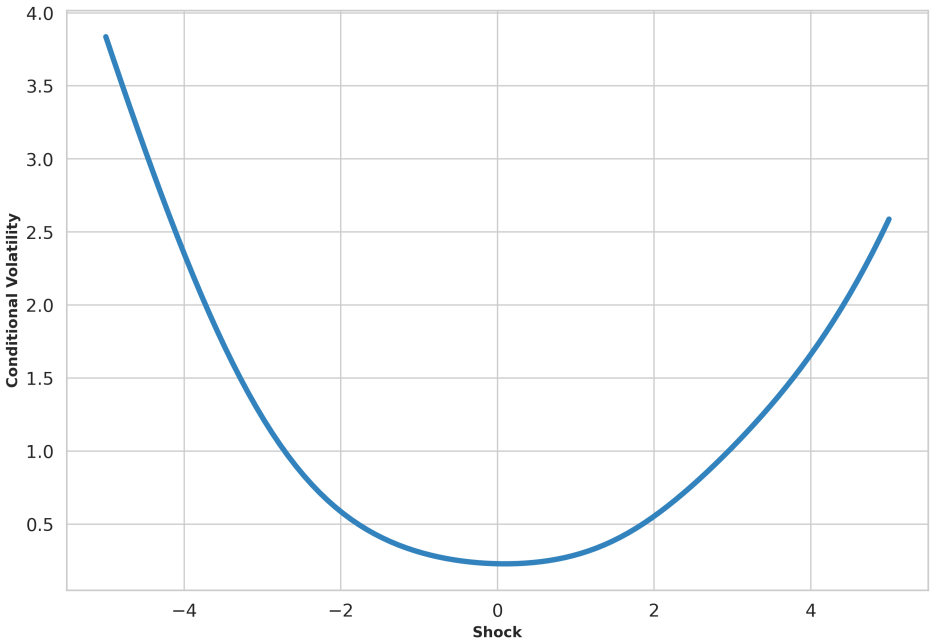
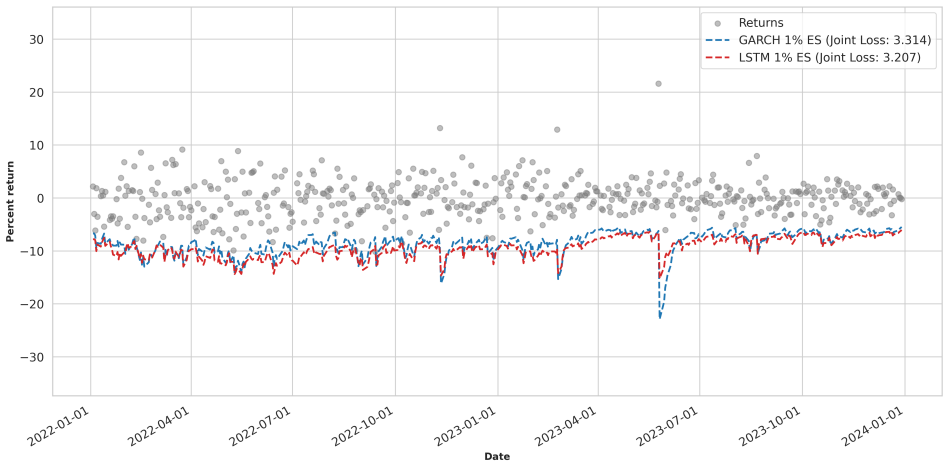
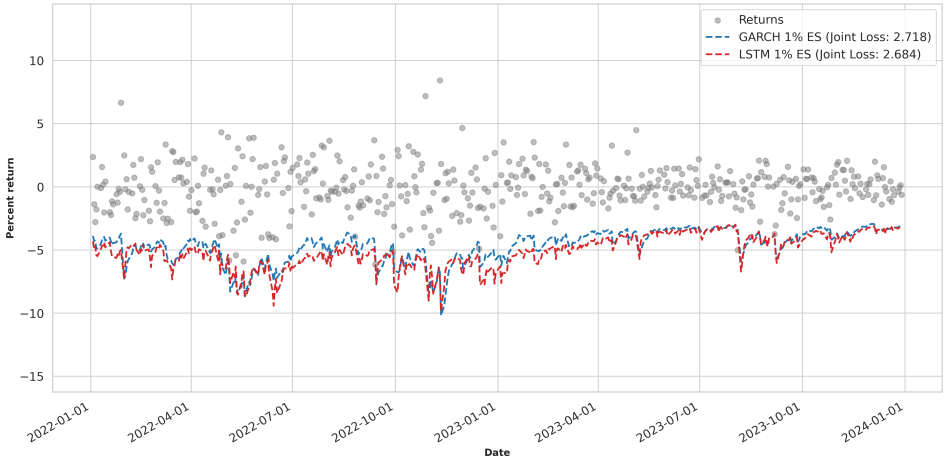


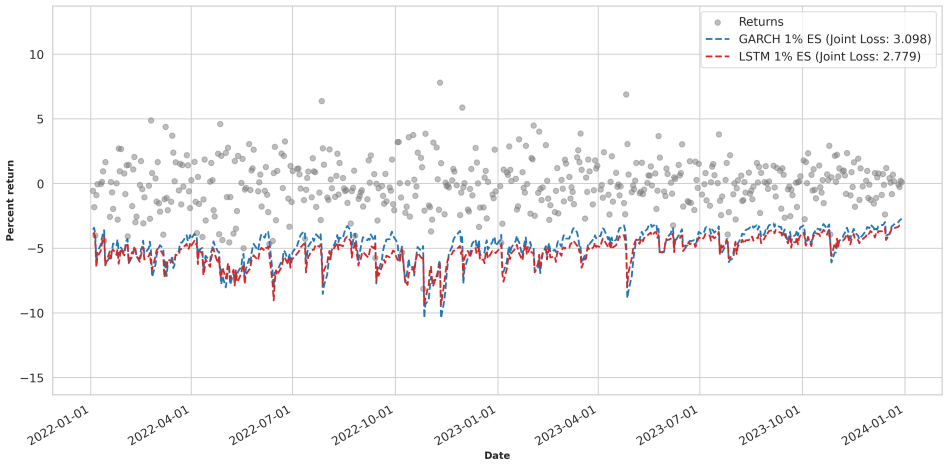
Figure 2.6: News impact curve of the universal LSTM model.



(a) Nvidia (NVDA)

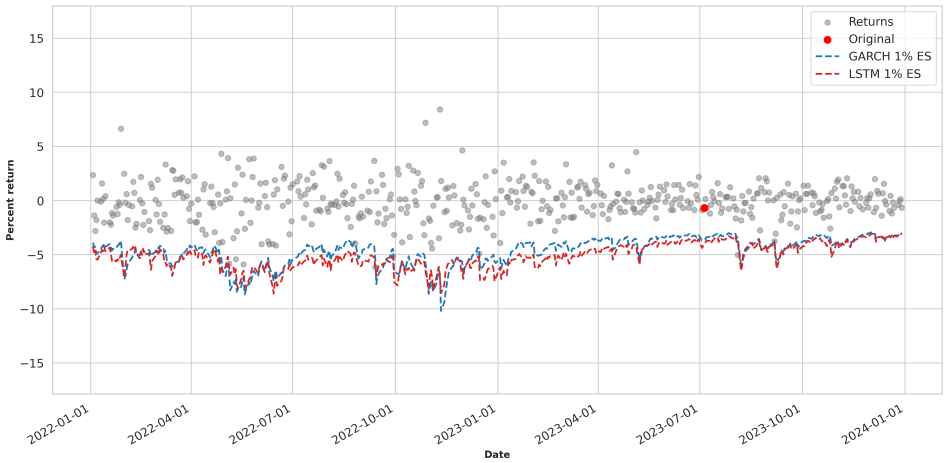


(b) Apple (AAPL)

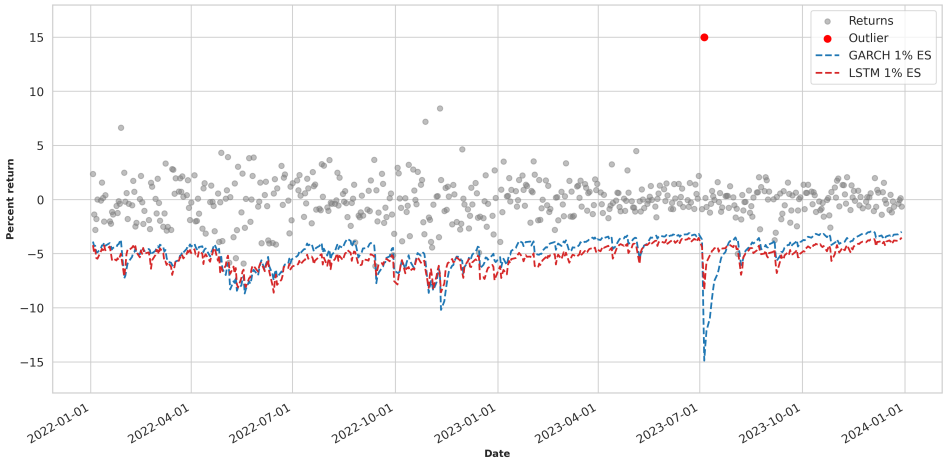


(c) Microsoft (MSFT)

Figure 2.7: 1% ES forecasts for top 3 market-cap companies.



(a) Without outliers



(b) With outliers

Figure 2.8: Impact of the outlier on the ES forecasts for Apple stock.

Chapter 3

Price Discovery in the Data-Centric Era

3.1 Introduction

In recent years, the empirical microstructure toolkit has expanded dramatically with the rise of machine learning (ML) and deep learning (DL) techniques. However, this rapid growth raises several important yet unanswered questions. First, can ML methods (e.g., random forests, gradient boosting trees) and DL methods (e.g., LSTMs, Transformers) uncover insights that traditional theoretical frameworks overlook, thereby informing and guiding the development of new financial theories? Second, to what extent do these techniques capture nonlinearities and interactions that traditional linear models fail to detect? Third, given the wide array of available ML and DL approaches, how should researchers choose among them? Addressing these questions is essential for understanding the value and potential of ML and DL in advancing microstructure research.

In this chapter, we address these questions by benchmarking traditional linear models with various advanced ML and DL techniques when modeling the price discovery process. Utilizing a highly granular dataset that records every order submission, cancellation, and amendment in a largely unfragmented market, we aim to assess how effectively established microstructure variables capture the dynamics of price discovery. While the-

oretical research has identified a suite of microstructure variables that proxy for order flow and market conditions, we have little understanding as to how effectively these proxy variables capture the price discovery process.

To bridge this gap, we first compare linear models based on established variables with ML models that use the same variables to quantify the role of nonlinearities and interaction effects within these variables. We then benchmark these linear and ML models against state-of-the-art DL models that utilize contemporaneous raw order book data (referred to as Level 3, or L3), which preserves market information in full detail. Through this benchmarking, we assess the extent to which L3 data contains information relevant to price discovery that is not captured by established variables. We then examine the common assumption that price formation is Markovian. Recent research challenges this view; for example, Riccò et al., 2023 proposes a non-Markovian model of a limit order market that incorporates path dependence in the price impact of orders. To engage with this debate, we extend a DL model to incorporate not only contemporaneous L3 data but also historical L3 data, which allows us to empirically assess the importance of temporal dependencies and the evolution of the order book in shaping the current market state.

We establish several main results. First, we examine the role of nonlinearities in established variables. By comparing univariate *linear* models with univariate *nonlinear* models, we find that, on average, the out-of-sample R^2 values are approximately 40% higher for the *nonlinear* models. This suggests that these variables exhibit nonlinear relationships with price discovery, indicating that nonlinearities should be taken into account. However, the benefits of incorporating nonlinearities are not uniform across all variables. For example, trade-related variables, such as directional trade size, display robust nonlinear effects that nearly double the R^2, whereas book-related variables, such as depth imbalance, show minimal improvement, suggesting a predominantly linear relationship. Furthermore, as expected, variables that proxy for market conditions—such as volatility or liquidity—and do not directly indicate buying or selling flow exhibit little explanatory power in a univariate setting. Interestingly, some directional variables, like VPIN and trade imbalance, which are anticipated to predict the direction of future price

movements, also demonstrate limited explanatory power when considered individually.

We then assess the role of variable interactions in multivariate models and find that, for linear specifications, incorporating interactions yields minimal incremental explanatory power. In contrast, nonlinear models benefit substantially from the inclusion of interaction terms. Notably, interactions between directional variables (eg. directional trade size) and non-directional variables (eg. liquidity) significantly enhance model performance. This finding aligns with Kwan et al., 2024, who demonstrate that certain market conditions can amplify or attenuate the price discovery process. Accordingly, variables that may exhibit limited predictive power in isolation should not be disregarded, as their interactions with other variables can be informative. Further analysis reveals that short-term price dynamics are predominantly driven by two key variables: (1) directional trade size and (2) the imbalance between the volume available at the best bid and ask. Together, these variables account for approximately 90% of the model's predictive power, underscoring their central role in the price discovery process.

Having presented the key properties of established variables, we assess their effectiveness in capturing market information by comparing DL models trained on contemporaneous L3 order book data with those trained on established variables. We find that the out-of-sample R^2 values are statistically indistinguishable between the two approaches, with the DL model using L3 data achieving an average R^2 of 2.29%, compared to 2.23% for the model using established variables. This finding suggests that the existing suite of microstructure variables effectively captures the information contained in contemporaneous L3 data. However, incorporating historical L3 data into the DL model leads to a substantial improvement in predictive performance. Specifically, including past order states increases the R^2 by more than 60%, a result that is robust across all stocks in our sample and statistically significant at the 0.1% level. This enhancement underscores the importance of temporal dependencies and the evolution of the order book in the price discovery process that is not adequately captured by existing theory driven variables.

Finally, we demonstrate that model selection in finance research needs to be carefully guided by two factors: the input format of the data and the sample size. DL methods

require substantially more training data than linear or ML models and typically exhibit comparative advantages when applied to high-dimensional raw inputs. By training DL models on different data sizes, we document a pronounced data scaling effect. Specifically, the predictive performance of DL models improves markedly as the training data increases. Consequently, studies using smaller datasets may find that a DL model using raw data underperform a ML model using established proxy variables. This observation aligns with findings from several ML studies in finance, which report that simpler ML models outperform DL architectures(Bali et al., 2023; Gu et al., 2020), a result that contrasts with typical conclusions in fields such as computer vision or bioinformatics. Our results suggest that this counterintuitive conclusion may stem from a systematic oversight of the data-intensive nature of DL methods in financial research.

Our findings have several important implications. First, our empirical results provide guidance for future theory. Notably, Riccò et al., 2023 has recently challenged the common assumption that the conditional distribution of future prices depends solely on the current market state. In particular, they propose that price formation exhibits path dependence and that order flow possesses persistent memory. Our results support their proposition and suggest that future theory should prioritize modeling the dynamics of order flow and order book evolution, which appear to contain economically and statistically significant predictive structure that is currently underexplored. Moreover, our results highlight that a majority of price discovery is driven by either directional trade size or the imbalance between the volume, or depth, available at the best bid and ask. Accordingly, these key features should be be a key component in future theory work. Further, our results demonstrate the importance that the interaction of market conditions on these two key variables can play. Second, we provide an overview of ML and FL techniques applicable to financial research. While recent studies have adopted various ML methods in market microstructure analysis, model selection often appears ad hoc. Our study aims to offer researchers a clearer understanding of the available methodologies and their respective strengths and weaknesses. Third, this chapter provides practical guidance when selecting among different ML and DL techniques. We demonstrate that model selection should be

guided by the data format and the available sample size. Specifically, DL models provide are most advntagesous when large raw data sets are avaialble, with no economic intuition on what variables or proxies should be used. In constrast, ML models are better suited to smaller data sets that use variables (or features enginneered from rat data) derived by economic intuition. Our study offers practical guidance for selecting among various ML and DL techniques in financial research. We demonstrate that model selection should be informed by two critical factors: the format of the input data and the available sample size. Specifically, DL models tend to be most advantageous when applied to large, high-dimensional raw datasets, particularly in scenarios lacking clear economic intuition for variable selection. In contrast, ML models are better suited to smaller datasets that utilize variables, or engineered features, that are derived from economic theory. By systematically benchmarking these models across varying data formats and sample sizes, we provide practical guidance on aligning model complexity with data characteristics, thereby facilitating more informed methodological choices in future financial research.

This chapter has several contributions to the existing literature. First, we provide empirical support for Riccò et al., 2023 who suggest that temporal dependencies among order flow is important. Similarly, our results confirm the conclusions of Kwan et al., 2024 who demonstrate that different market conditions can either amplify or attenuate the price discovery process of certain trades. Not only does this chapter provide empirical support for current concerns within the literature, it also contributes to the expanding body of literature that employ ML to microstructure problems. While Easley et al. (2021) and its extension by Karpman et al. (2023) utilize random forest to assess the predictive power of microstructure variables for short-term market measures. Further contributions include Kaniel et al. (2023), who leverage a unique dataset and MLP techniques to forecast mutual fund performance based on fund characteristics, and F. Jiang et al. (2024), who examine the efficacy of ML algorithms in forecasting stock price crash risks by exploiting firm-specific characteristics from the Chinese stock market. Prior studies have predominantly relied on human crafted variables, which represent only a partial extraction of the comprehensive market data and may discard subtle information that

closed-form constructs cannot capture. In contrast, this chapter is the first to explicitly compare the variable engineering approach with a data driven approach in which DL models extract latent patterns directly from raw market data.

3.2 Data

Our study focuses on the 20 largest stocks by market capitalization over a one-year period from January 3, 2023, to December 29, 2023. These stocks represent some of the most actively traded equities on the Australian Securities Exchange (ASX). We use a comprehensive dataset provided by Securities Industry Research Centre of Asia Pacific (SIRCA). The data contains millisecond-resolution records of all order book events, including stock symbols, order quantities and prices, unique order identifiers, broker codes, and event types (submit, trade, amend, or cancel). By tracing individual order identifiers through their life cycle of amendments, executions, and cancellations, we reconstruct the full limit order book with high precision. Given the ability to replay the complete order flow, our approach eliminates the need for trade classification algorithms to distinguish between buyer-initiated and seller-initiated trades, such as the Lee and Ready (1991) method. This direct methodology improves inference accuracy, aligning with prior findings (e.g., Ellis et al. (2000)) that the Lee and Ready rule can misclassify up to 20% of trades. Following Upson et al. (2021), we consolidate all trades executed at the same price and direction within a single millisecond into a unified marketable order. The ASX operates as a continuous price-time priority limit order market from 10:00 a.m. to 4:00 p.m., supplemented by randomized opening and closing auctions. To avoid distortions arising from auction intervals, following Kwan et al. (2024) we confine our analysis to trades and orders submitted between 10:10:00 and 16:00:00, and treat any orders still unexecuted at market close as canceled. Unlike the fragmented markets typical of the United States and Europe, the ASX maintained a dominant position, accounting for over 90% of daily equity turnover throughout our sample period. This concentrated market structure enables a more comprehensive analysis compared to fragmented systems, where

data aggregation can introduce biases. For instance, van Kervel (2015) demonstrate that consolidating quotes across fragmented venues can lead to liquidity overestimation, particularly when high-frequency trading (HFT) participants retract orders in response to trades on other platforms. Our study thus provides unique insights into HFT dynamics within a predominantly centralized market framework. Table 3.1 presents the descriptive statistics for the full set of 20 stocks. Each variable is computed on a daily basis and then averaged over the entire sample period.

Table 3.1: Summary statistics of trade data

This table presents summary statistics for the period under analysis. For each stock ticker symbol listed in the first column (and for the aggregate of all stocks, denoted as "ALL"), the subsequent columns report: `NTrades`, the total number of trades observed; `Price`, the average trade price in the relevant currency; `Spread`, the average quoted or effective bid–ask spread in currency units; `Size(Mean)`, the mean number of shares per trade; and `Size(Median)`, the median number of shares per trade.

	NTrades	Price	Spread	Size(Mean)	Size(Median)
ALL	1,135,342	38.44	0.015	104	33
ANZ	1,233,910	24.41	0.012	729	163
BHP	2,353,521	45.75	0.011	376	94
CBA	1,912,983	101.19	0.017	117	35
CSL	2,342,113	276.79	0.044	31	12
FMG	1,271,962	22.01	0.011	618	132
GMG	853,670	20.97	0.011	367	95
MQG	2,000,147	175.47	0.031	41	15
NAB	1,031,299	28.42	0.011	539	107
QBE	686,035	14.90	0.011	511	114
REA	1,085,889	144.77	0.058	17	8
RIO	2,471,242	117.84	0.021	61	21
RMD	1,270,232	27.58	0.013	241	84
STO	524,135	7.38	0.010	1971	313
TCL	489,519	13.71	0.010	819	176
WBC	856,756	21.78	0.011	931	170
WDS	1,625,800	34.88	0.012	318	73
WES	1,279,700	50.83	0.014	132	36
WOW	961,137	37.23	0.013	198	47
WTC	1,395,386	68.98	0.025	48	18

3.3 Methodology

In this section, we introduce the forecasting framework and the various models employed to predict short-term price movements. In its most general form, we describe the short-term price move for a stock as a prediction error model:

$$\hat{r}_{t:t+20} = f(X_t) + \epsilon_t, \tag{3.1}$$

where observations are indexed in tick time ($t = 1, \ldots, T$), $\hat{r}_{t:t+20}$ denotes the mid-price return forecasts over the subsequent 20 ticks and X_t denotes the input variables.[a] The target variable is defined as:

$$r_{t:t+20} = \frac{\text{mid price}_{t+20+\delta}}{\text{mid price}_{t+\delta}},$$

δ represents an infinitesimally small increment of time, allowing us to observe the midpoint shift immediately *after* a trade has occurred, thereby excluding any immediate price impact caused by the trade itself. This approach enables us to isolate and analyze any additional information the trade may convey that is not yet reflected in its immediate price impact. Here, $f(\cdot)$ is the model that maps the predictor X_t to the forecasting target $\hat{r}_{t:t+20}$. Without consideration of regularization (i.e., the techniques used to avoid overfitting), all model estimation is achieved by minimizing the mean squared error (MSE):

$$\mathcal{L}(\theta) = \frac{1}{T_{train}} \sum_{t=1}^{T_{train}} \Big(r_{t:t+20} - f\big(X_t; \theta\big) \Big)^2, \tag{3.2}$$

where θ are the model parameters. In this study, we consider three classes of predictive models: linear models, machine learning (ML) models, and deep learning (DL) models. From each class, we select two representative models for evaluation. The linear models include ordinary least squares (OLS) and the least absolute shrinkage and selection operator (LASSO). The ML models comprise the random forest (RF) and gradient-boosted

[a] Given that short-term price changes are often minimal or zero, we adopt a forecasting horizon of 20 ticks which reduces the proportion of observations with no price movement to approximately 33% in our training data.

regression trees (GBRT). The DL models include a multi-layer perceptron (MLP) and a Transformer. In the following sections, we provide a summary of each model and discuss the respective strengths and weaknesses of each approach.

3.3.1 Linear Models and Regularization

Linear models serve as a valuable baseline in our analysis for benchmarking more advanced machine learning (ML) and deep learning (DL) methods. Despite their restrictive linearity assumptions, these models offer a simple and transparent structure that helps mitigate overfitting and allows for intuitive interpretation of the relationship between explanatory variables and the prediction target

Among linear techniques, the OLS estimator remains a cornerstone in empirical finance. By minimizing squared residuals, OLS yields coefficient estimates that directly associate microstructure variables with observed price movements. However, a well-documented limitation of OLS is its tendency to include all available predictors regardless of their informativeness, potentially leading to overfitting when dealing with high-dimensional, unaggregated inputs such as market history data. To address this issue, we employ the LASSO estimator (Tibshirani, 1996), which incorporates a penalty on the model's coefficients in the objective function:

$$\mathcal{L}(\theta) = \frac{1}{T_{\text{train}}} \sum_{t=1}^{T_{\text{train}}} (r_{t:t+20} - f(X_t; \theta))^2 + \lambda \|\theta\|. \tag{3.3}$$

Here, $|\theta|$ denotes the penalty term and λ is the regularization parameter that governs the degree of penalization. When $\lambda > 0$, the penalty effectively shrinks the coefficients of uninformative predictors towards zero, thereby performing implicit variable selection. In our application, LASSO provides a more robust linear benchmark for analyzing unaggregated market data.

3.3.2 Machine Learning Models

While linear models remain favored for their simplicity and interpretability, empirical evidence increasingly suggests that price discovery is governed by nonlinear interactions among microstructure variables (Easley et al., 2021; Karpman et al., 2023). In this setting, ML methods-specifically, RF and GBRT-provide an attractive alternative, capturing these complex relationships while maintaining a relatively simple structure and low computational cost. The random forest method (Breiman, 2001) constructs a large collection of simple but diverse regression trees, each of which independently forecasts the same target variable. To promote heterogeneity across trees, each tree is trained on a randomly drawn subset of the full dataset, a process known as bootstrap sampling. The final prediction is obtained by averaging the outputs of the individual trees, which reduces variance and results in forecasts that are typically more accurate and reliable than those produced by a single regression tree. In contrast, the GBRT method (Friedman, 2001) builds a series of simple regression trees sequentially. Each new tree is trained to correct the prediction errors (i.e., the residuals) of the ensemble constructed thus far—a process known as boosting. The final prediction is then obtained by combining the outputs of all trees, typically via a weighted sum, which systematically reduces prediction bias. This sequential strategy enables GBRT to focus iteratively on its previous errors, often resulting in superior performance relative to RF in capturing complex, nonlinear dynamics. Moreover, the bootstrap sampling techniques in RF can also be integrated into the GBRT framework through stochastic gradient boosting, where each tree is fit to a subsample of the training data drawn at random. This further enhance GBRT's robustness and reduce overfitting, solidifying its position as the most effective method within the ML toolkit.

3.3.3 Deep Learning Models

Although RF and GBRT often perform well on engineered variables, their tree-based structure restricts their ability to learn complex patterns directly from high-dimensional, raw market data. By construction, each tree split is along a single feature axis, so

higher-order interactions of features cannot be uncovered without careful feature engineering ahead or yielding excessively deep trees prone to overfitting. More critically , RF and GBRT lack any built-in mechanism for temporal dependence, thereby ignoring the sequence structure of market history data. DL models overcome this limitation. Specifically, DL offers unparalleled flexibility and the ability to learn complex nonlinear patterns directly from large datasets. Both theoretical and empirical evidence confirms that even relatively simple DL architectures possess universal approximation capabilities, enabling them to represent virtually any mapping function under suitable conditions (Hornik et al., 1989; Lu et al., 2017). In applied settings, DL models are typically categorized into three groups based on the structure of the input data. First, multi layer perceptrons (MLPs) are most effective when applied to static inputs. As such, they are well suited for data that do not exhibit temporal dependencies, including either non time series data or time series data that can be treated as memoryless. Second, convolutional neural networks (CNNs) are primarily used in computer vision tasks, where they operate on spatially structured data such as images. Third, models designed for sequential or temporal data include recurrent neural networks (RNNs) or Transformer architectures, which are well suited for handling time series inputs due to their ability to capture dependencies across time. To assess whether conventional microstructure variables fully capture the predictive information embedded in high frequency market data, we compare the performance of a machine learning model trained on microstructure features with that of a multi layer perceptron trained directly on raw limit order book observations. A detailed description of the raw data format is provided in Section 3.4.2. Similarly, to investigate whether raw historical order flow contains predictive information beyond what is available in contemporaneous market snapshots, we compare the performance of a multi layer perceptron trained on snapshot data with that of a Transformer model trained on both raw historical and contemporaneous snapshot data.

Multi-Layer Perceptron A multilayer perceptron (MLP) (Hornik et al., 1989) is a type of neural network used to find patterns in data. It consists of layers of interconnected units: an input layer that receives the data, one or more hidden layers that process

the data and an output layer that generates the final prediction. Each layer applies a simple mathematical operation followed by a nonlinear transformation (like sigmoid) to help the network learn complex patterns. As we increase the number of hidden layers (depth) or add more units to each layer (width), the MLP becomes capable of capturing more complex nonlinear relationships. In price discovery, MLPs have shown promise in identifying subtle patterns from large and complex data sources—such as detailed records of orders and trades (Z. Zhang et al., 2019, 2021). However, a limitation of MLPs is that they treat each input in isolation and do not account for the order in which events occur. Because of this, they may struggle to capture the time-dependent nature of financial data, such as how past prices influence future ones.

Transformer Transformers (Vaswani et al., 2017) are a powerful type of model developed for processing sequences of data, such as language or time-series information. Unlike traditional neural networks that treat each input point separately, Transformers include an attention mechanism that helps the model decide which moments in the past are most relevant for making predictions. Consider a series of snapshots of market data, each snapshot representing the state of the order book and trades at a specific point in time. The Transformer looks at all pairs of snapshots, compares them, and assigns a score to show how similar or connected they are. These scores are then used to create weights that tell the model how much attention to pay to each point in time. This process results in a reweighted version of the market's history, highlighting the most important moments. The updated data is then passed through layers similar to those in traditional neural networks (MLPs), allowing the model to learn complex relationships. What makes Transformers especially effective is their ability to capture both short-term details and long-term trends in data. In simple terms, a Transformer is like a smart version of a traditional neural network that automatically learns which past events matter most. If the model receives only a single moment as an input, rather than a sequence, then it behaves just like a standard MLP. However, when given a full history, its attention mechanism allows it to identify subtle patterns across time.

DL Interpretation via Saliency Map While DL models may enhance predictions, one limitation is their 'black box' nature that can mask variable significance. To overcome this limitation we use an interpretative tool known as a saliency map (Simonyan et al., 2014). Saliency maps use backward gradients to quantify the importance of each input variable and have been widely applied in computer vision tasks to identify the regions of an image that most strongly influence DL model's predictions. In the study, we use them to reveal which components of the market data drive the model's output. At a high level, the values produced by a saliency map can be interpreted analogously to coefficients in a linear model applied to normalized inputs, directly quantifying the influence of each predictor. Formally, let $\tilde{X}_t$ denote the normalized input. The saliency map S is defined as the gradient of the model's forecast $\hat{r}_{t:t+20}$ with respect to $\tilde{X}_t$:

$$S = \frac{\partial \hat{r}_{t:t+20}}{\partial \tilde{X}_t}. \tag{3.4}$$

To facilitate interpretation, we apply min–max normalization to S so that the resulting saliency scores range from 0 to 1:

$$S_{\text{norm}} = \frac{|S| - \min(|S|)}{\max(|S|) - \min(|S|)}, \tag{3.5}$$

where $|S| = \left|\frac{\partial \hat{r}_{t:t+20}}{\partial \tilde{X}_t}\right|$. The normalized saliency map S_{norm} provides a clear visualization of the relative importance of each predictor, thereby highlighting which aspects of the market data DL models focused on to make their predictions.

3.3.4 Evaluation Metrics

Following standard practice in financial ML literature (Bali et al., 2023; Feng et al., 2020; Gu et al., 2020), we assess model performance using the out-of-sample R^2. This metric quantifies the proportion of variance in out-of-sample observations that is explained by

the model. Formally, the out-of-sample R^2 is defined as

$$R^2 = 1 - \frac{\sum_{t=1}^{T_{\text{test}}} \left(r_{t:t+20} - \hat{r}_{t:t+20}\right)^2}{\sum_{t=1}^{T_{\text{test}}} \left(r_{t:t+20} - \overline{r}\right)^2}, \tag{3.6}$$

where $\overline{r}$ denotes the mean of the observed returns over the out-of-sample period and T_{test} is the number of out-of-sample observations. In line with standard practice, we partition our dataset into training, validation, and testing samples. The training sample is used to estimate the model, the validation sample to calibrate its hyperparameters, and the testing sample to assess out-of-sample performance. Specifically, for the one-year period under study, we allocate seven months (January–July) for training, one month (Aug) for validation, and three months (September–December) for rigorous out-of-sample evaluation. Unless otherwise noted, all R^2 statistics reported in subsequent sections refer to out-of-sample evaluation.

3.4 Variables

This section details the predictor variables, X_t in Equation (3.1), employed in our analysis. A central objective of this study is to evaluate the information content of established, theory-motivated microstructure variables relative to the granular, unaggregated data streams increasingly available. Therefore, we structure our analysis around two distinct input representations: (i) a set of established variables derived from prior theoretical and empirical work, representing a traditional feature-engineering approach, and (ii) unaggregated market data, representing a data-driven approach that allows models to potentially capture nuances missed by pre-defined constructs.

3.4.1 Established Variables

We select fourteen variables grounded in market microstructure theory, categorized into trade-related, book-related, and order-history-related groups. Each group of variables aim to capture distinct facets of short-horizon market behavior—from instantaneous trading impulses to the dynamic evolution of liquidity over time.

Trade Variables

Trade variables capture the most direct mechanism of price discovery, namely, executed transactions through which private information flows into prices (Glosten & Milgrom, 1985; Kyle, 1985).

Trade Price Delta (TPD). Although the *absolute* trade price offers limited new information (Hasbrouck, 1991), recent trade-to-trade price changes can reveal evolving supply-demand imbalances. We thus define

$$\mathrm{TPD}_t \;=\; 100 \times \big(P_{\mathrm{trade},t} \;-\; P_{\mathrm{trade},t-1}\big), \tag{3.7}$$

where $P_{\mathrm{trade},t-1}$ is the most recent trade price before the current transaction. By scaling the difference by 100, we convert small high-frequency price variations into interpretable units (e.g., tick changes). Positive (negative) TPD values imply short-run upward (downward) price adjustments possibly driven by new information or transient liquidity shocks.

Signed Trade Size. Signed trade size assigns a positive or negative value to executed volumes according to the trade-initiating side (buyer- or seller-initiated). Larger trades often signal institutional participation and potentially elevated information content. Empirical evidence further suggests that large, aggressively executed trades can significantly impact short-term price dynamics (Hasbrouck, 1991), especially if market depth is limited or if traders perceive these orders to be informed. Formally, we compute

$$\text{Signede Size}_t \;=\; \begin{cases} +\,\text{Volume}_t, & \text{if buy-initiated,} \\ -\,\text{Volume}_t, & \text{if sell-initiated.} \end{cases}$$

By explicitly incorporating direction, we capture net buying or selling pressure at the transaction level.

Trade Time Delta. The spacing between successive trades can be highly informative in high-frequency markets (R. F. Engle & Russell, 1998). We measure the *Trade Time*

Delta (in milliseconds) as the gap between consecutive trades:

$$\text{TimeDelta}_t \;=\; (\tau_t - \tau_{t-1}),$$

where τ_t denotes the timestamp of the t th trade. Clusters of small intervals often coincide with bursts of trading intensity, potentially reflecting rapid news dissemination or algorithmic strategies. By contrast, large gaps can signal temporary market standstills, heightened uncertainty, or a lack of liquidity-taking interest.

Book Variables

Book variables represent the state of the limit order book at or immediately prior to a trade, thereby capturing liquidity conditions, prevailing market sentiment, and potential future price pressure (O'Hara, 1998).

Relative Spread (RS). We compute the inside (best) spread normalized by the midquote:

$$\text{RS}_t = \frac{P^{(1)}_{\text{ask},t} - P^{(1)}_{\text{bid},t}}{\frac{1}{2}\left(P^{(1)}_{\text{ask},t} + P^{(1)}_{\text{bid},t}\right)}, \tag{3.8}$$

where $P^{(1)}_{\text{ask},t}$ and $P^{(1)}_{\text{bid},t}$ are the best ask and bid prices, respectively. Narrow spreads suggest heightened competition among liquidity suppliers and lower transaction costs, which typically implies more efficient price discovery (Bessembinder & Venkataraman, 2010). Wider spreads, conversely, may indicate uncertainty, asymmetric information, or inventory concerns.

Depth Imbalance (DI). Depth imbalance quantifies the relative weighting of buy and sell volume at the top levels of the order book, often reflecting latent demand or supply. At a chosen depth L, we define

$$\text{DI}_L \;=\; \frac{\sum_{i=1}^{L} \text{Size}_{\text{Bid},i} \;-\; \sum_{i=1}^{L} \text{Size}_{\text{Ask},i}}{\sum_{i=1}^{L} \text{Size}_{\text{Bid},i} \;+\; \sum_{i=1}^{L} \text{Size}_{\text{Ask},i}}, \tag{3.9}$$

in line with Cont et al. (2014). Intuitively, a large positive (negative) DI_L suggests impending upward (downward) price movement, particularly if one side of the book is disproportionately thick relative to the other.

Order Imbalance (OI). Whereas DI_L concentrates on volumes, an *order imbalance* (OI) approach tracks the count of bid vs. ask orders:

$$\mathrm{OI}_L = \frac{\sum_{i=1}^{L} N_{\mathrm{Bid},i} - \sum_{i=1}^{L} N_{\mathrm{Ask},i}}{\sum_{i=1}^{L} N_{\mathrm{Bid},i} + \sum_{i=1}^{L} N_{\mathrm{Ask},i}}, \tag{3.10}$$

with $N_{\mathrm{Bid},i}$ and $N_{\mathrm{Ask},i}$ denoting the respective order counts at the ith price level. Empirical studies link sizable order-count imbalances to imminent order history pressure and short-term price drift (Chordia et al., 2002).

Order History Variables

Although instantaneous book measures are invaluable, capturing *temporal* evolution is equally critical. Recent theoretical and empirical findings underscore that dynamic order history interactions can be as informative as static snapshots (Easley et al., 2012; O'Hara, 1998).

VPIN. Volume-Synchronized Probability of Informed Trading (VPIN) aggregates buy- and sell-initiated volumes over short windows, thereby estimating adverse selection risk (Easley et al., 2012). In a rolling fashion, we define

$$\mathrm{VPIN} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{V_i^B - V_i^S}{V_i^B + V_i^S} \right|, \tag{3.11}$$

where V_i^B and V_i^S are total buy and sell volumes in the ith volume bucket, and n is the number of buckets in the lookback. High VPIN values often precede volatility spikes, reflecting intensified trading by (potentially) informed traders.

Amihud Illiquidity (Lambda). Amihud's lambda measures the average price impact per unit of volume, capturing temporary or structural liquidity deficits (Amihud, 2002).

Over a window W, we compute

$$\lambda = \frac{1}{W} \sum_{t=1}^{W} \frac{|r_t|}{\text{Volume}_t}, \tag{3.12}$$

where r_t is the return over a short interval. Larger values indicate that even modest trades can move prices substantially, potentially signaling fragility or heightened risk aversion among liquidity providers.

Realized Volatility (RV). Realized volatility measures the cumulative squared price changes across high-frequency intervals and is a standard metric of intraday risk (Andersen et al., 2001). For a short time partition $\{1, \ldots, W\}$ within the trading day, we define

$$\text{RV} = \sqrt{\sum_{i=1}^{W} r_i^2}, \tag{3.13}$$

where r_i are intraperiod returns. Markets with elevated RV often experience faster or more volatile price formation, underscoring the market's sensitivity to new information.

Trade Imbalance (TI). To capture trends in directional trading over a rolling lookback W, we define

$$\text{TI}_t = \frac{1}{W} \sum_{\tau=t-W}^{t-1} \text{sign}\big(\text{Trade}_\tau\big), \tag{3.14}$$

where sign$(\cdot)$ reflects whether each trade is buyer- or seller-initiated. This measure highlights persistent directional biases—often indicative of institutional meta-orders or sustained market sentiment shifts (Hendershott et al., 2011).

3.4.2 Market Information in Full Detail

Although the established variables described above provide theoretically grounded proxies for market conditions, they capture only a fraction of the information embedded in market data. By design, theory-driven variables may inadvertently discard subtle signals and complex nonlinear dynamics that defy closed-form representation. In particular, order histories may exhibit intricate temporal dependencies and cross-level interactions that are

not readily distilled into conventional microstructure measures. This raises the question of whether these established variables fully exploit the available information in market data. To address this issue, we introduce two alternative input formats that preserve the complete information content of the data at different levels.

Market Snapshot. A market snapshot is represented as a single vector $\mathbf{x} \in \mathbb{R}^d$ that encapsulates the state of the market at a given instant. In our implementation, we set $d = 63$, and the vector is constructed as follows:

$$\mathbf{x}_t = \Big[\underbrace{T_{\text{price}},\, T_{\text{size}},\, T_{\text{time}}}_{\text{Trade attributes}},\, \underbrace{B_1,\, B_2,\, \ldots,\, B_{10}}_{\text{Book attributes}}\Big], \tag{3.15}$$

where each $B_i \in \mathbb{R}^6$ corresponds to six book attributes—specifically, bid and ask prices, volumes, and order counts (i.e., the number of orders present in the book)—at the i^{th} depth level. The trade time T_{time} is normalized relative to the trading period as

$$T_{\text{time}} = \frac{t - t_{open}}{t_{close} - t_{open}}, \tag{3.16}$$

with t denoting the trade time converted to nanoseconds, t_{open} representing the market opening time (10:10 a.m.), and t_{close} the market closing time (4:00 p.m.). This snapshot encapsulates the raw information required to compute traditional static microstructure measures while preserving nuanced details that may reveal additional non-linear patterns.

Market History. In contrast to a single snapshot, market history captures the evolution of market conditions over time. It is represented as a matrix $\mathbf{X} \in \mathbb{R}^{d \times W}$, where $d = 63$ is the dimension of each snapshot and W is the lookback window size. Formally, we construct

$$\mathbf{X}_t = \big[\mathbf{x}_t,\, \mathbf{x}_{t-1},\, \ldots,\, \mathbf{x}_{t-W+1}\big], \tag{3.17}$$

where each $\mathbf{x}_i$ is defined as above. This sequential representation preserves the temporal evolution of quotes and trades, enabling models, particularly those designed for time-series analysis, to capture intertemporal dependencies and complex liquidity shifts that

aggregated variables may overlook.

3.5 Empirical Results

In this section, we assess the efficacy of different modeling techniques and data representation methods to model the price discovery process. First, we compare the predictive performance of linear and non-linear methods. Next, we investigate the importance of interactions between variables. Finally, we investigate how much additional information is contained in raw market data, or L3 data, compared to established microstructure variables found in the literature.

3.5.1 Nonlinear Dynamics in Price Discovery

We begin by examining whether nonlinear modeling techniques can enhance the explanatory power of established microstructure variables and identify the sources of any improvement. Accordingly, we estimate both the linear models and the ML models using the microstructure variables in isolation and all combined together. Table 3.2 reports the averaged out-of-sample R^2 for models estimated via both the linear (OLS and LASSO) and nonlinear (RF and GBRT) frameworks. Panel A reports the out-of-sample R^2 for microstructure variables that proxy for directional buying or selling pressure. Accordingly, these variables could provide some indication on future price movements as their information gets compounded into prices. In contrast, Panel B reports R^2's for non-directional variables that should not make any predictions about future price movements. For example, volatility gives no indication about the direction of future price changes, only the speed or magnitude of the change. While these variables make no directional predictions, they may proxy for market conditions when the directional variables are more pronounced. In other words, even though the non-directional variables have negative R^2's, these variables may serve as useful interactions with the directional variables.

Nonlinearity in Individual Variables

Table 3.2 reports mixed results depending on the variable of interest. We observe a substantial improvement in predictive accuracy for trade-related variables when nonlinearities are accounted for. Specifically, the R^2 for the signed trade size nearly doubles from 0.749% (OLS) to 1.543% (GBRT), and the R^2 for the trade price delta increases from 0.163% (OLS) to 0.323% (GBRT). This finding aligns with prior research documenting nonlinear relationships between trade characteristics and permanent price impact. For instance, Hasbrouck (1991) identifies a concave relationship between trade size and permanent price impact, and Kwan et al. (2024) demonstrates that ignoring nonlinearities in trade size effects can lead to incorrect inferences about price discovery.

In contrast, variables derived directly from the limit order book, such as Depth Imbalance and Order Imbalance at both Level 1 and Level 5, exhibit more consistent predictive performance across linear and nonlinear frameworks. The R^2 values for these variables are similar in models. This suggests that the relationship between the immediate LOB state (as captured by standard imbalance measures) and subsequent price changes is predominantly linear or only weakly non-linear. This may occur because LOB imbalances reflect relatively direct supply and demand pressures near the best quotes, whose marginal impact on prices is less state-dependent than the impact of executed trades, which can carry more complex information signals.

We also examine established proxies for order history, whose effectiveness in high-frequency prediction is debated in the literature. Andersen and Bondarenko (2014), for example, critically evaluate the VPIN and find its ability to consistently predict market volatility is limited across different assets and market conditions. While Chordia et al. (2002) find predictive power in aggregate trade imbalances for lower-frequency returns, Cartea et al. (2015, Chapter 6) suggest this may not hold for individual stocks at high frequencies. Our findings strongly support these concerns in our high-frequency setting. VPIN, Amihud Lambda, and Trade Imbalance, used as standalone predictors, yield negligible or negative out-of-sample R^2 values across all models. This poor performance suggests that, at the intraday horizon studied, these measures are either noisy proxies for

the relevant information flow or their informational content is already captured by the contemporaneous LOB state and recent trade activity.

Table 3.2: Predictive Power of Individual Microstructure Variables

The table reports the average out-of-sample R^2 (in percent) for models predicting short-term price changes. Models are estimated using Ordinary Least Squares (OLS), LASSO, Random Forest (RF), and Gradient Boosted Regression Trees (GBRT). Each row corresponds to a model using only the specified variable as a predictor. The "All Features" row reports the R^2 when all 13 engineered microstructure variables are used jointly. Negative R^2 values indicate performance worse than predicting the sample mean. Results are averaged across all sample stocks and test periods.

Variable	OLS	LASSO	RF	GBRT
Panel A: Directional Variables				
Signed Trade Size	**0.749**	**0.739**	**1.529**	**1.543**
Depth Imbalance L1	0.676	0.665	0.684	0.676
Order Imbalance L1	0.648	0.635	0.651	0.645
Order Imbalance L5	0.355	0.352	0.349	0.349
Depth Imbalance L5	0.232	0.232	0.216	0.233
Trade Price Delta	**0.163**	**0.152**	**0.310**	**0.323**
VPIN	-0.024	-0.023	-0.036	-0.012
Trade Imbalance	-0.073	-0.064	-0.081	-0.027
Panel B: Non-Directional Variables				
Time Delta	-0.017	-0.017	-0.025	-0.017
Trade Frequency	-0.019	-0.018	-0.051	-0.017
Relative Spread	-0.021	-0.020	-0.687	-0.021
Amihud Lambda	-0.023	-0.021	-0.052	-0.018
Volatility	-0.035	-0.025	-0.172	-0.027
All Features	**1.534**	**1.556**	**2.003**	**2.277**

The Role of Variable Interactions

ML models like RF and GBRT excel at capturing non-linear interactions among predictors, whereas linear models typically require pre-defined interaction terms. We quantify the importance of such interactions by calculating the marginal gain in predictive power from combining all variables pairwise [b]. We define the marginal R^2 gain for a pair of

[b] Based on the 13 unique microstructure variables considered, a total of $13 \times (12)/2 = 78$ distinct pairwise interactions were evaluated.

features (F_1, F_2) as:

$$R^2_{\text{Marginal}} = R^2(F_1, F_2) - \max(0, R^2(F_1)) - \max(0, R^2(F_2)), \tag{3.18}$$

where $R^2(F_i)$ is the out-of-sample R^2 from a model using only feature F_i, and $R^2(F_1, F_2)$ is the R^2 from a model using both features. We truncate negative individual R^2 values to zero to isolate the contribution purely from the interaction. Table 3.3 summarizes the marginal R^2 gains for the top five pairwise interactions identified by GBRT, comparing these gains across four modeling frameworks. Notably, we observe that some features—specifically Relative Spread, Time Delta, Trade Imbalance which show little or even negative predictive power in isolation and in linear models contribute non-negligible improvements when combined non-linearly with strong directional predictor Trade Size and Trade Price Delta.

Consider the interaction between Signed Trade Size and Relative Spread. GBRT identifies a marginal gain of $R^2_{\text{Marginal}} = 0.176\%$, whereas linear models find no such effect. This suggests a non-linear state-dependent relationship: the price impact of signed trade volume is conditional on the prevailing bid-ask spread. This finding resonates with theories of adverse selection in market making (Glosten & Milgrom, 1985; Kyle, 1985). If wider spreads reflect higher perceived risk of trading against informed agents, and large trades are more likely to be informed (Easley & O'Hara, 1987), then the price impact of a large trade should be amplified when the spread is wide. Lin et al. (1995) provide empirical support for a link between the adverse selection component of the spread and trade size. Our results offer data-driven evidence for this theoretically motivated interaction, demonstrating a nonlinear dependency missed by standard linear specifications.

Similarly, GBRT finds positive interactions involving Time Delta (time between consecutive trades) with Trade Size Signed ($R^2_{\text{Marginal}} = 0.135\%$) and Trade Price Delta ($R^2_{\text{Marginal}} = 0.098\%$). This indicates predictive value in jointly considering trade timing and trade characteristics. This aligns with the intuition that the informational content or market impact of a trade may depend on the recent trading intensity. Models of trade duration, such as the Autoregressive Conditional Duration (ACD) framework developed by

R. F. Engle and Russell (1998), explicitly acknowledge that trading activity is clustered and time-varying. High-frequency bursts of trading (small Time Delta) might signal information events or liquidity shocks, potentially altering the price response function compared to periods of market quiescence. Furthermore, strategic trading models, such as that of Admati and Pfleiderer (1988), predict trade clustering by both informed and liquidity traders, implying complex dependencies between arrival times and information content. The GBRT model appears adept at discerning these subtle, state-dependent effects related to the market's trading rhythm.

While the individual R^2 contributions from these pairwise interactions may appear modest compared to the main effects of dominant variables like Trade Size Signed or Depth Imbalance, their collective presence is significant. They underscore the existence of economically plausible interdependencies among microstructure variables. These nuanced relationships, largely opaque to linear specifications, are identified and exploited by RF and GBRT, contributing significantly to their superior performance when utilizing the full feature set. This aligns with the overarching perspective articulated regarding the intricate, multi-faceted nature of price discovery, involving complex interplay between liquidity, order flow, and information. It further motivates the application of machine learning methods capable of handling such complexity, a direction increasingly explored in financial econometrics (see, e.g., Easley et al., 2021; Gu et al., 2020).

Table 3.3: Marginal Predictive Gains from Pairwise Interactions

The table reports the marginal out-of-sample R^2 (in percent) from combining two features, calculated using Equation (3.18). Each column represents a different predictive model (OLS, LASSO, RF, GBRT). The pairs shown are those yielding the largest marginal R^2 under the GBRT model. Individual feature R^2 values are reported in Table 3.2; the R^2 values for the second feature (F_2) in these specific pairs were negligible (≈ 0) when used individually.

Feature Pair		Marginal R^2 (%) by Model			
Feature A	Feature B	OLS	LASSO	RF	GBRT
Signed Trade Size	Relative Spread	-0.003	-0.002	0.050	0.176
Signed Trade Size	Time Delta	0.000	0.000	0.055	0.135
Trade Price Delta	Time Delta	0.001	0.001	0.082	0.098
Trade Price Delta	Relative Spread	-0.003	-0.002	0.072	0.080
Signed Trade Size	Trade Imbalance	-0.058	-0.049	0.031	0.058

Predictive Contribution of Feature Categories

As previously discussed, we group our engineered variables into three categories: Trade Variables (capturing characteristics of executed transactions), Book Variables (capturing the state of the LOB), and Order History Variables (proxies for information in order flows). To further understand the predictability of each group, we estimate five separate models using variables within each group separately, as well as combinations, across the four modeling frameworks. Table 3.4 reports the resulting out-of-sample R^2 values.

Consistent with our individual variable analysis, both Trade Variables and Book Variables demonstrate substantial predictive content. Nonlinear models (RF and GBRT) significantly improve performance when using Trade Variables (e.g., GBRT R^2 of 1.748% vs. OLS R^2 of 0.762%), reaffirming the importance of nonlinearity for modeling trade impact. Book Variables, conversely, show more modest gains from nonlinearity and yield relatively high R^2 even with linear models (e.g., OLS R^2 of 0.905%), supporting the notion of a more linear relationship between the immediate LOB state and short-term price movements. Most notably, the Order History Variables group (VPIN, Amihud Lambda, Trade Imbalance, etc.) exhibits essentially zero or negative predictive power across all modeling frameworks when used in isolation. Furthermore, comparing the model using 'Trade+Book Variables' to the 'All Variables' model reveals almost identical R^2 values (e.g., for GBRT: 2.282% vs. 2.277%). This implies that, within our prediction horizon and using these specific historical proxies, order history offers no marginal predictive information beyond that contained in contemporaneous trade and LOB features, regardless of modelling approach.

This finding can be interpreted through two different theoretical lenses. From an Efficient Market Hypothesis (EMH; Fama, 1970, 1991) perspective, if markets rapidly incorporate information, the current market snapshot (recent trades and LOB state) should subsume all relevant history for short-term prediction, rendering lagged measures redundant. Our results for these standard order history proxies are consistent with this view at the high frequency. Alternatively, market dynamics may exhibit path-dependency. For instance, Lillo and Farmer (2004) document long memory in order flow signs, and theo-

retical frameworks like Roşu (2009) explicitly model LOB evolution as path-dependent. From this perspective, the failure of VPIN and Trade Imbalance here may not negate the importance of history, but rather suggest these specific, widely-used variables are insufficient proxies for the relevant historical dynamics influencing high-frequency price changes. This motivates our subsequent analysis using more granular historical data.

Table 3.4: Explanatory Power of Feature Groups.

The table reports the out-of-sample R^2 for OLS and GBRT across all feature groups. For each group of features, we fit OLS and GBRT models using all features in that group and report their corresponding out-of-sample R^2.

	OLS	LASSO	RF	GBRT
Trade Variables	0.762	0.756	1.614	1.748
Book Variables	0.905	0.906	0.827	0.974
Order History Variables	-0.086	-0.065	-0.269	-0.028
Trade+Book Variables	1.537	1.559	2.007	2.282
All Variables	1.534	1.556	2.003	2.277

3.5.2 Information Content of Granular Market Data

Having established that nonlinearities and interactions in standard microstructure measures enhance predictive accuracy and established order-history variables contribute little predictive power, we now turn to the central analysis of this chapter: do these engineered variables exhaust the information embedded in the high-frequency data stream, and whether the price discovery process is path-dependent? Specifically, we assess whether models trained directly on raw L3 order book snapshots and their recent history can outperform those based on conventional variables, thereby avoiding any information loss from common feature engineering practice. To this end, we evaluate four input representations:

- **Book and Trade Variables:** The ten conventional trade- and book-level variables.
- **All Variables:** The ten trade and book variables combined with four order-history variables.

- **Market Snapshot:** Raw L3 data at time t, comprising the most recent trade details and the prices, sizes, and order counts at the first ten LOB levels (a 63-element vector).

- **Market History:** A sequence of the 50[c] most recent Market Snapshots preceding t (a 50×63 matrix).

Given the high dimensionality and low signal-to-noise ratio of granular market data, we employ DL models for this analysis.[d] Table 3.5 reports the out-of-sample R^2 for each stock trained with four different input formats, and Table 3.6 presents the paired t-test statistics comparing their R^2.

Market Snapshot

The DL results first confirm the findings from Section 3.5.1 that established order-history variables contribute no incremental predictive power: the average out-of-sample R^2 for 'All Variables' (2.227%) is statistically indistinguishable from that of 'Book+Trade Variables' (2.203%), as confirmed by the paired t-test ($p = 0.485$). Second, the predictive performance derived from the 'Market Snapshot' is statistically equivalent to that obtained from the conventional 'Book+Trade Variables'. The average R^2 for the Market Snapshot (2.288%) is only slightly higher than for Book+Trade Variables (2.203%) which can attribute to training randomness, and the difference is not statistically significant ($p = 0.421$). This implies that the conventional set of established microstructure features effectively distills the relevant predictive information contained within the instantaneous market data. Feature engineering, in this context, does not appear to sacrifice significant predictive power regarding the current market condition. Finally, as shown in the gradient-based feature-importance map for the DL model trained on Market Snapshots (Figure 3.1a), the model attributes predominant importance to three inputs only: last-trade size and Level 1 bid and ask sizes. Notably, the latter two inputs are precisely the

[c] We select 50 steps with sufficient redundancy; as shown in the feature-importance analysis later, predictive contributions beyond ten snapshots are marginal.

[d] We provide a detailed discussion of the merits and limitations of other modeling approaches in Section 3.6.1.

Table 3.5: Predictive Performance Comparison of Input Formats using Deep Learning.

The table reports the out-of-sample R^2 (in percent) for deep learning models trained to predict short-term price changes using different input data representations. 'Book+Trade Variables' uses 10 contemporaneous engineered features. 'All Variables' uses all 13 engineered features. 'Market Snapshot' uses raw L3 data (10 levels LOB + last trade) at time t. 'Market History' uses a sequence of the last 50 Market Snapshots. Results are shown for each of the 20 sample stocks and the average.

Stock	Book+Trade Variables	All Variables	Market Snapshot	Market History
ALL	1.981	1.994	2.124	4.294
ANZ	1.186	1.269	1.183	1.811
BHP	1.541	1.117	1.384	2.374
CBA	2.107	1.847	1.959	2.958
CSL	5.047	5.067	4.423	5.002
FMG	0.998	1.138	1.121	1.978
GMG	0.894	0.936	0.944	2.498
MQG	4.336	4.559	4.850	5.522
NAB	1.752	1.845	2.046	2.404
QBE	1.146	1.137	1.129	2.884
REA	3.716	3.899	4.427	8.597
RIO	3.061	3.214	2.995	4.360
RMD	1.148	1.133	1.148	1.939
STO	1.662	1.576	3.019	5.374
TCL	2.162	2.250	1.887	3.930
WBC	2.195	2.218	1.812	2.622
WDS	2.154	2.044	1.547	3.148
WES	2.348	2.384	2.337	3.963
WOW	2.265	2.389	2.358	3.705
WTC	2.354	2.530	3.079	4.998
Average	**2.203**	**2.227**	**2.288**	**3.718**

components required to compute the engineered Level 1 Depth Imbalance variable. This observation yields two inferences. First, the DL model can implicitly learn to reconstruct signals analogous to engineered variables directly from raw data. Second, it corroborates our earlier findings based on established variables: when considering only current market status, the short-term price dynamics are driven primarily by the last trade size and the LOB imbalance at the top level.

Table 3.6: Paired t-test results for R^2 differences between input formats

The table shows results from paired t-tests comparing the out-of-sample R^2 values (from Table 3.5) across the 20 stocks for different input format pairs. ΔR^2 is the average difference in R^2 (in percent). 'ns' denotes not significant at the 5% level; '***' denotes significance at the 0.1% level.

Comparison	ΔR^2 (%)	t-statistic	p-value	Sig.
All Variables vs. Book+Trade Variables	+0.025	−0.712	0.485	ns
Market Snapshot vs. Book+Trade Variables	+0.086	−0.822	0.421	ns
Market Snapshot vs. All Variables	+0.061	−0.611	0.549	ns
Market History vs. Book+Trade Variables	+1.515	−5.850	< 0.001	***
Market History vs. All Variables	+1.491	−5.886	< 0.001	***
Market History vs. Market Snapshot	+1.429	−7.364	< 0.001	***

Market History

While DL models trained on raw market snapshots perform no better than DL models trained on microstructure variables, large differences do occur for DL models trained on historical market snapshots. The average out-of-sample R^2 rises to 3.718%, representing an approximate 68% increase relative to the 'All Variables' model and a 62% increase relative to the 'Market Snapshot' model. These gains are highly statistically significant ($p < 0.001$) and are observed consistently across nearly all individual stocks, providing strong evidence that the temporal evolution of the LOB and trade activity contains crucial predictive information that is not captured by established microstructure variables or even the most detailed market snapshot.

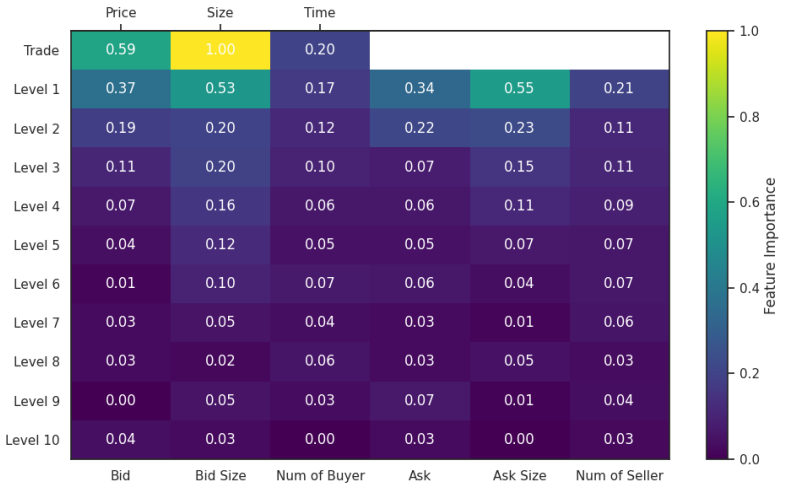
Further insights arise from comparing the feature importance patterns across input format. In the market snapshot model (Figure 3.1a), size related variables such as trade size and best bid/ask depths dominate. However, when historical information is incorporated (Figure 3.1b), price related features, including trade prices and bid-ask spreads, gain in relative importance. This shift is consistent with the hypothesis that while absolute prices have limited informativeness, their evolution over time reflects changing supply and demand conditions, liquidity imbalances, and short-term directional pressure.

A more detailed decomposition is presented in Figure 3.2, which plots feature importance across both features and time lags in the Market History model. We find that align with the economic intuition that the most recent snapshot (lag 1) consistently exhibits

Figure 3.1: Feature Importance Comparison: Market Snapshot vs. Market History.

The panels compare feature importance (FI) for deep learning models using two input formats. FI is computed as the magnitude of the input gradient with respect to the output prediction, averaged over all test samples, and normalized using min-max scaling to [0,1] within each panel. Higher values indicate greater local importance. **Row 1:** Trade features (Price, Size, Time since last trade). **Rows 2-11:** Book features (Bid Size, Bid Price, Ask Price, Ask Size) for Levels 1 through 10. For the Market History input (Panel b), FI values are additionally averaged across the 50 time steps.

(a) Market Snapshot (Importance at time t)



(b) Market History (Importance averaged over $t - 49$ to t)

the highest importance, with a gradual decay in influence as the lag increases—indicating that while past states are informative, their relevance diminishes with temporal distance. However, this decay pattern differs across feature types. As shown in Figure 3.3, which plots the temporal decay of FI and trade features and L1 book features, trade-related features (like trade size and price) exhibit a longer memory effect, retaining small influence

even after 50 ticks, whereas book-related features (like L1 sizes) show a faster decay, losing most influence beyond 20 ticks. The differential decay patterns accord with established views on the relative informational content of trades and quotes. The protracted memory embedded in trade variables is consistent with the permanent price adjustments implied by the continuous-auction models of Kyle (1985) and Glosten and Milgrom (1985), as well as the empirical evidence of persistent order-flow and trade-sign dependence in Lillo and Farmer (2004) and Bouchaud et al. (2009). By contrast, the quicker attenuation of book-feature importance reflects the transitory nature of limit-order book states, which primarily capture liquidity-provision and inventory-management effects (O'Hara, 1998) and short-lived supply–demand imbalances that dissipate rapidly (Cont et al., 2010).

Overall, our findings challenge the implicit Markovian assumption in many canonical market microstructure models, in which the conditional distribution of future prices depends solely on the current market state. Instead, our results strongly support growing theory models that posit price formation exhibits path dependence and that order flow possesses persistent memory. For example, Roşu (2009)'s theoretical work explicitly models path-dependent features. Riccò et al. (2023) propose a non-stationary model of a limit order market explicitly incorporating time dynamics and path dependency on the price impact of orders. Taken together with the results from models using static snapshots, our findings suggest that existing theoretical models and empirical proxies may have largely exhausted the predictive content of contemporaneous market state. Further theoretical development should prioritize modeling the dynamics of order flow and book evolution, which appear to contain economically and statistically significant predictive structure that is currently underexplored.

3.6 Model Choice

Model selection in high-frequency finance is often approached in an ad hoc manner, with researchers and practitioners frequently deploying various machine learning and deep learning methods without a rigorous justification for why one method might be preferable

Figure 3.2: Full Feature Importance Map for Market History Input.

The heatmap displays the feature importance (FI) for the deep learning model trained on the 'Market History' input (sequence of 50 snapshots). The x-axis represents the time lag (0 = current snapshot, 49 = oldest snapshot in sequence). The y-axis represents the input features (63 dimensions: 3 trade features + 10 levels * (bid size, bid price, ask price, ask size)). FI is computed as the average input gradient magnitude, log-normalized to [0,1] across all features and lags. Brighter colors indicate higher importance. Note the general decay in importance as lag increases and the concentration of importance on recent lags and features near the top of the book.

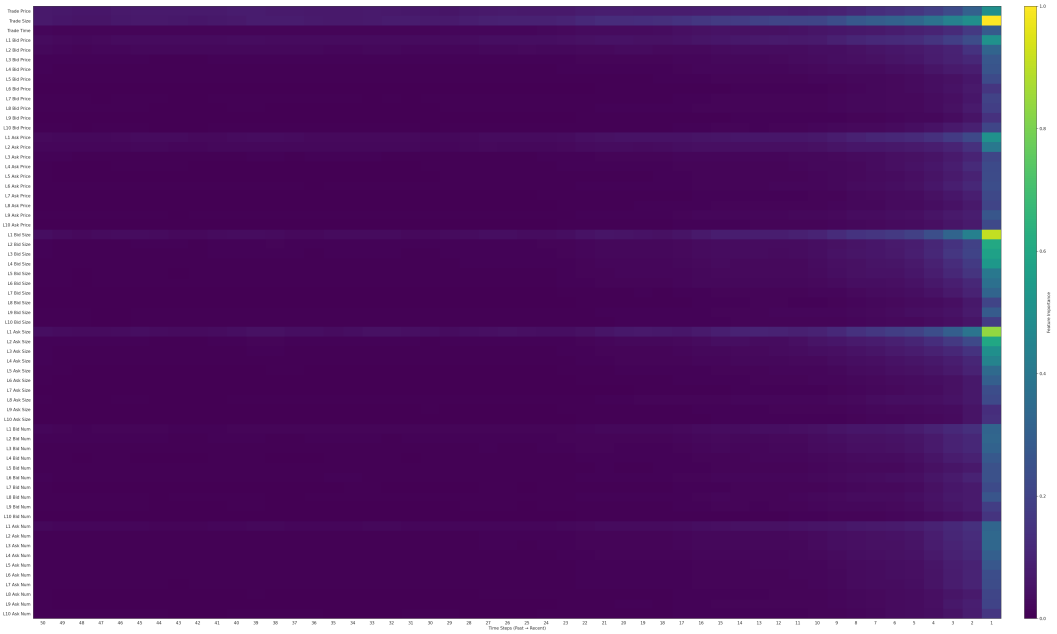
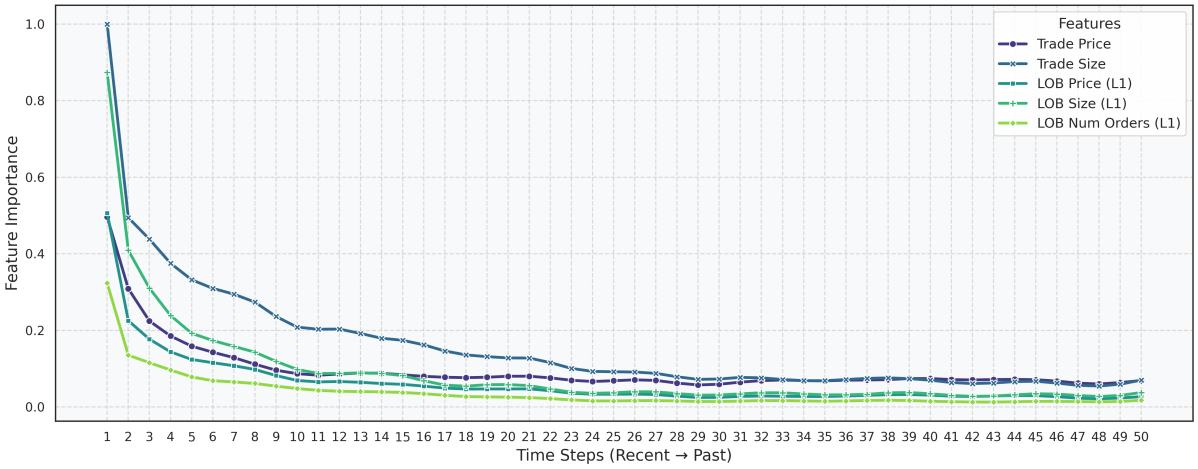


Figure 3.3: Feature Importance Decay for Trade and L1 Book Features.

The Figure plots the feature importance (FI) of trade size, trade price, trade time delta, and Level 1 book features (price, size, order count) across the 50-step lookback window for the Market History model. Book feature FI is averaged across the bid and ask sides. Note the longer persistence of importance for trade features compared to book features.



over another. In what follows, we confirm that our previous model choices are appropriate and discuss the suitability of each model class for specific use cases to provide guidance for future research.

3.6.1 Model Comparison

Here, we investigate how different ML models perform depending on both the amount and format of the input data. Specifically, we use five different models: two linear models (OLS and LASSO), two ML models (RF and GBRT), and the transformer model. Each model is then estimated on three different data formats. First, on the existing microstructure variables used in Section 3.5.1. Second, on the raw current market snapshots, while the third is estimated on the complete raw market history data used in Section 3.5.2. Table 3.7 reports the out-of-sample R^2 values achieved by each model when estimated using the three difference data formats.

Table 3.7: Out-of-Sample R^2 Across Models and Input Formats

This table presents the out-of-sample R^2 for different models trained on various input formats. All hyperparameters have been tuned to the best validation performance; a further discussion on robustness to hyperparameters is provided in the next section.

	Variable					Market Snapshot					Market History				
	OLS	LASSO	RF	GBRT	DL	OLS	LASSO	RF	GBRT	DL	OLS	LASSO	RF	GBRT	DL
ALL	1.431	1.399	1.674	2.260	1.814	1.236	1.214	1.615	1.668	2.412	-0.005	1.933	1.718	1.961	4.378
ANZ	0.505	0.567	0.756	0.987	1.282	0.661	0.663	0.648	1.174	1.479	-0.150	1.079	0.522	1.363	1.828
BHP	1.229	1.347	1.472	1.623	0.639	1.297	1.378	1.372	1.805	1.506	-0.030	1.795	1.405	1.928	1.871
CBA	1.042	1.068	1.914	2.123	1.721	1.133	1.174	1.915	2.359	2.160	0.010	1.723	1.862	2.365	2.517
CSL	1.982	2.003	3.583	4.357	5.197	1.920	1.844	3.027	3.684	4.316	0.006	3.091	3.325	3.682	5.353
FMG	0.704	0.834	1.031	1.052	0.613	0.683	0.735	1.102	1.042	1.275	-0.205	0.998	1.094	0.766	2.253
GMG	0.960	0.983	0.954	0.994	0.976	1.009	1.037	1.274	0.767	1.469	-0.165	1.204	1.282	0.864	2.519
MQG	2.376	2.316	3.519	4.323	4.631	2.073	1.997	2.877	3.903	4.287	-0.067	3.148	2.939	4.179	6.232
NAB	1.608	1.617	1.546	1.813	1.859	1.193	1.195	1.236	1.707	2.128	-0.141	1.609	1.210	1.629	2.709
QBE	1.345	1.359	1.148	1.308	1.212	1.185	1.208	1.227	1.201	1.561	-0.183	1.542	1.210	1.176	2.846
REA	2.437	2.412	3.793	4.322	4.194	2.245	2.199	3.014	3.663	3.223	0.242	3.731	3.364	3.752	8.386
RIO	1.400	1.387	2.473	2.872	3.081	1.258	1.301	2.193	2.761	3.061	-0.032	2.294	2.156	2.998	4.881
RMD	0.660	0.689	0.765	1.102	1.036	0.501	0.551	0.658	0.702	0.671	-0.001	0.970	0.651	0.400	0.783
STO	3.694	3.691	3.381	3.794	1.684	2.795	2.781	2.199	2.579	2.526	0.014	3.119	2.368	1.929	5.117
TCL	2.261	2.247	2.162	2.287	2.403	1.604	1.719	1.512	1.078	2.370	0.143	1.713	1.386	2.132	3.848
WBC	1.976	2.008	1.786	2.213	2.244	1.483	1.485	1.505	2.079	2.302	-0.161	1.804	1.515	2.105	2.702
WDS	0.975	1.019	1.643	1.783	2.033	0.264	0.489	1.654	1.797	1.752	-0.203	0.956	1.651	1.728	3.156
WES	1.074	1.134	1.773	1.853	2.479	1.174	1.128	1.635	1.721	2.202	-0.080	1.889	1.678	1.722	4.132
WOW	1.354	1.368	1.750	1.973	2.370	1.179	1.152	1.658	1.842	2.284	-0.101	1.992	1.633	1.858	4.013
WTC	1.664	1.665	2.984	3.053	2.618	1.387	1.374	2.413	2.661	2.335	-0.090	2.047	2.432	2.628	4.933
Average	1.534	1.556	2.005	2.305	2.204	1.314	1.331	1.737	2.010	2.266	-0.060	1.932	1.770	2.058	3.723

Comparison of Linear Models:

OLS offers the advantages of interpretability and computational simplicity. However, its restrictive linearity assumption can materially limit explanatory power. Across all three input formats we consider, we find that OLS consistently under performs nonlinear alternatives. A further limitation of OLS is its lack of regularization, which results in the inclusion of all explanatory variables irrespective of their informativeness. While this drawback is less consequential in low-dimensional settings, it becomes increasingly problematic as the number of explanatory variables grows. In high-dimensional settings, such as those using raw market history data, OLS exhibits a tendency to overfit. This pattern is reflected in Table X, where the average R-squared for OLS is highest when the model is estimated on established microstructure variables, which is when the number of explanatory variables is least. However, as we increase the number of explanatory variables by using market snapshots or extended market histories, the average R-squared declines sharply, even falling below zero in the latter case.

In contrast, the LASSO estimator addresses this issue by introducing an L_1 penalty term that effectively performs variable selection by shrinking the coefficients of irrelevant predictors toward zero. This regularization mechanism is particularly valuable in high-dimensional settings, where the risk of overfitting is most pronounced. Our empirical results confirm this intuition. Specifically, when the model is estimated using only microstructure variables, the average R-squared for LASSO is 1.556, closely aligning with that of OLS at 1.534. However, when the number of explanatory variables is dramatically expanded by using raw market history data, the performance gap widens significantly. Notably, LASSO yields an average R-squared of 1.893, whereas OLS deteriorates to –0.059. These findings underscore the importance of regularization for mitigating overfitting in models with large numbers of explanatory variables.

Comparison of Machine Learning Models

When nonlinear relationships are present, RF and GBRT are popular choices due to their ability to model complex interactions (Friedman, 2001). RF mitigates variance

through bootstrapping, in which each tree is trained on a bootstrap sample of the data, producing an ensemble of decorrelated models. In contrast, GBRT adopts a boosting paradigm, fitting successive shallow trees to the residuals of prior models. This iterative refinement enables GBRT to more effectively capture intricate structure in the data and, in many applications, yields superior predictive performance with less susceptibility to overfitting. While bootstrapping is commonly associated with RF, it is not exclusive to it. In our implementation, we employ subsampling in GBRT. Specifically, each weak learner is trained on 50% of the data, to further reduce variance. Empirically, we find that GBRT consistently outperforms RF across all three data input formats. Despite their architectural differences, the two approaches require comparable computational resources and effort in hyperparameter tuning. As such, GBRT is almost always the preferred choice when modeling nonlinear dependencies patterns.

Comparison of ML and DL Models:

DL models offer an alternative framework for capturing nonlinear relationships. While both ML and DL are capable of modeling such interactions, ML methods often demonstrate superior performance on small to medium-sized datasets, while DL methods perform better on large datasets. This difference is because ML has comparatively fewer parameters, which reduces the risk of overfitting when the data set is small. Accordingly, in empirical finance applications that involve relatively few observations (e.g., daily observations), ML models frequently outperform DL approaches, which generally require large sample sizes to reliably estimate complex function classes (Gu et al., 2020). In contrast, in high-frequency market microstructure settings, where datasets often contain millions of observations, the overfitting concerns associated with DL are mitigated, narrowing the performance differential between ML and DL.

Given the above, ML models tend to perform particularly well when supplied with carefully engineered explanatory variables. In our empirical results, Gradient Boosted Regression Trees (GBRT) perform comparably or even exceed DL models when applied to structured inputs derived from domain knowledge. In such cases, the relative ease

of model estimation and tuning makes ML methods a practical and efficient choice for modeling nonlinear relationships among curated features.

By contrast, when presented with raw market history data, ML models tend to underperform. As discussed in the methodology section, ML models based on decision trees lack the architectural depth to learn complex representations from high dimension raw data, especially when temporal dependencies are present. For instance, a market history consisting of 50 time steps with 63 variables per step yields an input space comprising of 3,150 variables. The simplistic architecture of traditional ML models limits their ability to extract meaningful patterns from such intricate, noisy data. Moreover, without an inherent variable selection mechanism like that in LASSO, the abundance of redundant variables not only overwhelms ML learning capacity but also increases the risk of overfitting. Consistent with this, we find that while both LASSO and DL models exhibit improved performance with market history inputs, RF and GBRT show limited gains or, in some cases, deteriorating performance.

In contrast, due to their extensive parameterization and architectural flexibility, deep learning (DL) models are capable of learning complex, nonlinear representations directly from raw input data, provided that a sufficiently large dataset is available. Accordingly, in our empirical analysis, the DL model consistently outperformed both Random Forests (RF) and Gradient Boosted Regression Trees (GBRT). This advantage was particularly pronounced when models were trained on raw market history inputs. These findings highlight the suitability of DL methods for applications in market microstructure, where large volumes of granular data and complex temporal dependencies render traditional approaches based on handcrafted variables comparatively less effective. It is important to emphasize that, although both ML and DL methods capture nonlinearities, they represent fundamentally different modeling paradigms. ML methods are essentially nonlinear extensions of linear models and adhere to a *theory-driven approach* that relies on human-engineered variables derived from well-established financial theories. Within this framework, DL offers little additional benefit while incurring extra computational cost. By contrast, DL embodies a *data-centric approach* in which the model learns patterns

directly from the rich data environment, reducing the need for extensive variable engineering. This approach enables DL models to uncover subtle patterns in order history and price discovery that are often overlooked by conventional methods. As high-frequency data becomes increasingly abundant, the ability of DL architectures to learn directly from raw inputs offers significant advantages in both predictive accuracy and insight into market microstructure dynamics.

Overall, the choice of model in market microstructure research should be guided by careful consideration of both the input format and the available data volume. Linear models are effective for establishing a theoretical framework and serving as robust baselines, ensemble ML methods effectively capture moderate nonlinearities, and a data-centric DL approach is indispensable for uncovering complex patterns that traditional variable engineering may miss in high-dimensional, raw data.

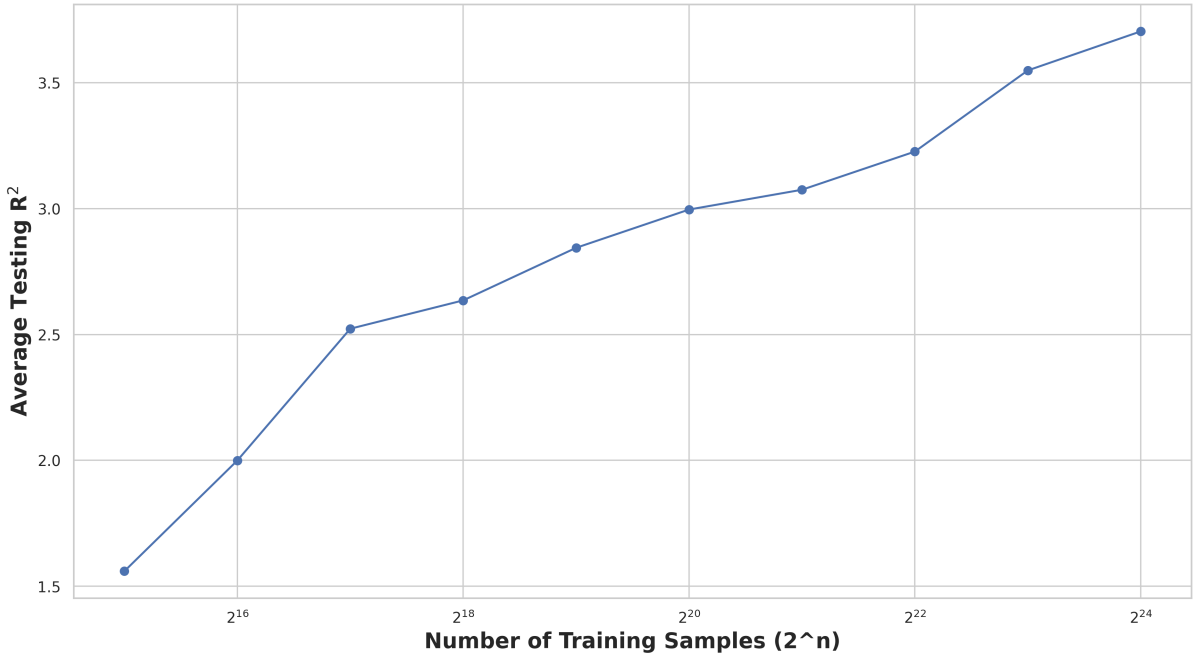
3.6.2 Data Scaling Effect

We find that existing finance research using DL models often draws conclusions without sufficiently considering the size of the training data. As discussed earlier, the success of DL applications is driven by data size more than by model architecture or hyperparameter settings. This phenomenon is well documented in other fields such as computer vision (Zhai et al., 2022) and natural language processing (Kaplan et al., 2020). To illustrate, we trained ten identical models with the same architecture and hyperparameters but only varying training data sizes, from 2^{15} to 2^{24} samples. We doubled the training data size each run. Figure 3.4 illustrates the clear data-centric nature of DL methods: as the training set size increases, the predictive performance of the DL model improves significantly. Much existing research may underestimate DL's capabilities by not sufficiently accounting for training data volume, a factor often more critical than model architecture or hyperparameters. The insufficient training data can lead to false conclusions, such as underestimating DL capabilities relative to traditional machine learning (ML) models or, in our case, determining that there is no additional predictive power of market history data. The popularity of Transformers as the backbone of modern large language models

(e.g., ChatGPT) is not only due to their superior handling of sequential relationships, but also their computational efficiency on modern GPU-powered data centers compared to traditional RNN-based models. This efficiency enables both model and data size to scale significantly, yielding unprecedented performance. Given the vast amounts of financial data available, especially in microstructure research, fully utilizing the potential of DL methods remains a promising research direction.

Figure 3.4: Data Scaling Effect of DL models

The figure displays the average out-of-sample R^2 of Transformer models trained on varying data sizes, ranging from 2^{15} to 2^{24} samples. All models share the same architecture and hyperparameters, with data size being the only variable.



3.7 Conclusion

In this chapter, we re-evaluated the price discovery process using DL techniques.We systematically compare the predictive content of established, theory-motivated microstructure variables with raw, high-frequency L3 data. Our objective is to determine what new insights into market dynamics can be gained from data-centric DL methods and to provide practical guidance on their use. Our analysis yields three key results. First, we confirm that significant nonlinearities and interaction effects exist within established microstruc-

ture variables. The predictive power of trade-related variables, such as signed trade size, nearly doubles when modeled with nonlinear techniques. Furthermore, interactions between directional order flow and market condition variables are critical, supporting the view that liquidity and volatility act as conduits for price discovery (Kwan et al., 2024).

Second, we find a crucial distinction between the information content of contemporaneous and historical data. The established suite of microstructure variables effectively captures the predictive information in a contemporaneous market snapshot; a deep learning (DL) model trained on raw L3 data offers no significant performance gain. However, these variables fail to incorporate the rich temporal information embedded in the recent history of order flow. A DL model trained on a sequence of historical L3 data improves predictive accuracy by over 60% compared to all other specifications. This result provides strong empirical evidence that price formation is a non-Markovian process. The market possesses a memory that is not captured by conventional static or memoryless proxies, a finding that supports emerging theories on path-dependent price discovery (Ricco, 2023).

Third, our research provides clear guidance on methodological choice in empirical finance. We frame the choice as one between a theory-driven approach, where machine learning models like gradient-boosted trees excel at modeling nonlinearities among engineered features, and a data-centric approach, where DL models are uniquely capable of learning complex temporal patterns directly from large, raw data streams. We demonstrate that the efficacy of DL models is highly dependent on data scale, cautioning that studies using smaller datasets may incorrectly conclude that simpler ML models are superior.

The main implication of this chapter is twofold. First, the theoretical developments have already captured most information in the contemporaneous market state, future research should focus on the path-dependent dynamics of order flow. Second, DL offers a data-centric modeling paradigm that, combined with large-scale raw data, can enhance our understanding of market microstructure and guide future research directions.

Chapter 4

Hybridizing Econometric and NN Models

4.1 Introduction

Volatility modeling is an active area of research in financial econometrics with implications for risk management, portfolio allocation, and option pricing. The GARCH model (Bollerslev, 1986; R. F. Engle, 1982), and its variants including the exponential GARCH (Nelson, 1991) and the GJR model (Glosten et al., 1993), provide an excellent modeling framework for understanding and forecasting volatility. While GARCH and its derivatives work well in many scenarios, when it comes to daily volatility modeling, they predominantly rely on the signal in daily squared returns. This can sometimes fall short in capturing swift intraday volatility fluctuations. This problem can be mitigated by incorporating more effective volatility proxies based on high-frequency intraday return data.

The GARCH models of R. F. Engle (1982) and Bollerslev (1986), and their variants, such as the exponential GARCH of Nelson (1991) and the GJR model of Glosten et al. (1993), are widely used in traditional financial econometric literature. The GARCH framework models the current conditional variance of the return as a linear function of past conditional variances and squared returns. However, traditional GARCH-type

models based on daily returns respond slowly to rapid changes in volatility, which can take several periods to reach a new level. This problem can be mitigated by incorporating more effective volatility proxies based on high-frequency return data.

In the past two decades, many ex-post estimators of asset return volatility using high-frequency data have been introduced to the literature. Examples include the realized variance of Andersen and Bollerslev (1998), realized kernel variance of Barndorff-Nielsen et al. (2008) and bipower variation of Barndorff-Nielsen and Shephard (2004). These estimators, collectively referred to as realized volatility measures, are more informative than daily squared returns about representing the underlying volatility level, thus providing a better tool for modeling and forecasting volatility. R. Engle (2002) explored the idea of including realized volatility measures in the GARCH model and found that it significantly improves its fit to return data. Engle's model only uses a realized volatility measure as a deterministic input to the GARCH equation and pays no attention to explaining the variation in realized volatility measures which should be viewed as noisy proxies of the underlying volatility.

Hansen et al. (2012) developed a complete model, called the realized GARCH (RealGARCH) model, that incorporates a realized measure into a GARCH framework and at the same time provides a measurement equation to explain the realized volatility measure dynamics. This model has been proven superior empirically (W. Jiang et al., 2018; Li et al., 2021; Xie & Yu, 2020). Hansen and Huang (2016) further extended the RealGARCH model to realized EGARCH which incorporates multiple realized volatility measures of volatility. The RealGARCH model expresses the underlying volatility as a *linear* combination of several lagged realized measures. This approach might fall short in grasping the nonlinear relationships and enduring impacts that prior realized measures have on the present intrinsic volatility. This issue is similar to the well-known problem of GARCH that has led to the development of long-memmory and non-linear GARCH models such as Recurrent Conditional Heteroskedasticity (RECH) of Nguyen et al. (2022).

This chapter aims to extend RealGARCH and address its limitation by leveraging modern deep learning techniques together with recent advancements in Bayesian com-

putation techniques. With the advent of deep learning models and advancements in computational power, neural networks (NN) have recently been introduced in volatility modeling in the mainstream econometric literature. NNs are capable of learning complex non-linear functions and capturing long-range dependence in time series data. Y. Liu (2019) and Bucci (2020) compared the predictive performance of feed-forward neural networks (FNN) and recurrent neural networks (RNN) with traditional econometric approaches, and found that deep learning models generally outperform econometric models on several stock markets. Some hybrid models, proposed by Hyup Roh (2007) and Kim and Won (2018), add a neural network as another layer on top of an econometric model, using the volatility estimates produced by an econometric model as the input to a neural network, which then outputs the final estimate of the volatility. Although these models perform well on specific stock datasets, they are often engineering-oriented and lack interpretability in a financially or economically meaningful way, ignoring important stylized facts such as the leverage and clustering effects commonly observed in financial time series data. However, as FNNs are typically designed for cross-sectional data analysis, the FNN component in the FNN-GJR model might be inefficiently at capturing serial dependence in financial time series.

An exception is the recurrent conditional heteroscedastic model (RECH) of Nguyen et al. (2022), that provides a flexible framework for combining deep learning with GARCH-type models. The RECH model represents the volatility as a sum of two components. The first component is governed by a GARCH-type model that retains the characteristics and interpretability of econometric models. The second component is governed by a RNN that can capture non-linear and long-term serial dependence structure in financial time series. As demonstrated in Nguyen et al. (2022), RECH retains much of the interpretable characteristics from the GARCH model, while enjoying the modeling flexibility and prediction accuracy from RNN.

This chapter proposes a new approach to volatility modeling by combining deep learning and realized volatility measures. Our framework, the Realized Recurrent Conditional Heteroskedasticity (RealRECH) model, incorporates and distills modeling advances from

financial econometrics, high frequency trading data and deep learning. Inspired by RECH, we incorporate the long short-term memory model (LSTM) of Hochreiter and Schmidhuber (1997) into RealGARCH. The LSTM architecture is one of the most advanced and sophisticated RNN techniques and has proven highly efficient for time series modeling. By incorporating LSTM into RealGARCH, we unlock its modeling power and allow it to be able to capture complex underlying dynamics in financial volatility. In this sense, the RealRECH model can be viewed as an extension of RealGARCH using deep learning. The LSTM architecture is one of the most advanced and sophisticated RNN techniques and has proven highly efficient for time series modeling. By incorporating LSTM into RECH, we unlock its modeling power and allow it to be able to capture complex underlying dynamics in financial volatility. We compare the performance of the new model with several existing benchmark models on 31 stock market indices. We show that the new model substantially improves on previous approaches in both in-sample fit and out-of-sample forecasting.

4.2 Model Formulation

4.2.1 Conditional heteroscedastic models and realized volatility measures

Let $\boldsymbol{y} = \{y_t, t = 1, \ldots, T\}$ be a time series of daily returns. The key term of interest in volatility modeling is the conditional variances, $\sigma_t^2 = \text{var}\left(y_t \mid \mathcal{F}_{t-1}\right)$, where $\mathcal{F}_{t-1}$ denotes the σ-field of information up to and including time $t-1$. We assume here that $\mathbb{E}(y_t|\mathcal{F}_{t-1}) = 0$, but the present method is easily extended to model the conditional mean $\mathbb{E}(y_t|\mathcal{F}_{t-1})$. The GARCH model expresses the conditional variance σ_t^2 as a linear combination of the previous squared returns and conditional variances as an ARMA(p, q)

model:

$$y_t = \sigma_t \epsilon_t, \quad \epsilon_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1), \quad t = 1, 2, \ldots, T \tag{4.1}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \quad t = p+1, \ldots, T. \tag{4.2}$$

The restriction $\omega > 0, \alpha_i, \beta_j \geq 0, i = 1, \ldots, p, j = 1, \ldots, q$ is used to ensure positivity of σ_t^2, and $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$ is needed to ensure the stationarity of the time series y_t. The errors ϵ_t are independently and identically distributed as normal distributions with zero mean and unit variance; other distributions for ϵ_t such as Student's t are also considered in the literature, e.g., Gerlach and Wang (2016). For other GARCH-type models, the reader is referred to Nelson (1991), Glosten et al. (1993) and Bollerslev (2008). The GARCH model relies on daily squared returns, which only contain a weak signal of the daily volatility σ_t^2. It is widely known in the financial econometric literature that high-frequency return data, such as 5-minute data, can be used to estimate daily volatility with high accuracy. In the past twenty years, many estimators of daily volatility using high-frequency data were developed and are referred to as "realized volatility measures" (Andersen & Bollerslev, 1998; Barndorff-Nielsen & Shephard, 2004; Barndorff-Nielsen et al., 2008). As realized volatility measures are ex-post, they cannot be directly used for volatility forecasting but they are effective volatility proxies for volatility modeling. R. Engle (2002) is among the first to explore this idea by incorporating the realized volatility measure of Andersen and Bollerslev (1998) into the GARCH model. Since then, many volatility models incorporating realized volatility measures have been developed, e.g., Forsberg and Bollerslev (2002), R. F. Engle and Gallo (2006), Corsi (2009), Shephard and Sheppard (2010). The realized GARCH model (RealGARCH) of Hansen et al. (2012)

$$y_t = \sigma_t \epsilon_t, \quad t = 1, 2, \ldots, T \tag{4.3a}$$

$$\sigma_t^2 = \omega + \gamma \mathrm{rv}_{t-1} + \beta \sigma_{t-1}^2 \tag{4.3b}$$

$$\mathrm{rv}_t = \xi + \varphi \sigma_t^2 + \tau(\epsilon_t) + u_t \tag{4.3c}$$

is an important development in this direction. Here $\epsilon_t \overset{i.i.d}{\sim} N(0,1)$, $u_t \overset{i.i.d}{\sim} N(0,\sigma_u^2)$, rv_t is a realized volatility measure, and $\tau(\epsilon)$ is regarded as the leverage function and used to capture the leverage effect often observed in volatility. Hansen et al. (2012) set $\tau(\epsilon) = \tau_1\epsilon + \tau_2(\epsilon^2 - 1)$. An attractive feature of the RealGARCH model is that it contains the measurement equation (4.3c) that accounts for the variation in the realized volatility measure rv_t. It associates the observed realized volatility measure with the underlying latent volatility, in which the integrated high-frequency variance rv_t is explained as a linear combination of σ_t^2 plus a random innovation. This chapter employs a simple yet effective realized volatility measure, the 5-minute realized variance (RV_5) of Andersen and Bollerslev (1998). Suppose that we observe the asset price at n trading times within a trading day t, $t_j = t - 1 + j/n, j = 1, \ldots, n$. Let $\{P(t_j), j = 1, ..., n\}$ be the observed prices and $r_{t_j} = \log P(t_j) - \log P(t_{j-1})$ be the log-returns. The RV for the trading day t is defined as

$$\text{rv}_t := \sum_{j=1}^{n} r_{t_j}^2.$$

It can be shown that (Andersen & Bollerslev, 1998), as $n \to \infty$, rv_t converges in probability to the true latent variance σ_t^2. For RV_5, the return r_{t_j} in the above equation is recorded at 5 minutes frequency. There are various definitions of realized volatility measures but there is little evidence that any outperform RV_5 as a volatility proxy; see L. Y. Liu et al. (2015) for a detailed comparison of more than 400 realized volatility measures.

4.2.2 Recurrent Neural Network

RNN is a special class of neural network designed for modeling sequential data. Let $\{D_t = (x_t, y_t), t = 1, 2, ...\}$ be the data with x_t the input and y_t the output. We use (x_t, y_t) in this section as generic notation, not necessarily applicable to the return data in other sections. The task is to model the conditional mean $\widehat{y}_t = \mathbb{E}(y_t | x_t, D_{1:t-1})$. The

basic RNN framework is

$$h_t = g_h\left(W^h\left[h_{t-1}, x_t\right] + b^h\right), \tag{4.4a}$$

$$\widehat{y}_t = g_y\left(W^y h_t + b^y\right). \tag{4.4b}$$

The main feature of the RNN structure is its vector of hidden states h_t which is defined recurrently. At each time t, two information sources are fed into h_t: the historical information stored in h_{t-1} and the current information from the input x_t. The functions g_h and g_y are activation functions such as $\text{sig}(z) := 1/(1+e^{-z})$, or $\tanh(z) := (e^z - e^{-z})/(e^z + e^{-z})$. Finally, the W and b are trainable model parameters.

The basic RNN model in (4.4a)-(4.4b) has some limitations in terms of both modeling flexibility and training difficulty. Many sophisticated RNN structures are proposed to overcome these limitations, and the LSTM model of Hochreiter and Schmidhuber (1997) stands out as one of the most successful methods. LSTM uses a gate structure to control the memory in the data. It is written as follows:

$$g_t^i = \text{sig}\left(W^i\left[h_{t-1}, x_t\right] + b^i\right) \tag{4.5a}$$

$$g_t^f = \text{sig}\left(W^f\left[h_{t-1}, x_t\right] + b^f\right) \tag{4.5b}$$

$$g_t^o = \text{sig}\left(W^o\left[h_{t-1}, x_t\right] + b^o\right) \tag{4.5c}$$

$$\tilde{c}_t = \tanh\left(W^c\left[h_{t-1}, x_t\right] + b^c\right) \tag{4.5d}$$

$$c_t = g_t^i \cdot \tilde{c}_t + g_t^f \cdot c_{t-1} \tag{4.5e}$$

$$h_t = g_t^o \cdot \tanh\left(c_t\right) \tag{4.5f}$$

$$\widehat{y}_t = g_y\left(W^y h_t + b^y\right). \tag{4.5g}$$

Unlike the basic RNN that fully overwrites the memory stored in the hidden states at each step, LSTM can decide to keep, forget or update the memory via the memory cell c_t in (4.5e). This memory cell c_t is updated by partially forgetting the previous memory from c_{t-1} and adding new memory from $\tilde{c}_t$. The extent of forgetting the history and adding new information is controlled by the forget gate g_t^f and input gate g_t^i, respectively. Finally,

the degree of current memory usage for final output is controlled by the output gate. The ability of LSTM to control its memory and quickly adapt to new data patterns makes it very suitable to model volatility dynamics. Section 4.2.3 describes how to incorporate LSTM into volatility modeling.

4.2.3 The Realized Recurrent Conditional Heteroskedasticity Model

This section presents our RealRECH model, that incorporates LSTM into RealGARCH for flexible and accurate volatility modeling. Its key motivation is using an additive component governed by LSTM, that takes the realized measure as its input, to capture complex serial dependence structure in the volatility dynamics. As intraday realized volatility measures are accurate volatility proxies (Andersen & Bollerslev, 1998; Barndorff-Nielsen et al., 2008), feeding them to the LSTM unit further empowers its modeling capacity. The RealRECH model of order p and q, RealRECH(p, q), is written as:

$$y_t = \sigma_t \epsilon_t, \quad t = 1, 2, \ldots, T \tag{4.6a}$$

$$\sigma_t^2 = g(\omega_t) + \sum_{i=1}^{p} \gamma_i \mathrm{rv}_{t-i} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{4.6b}$$

$$\mathrm{rv}_t = \xi + \varphi \sigma_t^2 + \tau\left(\epsilon_t\right) + u_t \tag{4.6c}$$

$$\omega_t = \mathrm{LSTM}\left(x_t\right) \tag{4.6d}$$

$$x_t = \left(\omega_{t-1}, y_{t-1}, \sigma_{t-1}^2, \mathrm{rv}_{t-1}\right). \tag{4.6e}$$

Compared to the RealGARCH model, equation (4.3b), that only allows a linear and short-term dependence of the true latent conditional variance σ_t^2 on the realized volatility measure, the RealRECH model is much more flexible. The latter, via its LSTM component $g(\omega_t)$, allows for both non-linear and long-term dependence that the previous realized volatility measures might have on σ_t^2. As σ_{t-1}^2 is also an input to the LSTM component, RealRECH also allows for complex serial dependence that the previous volatility might have on σ_t^2. We found empirical evidence for the inclusion of ω_{t-1} and y_{t-1} in the input vector x_t. The function g in (4.6b) is a non-negative activation function, applied

to the ω_t to ensure the positive conditional variance. We adopt the RELU function, $g(\omega_t) := \max\{\omega_t, 0\}$, for this purpose.

Inspired by RealGARCH, RealRECH includes the measurement equation (4.6c) to account for the variation in the realized volatility measures. Rather than using the linear structure as in (4.6c), one could easily use an RNN model to explain rv_t based on σ_t^2; we have tried this, however, did not observe any meaningful improvement. We only include one realized volatility measure in the RealRECH model; however, it is easy to incorporate as many realized volatility measures as possible by using them as additional inputs into x_t. We only consider RealRECH(1,1) in this chapter, which, for ease of reading and later cross-reference, is expressed as

$$y_t = \sigma_t \epsilon_t, \quad t = 1, 2, \ldots \tag{4.7a}$$

$$\sigma_t^2 = g(\omega_t) + \gamma \text{rv}_{t-1} + \beta \sigma_{t-1}^2 \tag{4.7b}$$

$$\text{rv}_t = \xi + \varphi \sigma_t^2 + \tau(\epsilon_t) + u_t \tag{4.7c}$$

$$\omega_t = \beta_0 + \beta_1 h_t \tag{4.7d}$$

$$h_t = \text{LSTM}(x_t) \tag{4.7e}$$

$$x_t = \left(\omega_{t-1}, y_{t-1}, \sigma_{t-1}^2, \text{rv}_{t-1}\right). \tag{4.7f}$$

4.3 Bayesian inference for the RealRECH model

4.3.1 The likelihood and prior

We adopt the Bayesian inference approach for the estimation of RealRECH. Following Hansen et al. (2012), we assume Gaussian errors $\epsilon_t \overset{i.i.d}{\sim} N(0,1)$ and $u_t \overset{i.i.d}{\sim} N(0, \sigma_u)$. Hence, given the observed data ($\boldsymbol{y} = \{y_1, ..., y_T\}, \boldsymbol{rv} = (\text{rv}_1, ..., \text{rv}_T)$) the log-likelihood function of the RealRECH model is:

$$\ell(\boldsymbol{y}, \boldsymbol{rv}|\theta) = -\frac{1}{2}\sum_{t=1}^{T}\left[\log(2\pi) + \log\left(\sigma_t^2\right) + y_t^2/\sigma_t^2\right] - \frac{1}{2}\sum_{t-1}^{T}\left[\log(2\pi) + \log\left(\sigma_u^2\right) + u_t^2/\sigma_u^2\right], \tag{4.8}$$

where $u_t = \text{rv}_t - \xi - \varphi\sigma_t^2 - \tau_1 y_t/\sigma_t - \tau_2\left(y_t^2/\sigma_t^2 - 1\right)$. Recall that the vector of model parameters θ consists of the GARCH and LSTM parameters. For example, the RealRECH(1,1) model with a single hidden state LSTM has 7 GARCH parameters, $(\beta, \gamma, \xi, \varphi, \tau_1, \tau_2, \sigma_u)$, and 26 LSTM parameters including β_0, β_1 and 24 parameters within the LSTM structure. For the prior distributions on the GARCH parameters, we use the commonly used priors in the literature; see, e.g., Gerlach and Wang (2016). The prior $N(0, 0.1)$ is used for the LSTM parameters.

4.3.2 Model estimation and prediction

For Bayesian inference and prediction in volatility modeling, the Sequential Monte Carlo (SMC) method (Chopin, 2002; Del Moral et al., 2012; Neal, 2001) is an effective approach for computing rolling-window volatility forecasts which can effectively sample from non-standard posteriors, while also providing the marginal likelihood estimate as a by-product. The SMC technique uses a set of M weighted particles initially sampled from an easy-to-sample distribution, such as the prior $p(\theta)$, with the particles then traversed through intermediate distributions which eventually become the target distribution. See Gunawan et al. (2022) for a review of the SMC method.

For in-sample model estimation and inference, we use the likelihood annealing version of SMC that samples from the sequence of distributions

$$\pi_k(\theta) \propto p(\theta) p\left(\boldsymbol{y}, \boldsymbol{rv} \mid \theta\right)^{\gamma_k}, \quad k = 0, 1, \ldots K; \tag{4.9}$$

here, $0 = \gamma_0 < \gamma_1 < \gamma_2 < \ldots < \gamma_K = 1$ are called the temperature levels. Reweighting, resampling, and a Markov transition are the three primary components of the SMC approach. Several methods exist to implement SMC in practice, and we briefly describe one of them now. The collection of weighted particles $\left\{W_{k-1}^j, \theta_{k-1}^j\right\}_{j=1}^M$ that approximate the intermediate distribution $\pi_{k-1}(\theta)$ is reweighted at the start of iteration k to approximate the target $\pi_k(\theta)$. The efficiency of these weighted particles is measured by the effective

sample size (ESS) (Kass et al., 1998)

$$\text{ESS} = \frac{1}{\sum_{j=1}^{M} \left(W_t^j\right)^2}, \tag{4.10}$$

where $W_t := (W_t^1, \ldots, W_t^M)$ is the weight vector for the M particles at time t. If the ESS is below a threshold, the particles are resampled to obtain equally weighted particles, and a Markov kernel with the invariant distribution $\pi_k(\theta)$ is then applied to refresh these equally weighted particles. Following Del Moral et al. (2012), we adaptively choose the tempering sequence γ_k in order to maintain an adequate particle diversity.

For out-of-sample rolling-window volatility forecasting which updates the posterior each time a new data observation arrives, we employ the data annealing SMC method that samples from the sequence

$$\pi_t(\theta) \propto p(\theta)p(\boldsymbol{y}_{1:T+t}, \boldsymbol{rv}_{1:T+t}|\theta), \quad t = 1, 2, ... \tag{4.11}$$

4.4 Empirical Analysis

This section studies the performance of the RealRECH(1,1) model and compares it to GARCH, RealGARCH and RECH using 31 stock indices including the Amsterdam Exchange Index (AEX), the Dow Jones Index (DJI), the Frankfurt Stock Exchange (GDAXI) and the Standard and Poor's 500 Index (SP500). The datasets were downloaded from the Realized Library of The Oxford-Man Institute. In the main text, we present results for the above mentioned four representative indices and the average over 31 indices to conserve space; the detailed results for all 31 indices are shown in the appendix. Given the closing prices $\{P_t, t = 0, ..., T\}$, we compute the demeaned close-to-close return process as

$$y_t = 100 \left(\log \frac{P_t}{P_{t-1}} - \frac{1}{T} \sum_{i=1}^{T} \log \frac{P_i}{P_{i-1}} \right), \quad t = 1, 2, \ldots, T. \tag{4.12}$$

We adopt the 5 mins realized variance (Andersen et al., 2003) as the realized volatility measure, rv_t, in RealGARCH and RealRECH. As the realized volatility measures ignore

the overnight variation of the prices and sometimes the variation in the first few minutes of the trading day when recorded prices may contain large errors (Shephard & Sheppard, 2010), we follow Hansen and Lunde (2005) and scale the realized volatility measure as

$$\widetilde{\sigma}_t^2 = \hat{c} \cdot \mathrm{rv}_t \text{ where } \hat{c} = \frac{\sum_{i=1}^{T} y_i^2}{\sum_{i=1}^{T} \mathrm{rv}_i}, \quad t = 1, 2, \ldots, \tag{4.13}$$

and use $\tilde{\sigma}_t^2$ as the estimate of the latent conditional variance σ_t^2. Our sample is from 1 January 2004 to 1 January 2022, including the COVID-19 pandemic period, which we divide it into a first half for training and the second half for out of sample analysis. There are, on average, 2139 trading days for both. See appendix for a detailed data description.

4.4.1 Parameter estimation and in-sample fit

As mentioned in Section 4.3, the marginal likelihood estimate is a by-product of SMC, which is useful for in-sample model comparison using a Bayes factor. Given the training data ($\boldsymbol{y} = y_{1:T}, \boldsymbol{rv} = \mathrm{rv}_{1:T}$), let $\widehat{p}(\boldsymbol{y}, \boldsymbol{rv}|M_i)$ be the marginal likelihood estimate of model M_i, $i = 1, 2$. The Bayes factor of model M_1 relative to M_2 is

$$\mathrm{BF}_{M_1,M2} = \frac{\widehat{p}(\boldsymbol{y}, \boldsymbol{rv}|M_1)}{\widehat{p}(\boldsymbol{y}, \boldsymbol{rv}|M_2)}. \tag{4.14}$$

The higher the Bayes factor, the more decisive is the evidence that model M_1 is superior to M_2 (Kass & Raftery, 1995). Table 4.1 summarizes the parameter estimation (for the five main parameters) and the Bayes factor (with GARCH used as the baseline model) for the four models, GARCH, RealGARCH, RECH and RealRECH. The last panel reports the results for the average over the 31 indices.

We draw the following conclusions from the estimation results. First, the marginal likelihood estimates show that the RealRECH model fits the index datasets better than the competing models for 28 of 31 indices. On average, the Bayes factors of the RealRECH model compared with the GARCH, RECH, and RealGARCH models are roughly e^{58}, e^{30} and e^{14}, respectively, which according to Jeffery's scale for interpreting the Bayes factor (Jeffreys, 1998), decisively support the RealRECH model. Second, the estimated

Table 4.1: In-sample analysis of model parameters

		α	β	β_0	β_1	γ	Mar.llik	BF
AEX	garch	0.105	0.884	-	-	-	−3450.8	1
		(0.012)	(0.013)				(0.183)	
	rech	0.038	0.741	0.049	0.907	-	−3400.6	e^{50}
		(0.014)	(0.061)	(0.019)	(0.182)		(0.280)	
	realgarch	-	0.549	-	-	0.371	−3376.3	e^{75}
			(0.028)			(0.028)	(0.018)	
	realrech	-	0.527	0.112	0.801	0.341	−3370.5	$\mathbf{e^{80}}$
			(0.021)	(0.022)	(0.081)	(0.015)	(0.220)	
DJI	garch	0.094	0.890	-	-	-	−3027.2	1
		(0.011)	(0.012)				(0.100)	
	rech	0.049	0.636	0.063	0.912	-	−2991.0	e^{36}
		(0.011)	(0.063)	(0.019)	(0.182)		(0.785)	
	realgarch	-	0.634	-	-	0.318	−2968.4	e^{59}
			(0.024)			(0.027)	(0.075)	
	realrech	-	0.840	1.413	11.827	0.005	−2962.9	$\mathbf{e^{64}}$
			(0.014)	(0.148)	(0.577)	(0.004)	(0.421)	
GDAXI	garch	0.098	0.889	-	-	-	−3615.5	1
		(0.013)	(0.014)				(0.107)	
	rech	0.049	0.666	0.079	1.000	-	−3570.0	e^{46}
		(0.013)	(0.071)	(0.036)	(0.206)		(0.349)	
	realgarch	-	0.439	-	-	0.452	−3534.8	e^{81}
			(0.025)			(0.031)	(0.055)	
	realrech	-	0.414	0.126	5.561	0.307	−3523.6	$\mathbf{e^{92}}$
			(0.029)	(0.084)	(0.409)	(0.022)	(0.141)	
SPX	garch	0.091	0.895	-	-	-	−3177.1	1
		(0.010)	(0.011)				(0.158)	
	rech	0.049	0.657	0.071	0.871	-	−3150.4	e^{27}
		(0.010)	(0.059)	(0.020)	(0.193)		(0.491)	
	realgarch	-	0.633	-	-	0.317	−3100.1	$\mathbf{e^{77}}$
			(0.022)			(0.025)	(0.078)	
	realrech	-	0.831	1.549	11.432	0.007	−3104.7	e^{72}
			(0.016)	(0.160)	(0.660)	(0.005)	(0.205)	
Mean	garch	0.102	0.882	-	-	-	−3411.9	1
		(0.013)	(0.015)				(0.132)	
	rech	0.059	0.729	0.071	0.850	-	−3384.3	e^{28}
		(0.014)	(0.057)	(0.032)	(0.193)		(0.226)	
	realgarch	-	0.628	-	-	0.306	−3368.4	e^{44}
			(0.024)			(0.025)	(0.214)	
	realrech	-	0.667	0.613	5.203	0.168	−3354.0	$\mathbf{e^{58}}$
			(0.034)	(0.118)	(0.453)	(0.021)	(0.472)	

Note: The last two columns show the natural logarithms of the estimated marginal likelihood (Mar.llik) with Monte Carlo standard errors (in parentheses) across 6 different runs of SMC and the Bayes factors (BF) that uses GARCH as the baseline.

posterior mean of the parameter β_1 in the RealRECH model is significantly different from zero in all cases, providing evidence of the volatility effects rather than linearity that the previous conditional variance σ_{t-1}^2 and realized volatility measure rv_{t-1} have on σ_t^2. This also suggests that the LSTM component $g(\omega_t)$ in RealRECH can detect these effects effectively. Additionally, the estimated value of the parameter γ (the coefficient concerning the realized volatility measure) in RealRECH is always smaller than that in RealGARCH. This is perhaps because the effect of the realized volatility measure rv_{t-1} on the underlying volatility σ_t^2 is well captured by the LSTM component, which, unlike RealGARCH, allows non-linear dependence of σ_t^2 on rv_{t-1}.

4.4.2 Volatility forecast error compared to ex-post realized volatility measures

We now test the predictive performance of the RealRECH model for volatility forecasting. The forecast performance is measured by mean squared error (MSE) and mean absolute deviation (MAD) computed on test data D_{test} of size T_{test}

$$\text{MSE} = T_{test}^{-1} \sum_{D_{test}} (\widehat{\sigma}_t - \widetilde{\sigma}_t)^2 \tag{4.15}$$

$$\text{MAE} = T_{test}^{-1} \sum_{D_{test}} |\widehat{\sigma}_t - \widetilde{\sigma}_t| \tag{4.16}$$

where $\widehat{\sigma}_t$ is the one-step-ahead rolling-window forecast of the latent σ_t and $\widetilde{\sigma}_t$ is the squared root of an ex-post realized volatility measure after rescaling as in (4.13). We use five ex-post volatility proxies, the Realized Variance (RV), Bipower Variation (BV), Median Realized Volatility (MedRV), Realized Kernel Variance with the Non-flat Parzen kernel (RK-Parzen) and the Tukey-Hanning kernel (RSV). See Shephard and Sheppard (2010) for details about the Realized Library.

Tables 4.2 and 4.3 summarize the forecast performance of the four models. The last column in each table shows the number of times a model has the best predictive score across the five volatility proxies. The last panel in each table shows the average scores over the 31 indices. Using the five ex-post realized volatility measures as the ground

truth, the results show that the RealRECH model performs the best in terms of volatility forecasting.

Table 4.2: Forecast performance across realized volatility measures

		RV5	BV	MedRV	RK-Parzen	RSV	Count
AEX	garch	0.151	0.142	0.141	0.209	0.190	0.0
	rech	0.143	0.131	0.126	0.188	0.180	0.0
	realgarch	0.121	0.109	0.102	0.173	0.158	0.0
	realrech	**0.116**	**0.103**	**0.094**	**0.157**	**0.152**	**5.0**
DJI	garch	0.203	0.174	0.177	0.202	0.274	0.0
	rech	0.194	0.176	0.176	0.193	0.266	0.0
	realgarch	0.121	0.124	0.127	0.119	0.196	0.0
	realrech	**0.112**	**0.122**	**0.118**	**0.109**	**0.184**	**5.0**
GDAXI	garch	0.180	0.174	0.182	0.203	0.221	0.0
	rech	0.159	0.154	0.160	0.180	0.200	0.0
	realgarch	0.111	0.110	0.114	0.140	0.152	0.0
	realrech	**0.106**	**0.104**	**0.105**	**0.132**	**0.146**	**5.0**
SPX	garch	0.183	0.166	0.180	0.182	0.243	0.0
	rech	0.174	0.163	0.172	0.172	0.237	0.0
	realgarch	0.117	**0.119**	0.129	0.119	0.182	1.0
	realrech	**0.112**	0.121	**0.125**	**0.113**	**0.176**	**4.0**
Mean	garch	0.196	0.174	0.184	0.243	0.267	0.0
	rech	0.184	0.163	0.172	0.227	0.257	0.3
	realgarch	0.153	0.138	0.151	0.204	0.225	1.2
	realrech	**0.147**	**0.133**	**0.146**	**0.192**	**0.219**	**3.5**

Note: The last column reports the number of times a model has the lowest MSE among the five realized volatility measures. The bottom panel reports the average across the 31 indices. The bold numbers indicate the best scores.

4.4.3 Fitness to return series and tail risk forecast

An attractive feature of GARCH-type models including RealRECH is that they model both the volatility and return processes. This feature is crucial to risk management which is one of the most important applications of volatility modeling. For risk management, the key task is to forecast the Value at Risk (VaR) and Expected Shortfall (ES). An α-level VaR is the α-level quantile of the distribution of the return, and the α-level ES is the conditional expectation of return values that exceed the corresponding α-level VaR. Both VaR and ES are used as the two key risk measures in financial regulation and recommended by the Basel Accord. To evaluate the quality of the VaR forecast, we

Table 4.3: Forecast performance: MAD for different realized volatility measures.

		RV5	BV	MedRV	RK-Parzen	RSV	Count
AEX	garch	0.259	0.256	0.266	0.332	0.298	0.0
	rech	0.237	0.233	0.242	0.300	0.276	0.0
	realgarch	0.218	0.211	0.219	0.294	0.260	0.0
	realrech	**0.209**	**0.202**	**0.209**	**0.274**	**0.249**	**5.0**
DJI	garch	0.291	0.273	0.301	0.308	0.344	0.0
	rech	0.267	0.253	0.278	0.281	0.323	0.0
	realgarch	0.218	0.214	0.239	0.237	0.279	0.0
	realrech	**0.208**	**0.204**	**0.226**	**0.223**	**0.272**	**5.0**
GDAXI	garch	0.317	0.310	0.320	0.343	0.356	0.0
	rech	0.297	0.290	0.297	0.324	0.337	0.0
	realgarch	0.237	0.233	0.242	0.274	0.284	0.0
	realrech	**0.233**	**0.229**	**0.237**	**0.266**	**0.277**	**5.0**
SPX	garch	0.295	0.286	0.312	0.305	0.340	0.0
	rech	0.265	0.262	0.285	0.275	0.315	0.0
	realgarch	0.222	0.225	0.247	0.240	0.277	0.0
	realrech	**0.211**	**0.217**	**0.237**	**0.226**	**0.270**	**5.0**
Mean	garch	0.289	0.278	0.293	0.338	0.337	0.0
	rech	0.278	0.267	0.282	0.323	0.327	0.2
	realgarch	0.248	0.238	0.258	0.305	0.302	0.7
	realrech	**0.241**	**0.233**	**0.254**	**0.292**	**0.294**	**4.1**

Note: The last column reports the number of times a model has the lowest MAD among the five realized volatility measures. The last panel reports the average across th 31 indices. The bold numbers indicate the best scores.

adopt the standard quantile loss function

$$\text{Qloss} := \sum_{y_t \in D_{test}} \left(\alpha - I\left(y_t < Q_t^\alpha\right)\right)\left(y_t - Q_t^\alpha\right), \tag{4.17}$$

where Q_t^α is the forecast of α-level VaR of y_t (Koenker & Bassett, 1978). We note that, given the volatility σ_t and the normality assumption of the random shock ϵ_t in (4.6a), it is straightforward to compute Q_t^α. The quantile loss function is strictly consistent (Fissler & Ziegel, 2016), i.e., the expected loss is lowest at the true quantile series. The most accurate VaR forecasting model should therefore minimize the quantile loss function. There is no strictly consistent loss function for ES; however, Fissler and Ziegel (2016) found that ES and VaR are jointly elicitable, i.e., there is a class of strictly consistent loss functions for evaluating VaR and ES forecasts jointly. Taylor (2019) showed that the negative logarithm of the likelihood function built from the Asymmetric Laplace (AL) distribution is strictly consistent for VaR and ES considered jointly and fits into the class developed by Fissler and Ziegel (2016). This AL based joint loss function is given as

$$\text{JointLoss} := \frac{1}{T_{\text{test}}} \sum_{D_{\text{test}}} \left(-\log\left(\frac{\alpha - 1}{\text{ES}_t^\alpha}\right) - \frac{\left(y_t - Q_t^\alpha\right)\left(\alpha - I\left(y_t \leq Q_t^\alpha\right)\right)}{\alpha \text{ES}_t^\alpha}\right) \tag{4.18}$$

with ES_t^α the forecast of α-level ES of y_t. To measure the fit of a volatility model to the return series, we also consider the Partial Predictive Score (PPS), which is one of the most commonly used metrics for evaluating predictive performance in statistical modeling. It measures the negative log-likelihood of observing the return series based on our volatility forecast. The model with the smallest PPS is preferred. The PPS score is defined as

$$\text{PPS} := -\frac{1}{T_{\text{test}}} \sum_{D_{\text{test}}} p\left(y_t \mid y_{1:t-1}\right). \tag{4.19}$$

Table 4.4 reports the quantile loss, joint loss for 1% and 5% VaR and ES forecast, and PPS score. The Count panel reports the number of indices where a model achieves the best predictive score. The results show that for most of the indices, the RealRECH model produces the best VaR and ES forecasts. RealRECH is also superior to its competitors

in terms of overall fit to the return series. It reports the lowest PPS on most returns series and on average. Additionally, we find that for most of the metrics, the RECH model performs the best on more indices than RealGARCH. In contrast, in the previous section, RealGARCH dominates RECH regarding forecast error compared ex-post proxies. This suggests that ranking conditional volatility forecasts by ex-post proxies can sometimes lead to undesirable outcomes, and we should also evaluate predictive performance by economic loss.

Table 4.4: Forecast performance: Tail risk forecast and Partial Predictive Score.

		Qloss_1%	JointLoss_1%	Qloss_5%	JointLoss_5%	PPS
AEX	garch	92.549	2.462	282.507	1.909	1.342
	rech	88.654	2.404	270.841	1.851	1.307
	realgarch	86.838	2.317	273.663	1.847	1.298
	realrech	**84.982**	**2.279**	**267.902**	**1.815**	**1.288**
DJI	garch	**80.691**	2.340	250.738	1.767	1.175
	rech	81.502	2.243	247.145	1.706	1.135
	realgarch	82.620	2.299	242.673	1.712	1.142
	realrech	81.193	**2.236**	**242.049**	**1.689**	**1.134**
GDAXI	garch	100.503	2.514	316.005	2.034	1.484
	rech	**97.312**	**2.480**	**305.339**	**1.991**	**1.453**
	realgarch	103.330	2.577	314.835	2.039	1.460
	realrech	100.372	2.557	309.042	2.016	1.453
SPX	garch	83.164	2.424	253.502	1.803	1.192
	rech	81.716	2.327	247.041	1.735	1.149
	realgarch	**80.571**	2.311	244.235	1.728	1.136
	realrech	81.334	**2.287**	**243.364**	**1.714**	**1.135**
Mean	garch	78.545	2.262	260.099	1.859	1.355
	rech	76.549	2.224	253.029	1.825	1.339
	realgarch	79.518	2.266	258.329	1.853	1.348
	realrech	**76.141**	**2.223**	**248.735**	**1.812**	**1.336**
Count	garch	1	3	0	0	0
	rech	9	7	6	7	11
	realgarch	4	3	3	1	4
	realrech	**17**	**18**	**22**	**23**	**16**

Note: Qloss_1%, JointLoss_1%, Qloss_5%, JointLoss_5% are the quantile loss and jointloss at 1% and 5% respectively. The last two panels report the average scores and the number of times a model has the best predictive scores across the 31 indices.

4.4.4 Simulated option trading

Apart from risk management, options trading is also one of the most attractive applications of volatility forecasting. In a theoretical pricing model, volatility is the most difficult input for traders to predict among all the inputs required for option evaluation. At the same time, volatility often plays the most crucial role in actual trading decisions. Consider the inputs of a Black-Scholes model for European options:

1. The current price of the underlying security
2. The option's exercise price
3. The expiration time
4. The risk-free interest rate of the life of the option
5. The volatility of the underlying contract.

Volatility is the only unknown input here; hence the profitability of an options trader is greatly affected by their ability to forecast volatility. This section examines model performance based on its ability to price options correctly. We follow R. F. Engle et al. (1990) to design a hypothetical option market where each agent uses their volatility forecast and the Black-Scholes model to price options and trade with competing agents. The experiment is organized as follows:

1. An agent in the experiment trades on the options of \$1 share of the S&P500 index with an at-the-money exercise price (1\$) and 1-day expiration. The risk-free interest rate is set to zero.
2. Each agent M determines their call options price

$$P_{t,M} = 2\Phi\left(\frac{1}{2}\sigma_{t,M}\right) - 1 \tag{4.20}$$

given the volatility forecast $\sigma^2_{t,M}$ and the Black-Scholes formula, with Φ the standard normal cumulative distribution function.

3. The pair-wise trading then takes place between agents M_1 and M_2 at their predicted mid-price P_t,

$$P_t = (P_{t,M_1} + P_{t,M_2})/2. \tag{4.21}$$

Each agent either buys or sells a straddle (a combination of put and call options) and uses its variance forecast to determine the hedge ratio, δ. In our case, $\delta_{\text{straddle}} = 1 - 2\Phi\left(\frac{1}{2}\sigma_t\right)$. The intuition is that the agent with the higher volatility forecast will believe the straddle is underpriced from P_t, thus buying the straddle from its counterpart and vice versa.

4. For each pair-wise trade, the daily profit of buying a straddle is then calculated as

$$|r_t| - 2P_t + r_t\left(1 - 2\Phi\left(\frac{1}{2}\sigma_{t,M}\right)\right), \tag{4.22}$$

and the daily profit of selling a straddle is

$$2P_t - |r_t| - r_t\left(1 - 2\Phi\left(\frac{1}{2}\sigma_{t,M}\right)\right). \tag{4.23}$$

With a total of k agents, each agent conducts $k-1$ trades per day. The daily sum of the trading profit is then divided by $k-1$ and averaged throughout the testing period.

Table 4.5 reports the daily, annual profit (in cents) and Sharpe ratio of the options agents that use GARCH, RealGARCH, RECH and RealRECH as their forecast models. The simulations are repeated on 31 stocks indices and the following four scenarios:

1. All agents trade in the market.

2. Only RealGARCH, RECH, and RealRECH agents trade in the market.

3. Only RealGARCH and RealRECH agents trade against each other.

4. Only RECH and RealRECH agents trade against each other.

In scenario (1), where all agents are trading against each other, the RealRECH agent generates the highest profits and Sharpe ratio in most stock indices on average. Scenarios (2), (3) to (4) further illustrate the consistent profitability of the RealRECH agent against its two direct ancestors, RealGARCH and RECH, in our hypothetical market. A surprising finding is that the RealGARCH performs almost as badly as the GARCH model.

Table 4.5: Forecast performance: Annualized return (Ret.) and Sharpe ratio (Sharpe) of option trading simulation

		Scenario1		Scenario2		Scenario3		Scenario4	
		Ret.	Sharpe	Ret.	Sharpe	Ret.	Sharpe	Ret.	Sharpe
AEX	garch	−22.4	−3.2	-	-	-	-	-	-
	rech	5.9	0.5	−4.1	−0.6	−9.5	−1.1	-	-
	realgarch	1.9	0.0	−7.4	−1.0	-	-	−12.4	−1.4
	realrech	**17.8**	**1.9**	**11.3**	**1.2**	**9.2**	**0.7**	**12.9**	**1.0**
DJI	garch	−12.7	−1.7	-	-	-	-	-	-
	rech	3.7	0.3	−1.5	−0.4	−3.6	−0.5	-	-
	realgarch	3.3	0.2	−1.2	−0.3	-	-	−1.5	−0.3
	realrech	**5.5**	**0.4**	**1.6**	**0.0**	**2.6**	**0.1**	**0.3**	**−0.1**
GDAXI	garch	−19.3	−2.3	-	-	-	-	-	-
	rech	9.6	0.9	−1.0	−0.2	−4.7	−0.5	-	-
	realgarch	−0.5	−0.2	−7.0	−0.8	-	-	−10.6	−1.0
	realrech	**11.7**	**1.0**	**7.0**	**0.5**	**3.3**	**0.2**	**10.0**	**0.7**
SPX	garch	−17.4	−2.4	-	-	-	-	-	-
	rech	1.7	−0.0	−6.5	−1.0	−6.1	−0.7	-	-
	realgarch	**8.5**	**0.9**	**3.4**	**0.2**	-	-	**−0.0**	**−0.1**
	realrech	8.4	0.8	2.3	0.1	**5.3**	**0.4**	−1.1	−0.2
Mean	garch	−6.7	−1.0	-	-	-	-	-	-
	rech	5.2	0.4	2.0	−0.0	−2.8	−0.4	-	-
	realgarch	−6.2	−1.1	−10.0	−1.4	-	-	−13.2	−1.6
	realrech	**10.8**	**0.8**	**11.1**	**0.9**	**4.1**	**0.1**	**19.1**	**1.2**
Count	garch	1	1	0	0	0	0	0	0
	rech	8	9	9	9	9	9	0	0
	realgarch	1	1	1	1	0	0	4	4
	realrech	**21**	**21**	**21**	**21**	**22**	**22**	**27**	**27**

Note: Annualized returns are reported in cent(%). The marks "-" indicate models that are not trading. The last two panels report the average Ret./Sharpe and the number of times a model has the best predictive scores across 31 indices respectively.

4.4.5 Statistical significance

The previous sections show that RealRECH outperforms the competing models in terms of in-sample fit, forecasting error, tail risk forecast and option pricing. This section tests whether these improvements are statistically significant using the Model Confidence Set (MCS) introduced by Hansen et al. (2011). Let $\mathcal{M}$ be a set of competing models. A set of superior models (SSM) is established under the MCS procedure, which consists of a series of equal predictive accuracy tests given a specific confidence level. Let $L_{i,t}$ be a performance loss, such as the MSE or quantile loss, incurred by model $i \in \mathcal{M}$ at time t. Define $d_{i,j,t} = L_{i,t} - L_{j,t}$ to be the relative loss of model i compared to model j at time t. The MCS test assumes that $d_{i,j,t}$ is a stationary time series for all i, j in $\mathcal{M}$, i.e., $\mu_{i,j} = \mathbb{E}(d_{i,j,t})$ for all t. By testing the equality of the expected loss difference $\mu_{i,j}$, MCS determines if all models have the same level of predictive accuracy. The null hypothesis is

$$H_0 : \mu_{i,j} = 0, \quad \text{for all } i, j \in \mathcal{M}. \tag{4.24}$$

A model is eliminated when the null hypothesis H_0 of equal forecasting ability is rejected. The collection of models that do not reject the null hypothesis H_0 is then defined as the SSM. For each model $i \in \mathcal{M}$, the MCS produces a p-value p_i. The lower the p-value of a model, the less likely that it will be included in the SSM. See Hansen et al. (2011) for more details.

Table 4.6 reports the model confidence sets computed by all the predictive scores. For each model, we report the total number of times, across the 31 indices, that the model is included in the MCS and its average p-value (in parentheses). We note that a small p-value indicates that the model is unlikely to be the best model (Hansen et al., 2011). At a 75% confidence level and across the predictive scores, the RealRECH models are included in SSM for 24 indices, and have the highest p-value of 1 for 20 indices. The result shows that RealRECH is the most likely to be included in the SSM and have statistically significant superiority over the other models in all the considered predictive metrics: forecasting error (MSE and MAD), fit to return series (PPS), risk forecast (quantile loss

and joint loss) and option pricing.

Table 4.6: Statistical significance of model performance across indices

This table reports the number of times each model is included in the set of superior models (SSM) and the average p-value, shown in parentheses, across the 31 indices. Higher p-values indicate a greater likelihood of inclusion in the SSM.

	GARCH	RECH	RealGARCH	RealRECH
MSE	1	5	18	24
	(0.01)	(0.10)	(0.48)	(0.67)
MAD	1	4	11	26
	(0.02)	(0.09)	(0.27)	(0.84)
Qloss_1%	9	20	12	22
	(0.16)	(0.51)	(0.25)	(0.65)
Qloss_5%	1	12	6	25
	(0.01)	(0.30)	(0.15)	(0.80)
JointLoss_1%	12	16	14	24
	(0.25)	(0.40)	(0.30)	(0.70)
JointLoss_5%	2	13	6	25
	(0.03)	(0.31)	(0.12)	(0.80)
PPS	7	16	10	22
	(0.13)	(0.43)	(0.24)	(0.63)
OptTrading	5	13	6	22
	(0.10)	(0.39)	(0.14)	(0.70)

4.5 Conclusions

This chapter developed a new and enhanced variant of the RECH model that has two novel features. First, we add realized volatility measures into RECH, which is linked directly to the conditional volatility. Second, the RealRECH model employs LSTM, a more robust RNN architecture than the basic RNN in RECH. Sequential Monte Carlo, with likelihood annealing and data annealing, is employed for both in-sample and out-of-sample Bayesian inference and forecasting. We examined the RealRECH model on 31 of the world's major stock markets, and demonstrated that the model offers a substantial improvement in terms of both statistical criteria (in-sample fit, forecast error to realised measures) and economic criteria (tail risk forecast and option pricing) relative to the standard GARCH, RECH and RealGARCH models.

This chapter presented a novel methodology for volatility modeling, effectively harnessing the strengths of deep learning and the invaluable insights from realized volatility data. Bayesian inference and forecasting are performed using Sequential Monte Carlo with likelihood and data annealing. The RealRECH model is evaluated on 31 major world stock markets, demonstrating improved performance in statistical criteria (in-sample fit, forecast error to realized measures) and economic criteria (tail risk forecast and option pricing) compared to the standard GARCH, RECH, and RealGARCH models. Our proposed framework could be extended in several ways. First, it is possible to include multiple realized volatility measures since Hansen and Huang (2016) discovered that including multiple realized volatility measures evidently improves both the in-sample and out-of-sample fit. The multiple measurement equations then could be replaced by a single LSTM with multiple outputs which would allow us to capture the non-linear relationship between conditional volatility and realized volatility measures as well as the interactions between the realized volatility measures. Second, incorporating financial news into the input of RealRECH is an interesting topic to study, as news has been shown to have a major influence on volatility movement (Atkins et al., 2018; Rahimikia et al., 2021; Xing et al., 2019). Lastly, incorporating transfer learning would be a natural extension. Transfer learning can help mitigate the problem of insufficient training data and enable us to use a large number of pre-trained deep learning models and traditional financial econometric models for volatility modeling.

Chapter 5

Conclusion

5.1 Summary of Principal Findings

This dissertation examines the role of deep learning in financial forecasting and argues that mixed results in the literature largely reflect a methodological mismatch: data-intensive models are often assessed in data-scarce environments. We advocate a shift from a model-centric mindset marked by local estimation and an overemphasis on architectural choices to a data-centric paradigm. Three empirical studies provide evidence on the conditions under which reliable DL performance gains arise, the mechanisms that drive them, and practical guidance for deploying deep learning models in an era of large-scale financial data.

First, this thesis establishes that data scale and diversity are the primary drivers of success for deep learning models in finance. Chapter 2 provides decisive evidence for this data-scaling effect. When trained locally on individual time series, neural networks consistently fail to outperform simpler, well-specified benchmarks like GARCH. However, when trained globally on a large and diverse cross-section of over 11,000 stocks, these same neural network architectures exhibit a dramatic and monotonic improvement in predictive accuracy as the data size increases. A globally trained "universal" model not only dominates its locally trained counterparts but also delivers robust zero-shot forecasts for assets entirely unseen during training. This finding reconciles many of the

conflicting results in the existing literature, attributing the reported underperformance of neural networks to the constraints of a data-scarce, local training paradigm rather than a fundamental flaw in the models themselves.

Second, this thesis demonstrates that deep learning models, when appropriately applied, are not merely "black boxes" but can serve as powerful instruments for economic discovery. The investigation into high-frequency market microstructure in Chapter 3 provides the strongest evidence for this claim. While established, theory-driven variables were found to effectively summarize the information in a contemporaneous snapshot of the limit order book, they failed to capture the rich, predictive information contained in the history of order flow. A Transformer model, trained directly on a sequence of raw order book data, improved predictive accuracy by over 60

Third, the research demonstrates that synergistic integration of econometric and deep learning models yields a superior hybrid framework. Chapter 4 introduced the Realized Recurrent Conditional Heteroskedasticity model, which embeds an LSTM network within the established RealGARCH framework. This hybrid model consistently outperformed both its pure econometric and pure deep learning parents across a wide range of statistical and economic metrics. The GARCH component provides a robust structural foundation grounded in financial theory, effectively regularizing the model, while the LSTM component flexibly captures complex nonlinear dynamics and long-range dependencies that the rigid structure of the econometric model misses. This result shows that the two approaches need not be mutually exclusive; a thoughtful synthesis can harness the interpretability and structural priors of econometrics with the expressive power of deep learning.

In summary, this dissertation's central message is that successful applications of deep learning in finance require a paradigm shift. Rather than treating each time series as a separate estimation task and overemphasizing architectural choices, we should adopt a data-centric perspective that focuses on increasing the quantity and quality of data fed to the model. Whether through carefully designed hybrids, globally trained models, or sequence-aware architectures applied directly to granular data streams, the key to

realizing deep learning's potential lies in providing models with the scale and diversity of data they require.

5.2 Implications and Contributions

The findings of this thesis carry implications for both academic research and industry practice.

5.2.1 Implications for Theoretical Research

The primary contribution of this work to financial theory is the strong empirical evidence against the Markovian assumption in price discovery. The results from Chapter 3 suggest that the evolution of the limit order book contains predictive information that persists over time, a dynamic not captured by models that rely solely on the current market state. This finding calls for a renewed focus on developing and testing non-Markovian theories of market microstructure that explicitly incorporate path dependency and the memory of order flow, such as those proposed by Riccò et al. (2023).

Furthermore, this thesis provides a template for using deep learning as a tool for economic inquiry. By comparing models trained on raw data versus engineered features, researchers can systematically test the sufficiency of existing theoretical constructs. Our finding that established variables capture contemporaneous information well but miss historical dynamics suggests that while past theoretical work has been successful, the next frontier lies in modeling temporal dependencies. The ability of deep learning models to learn stylized facts like the leverage effect and identify key predictive variables from raw data reinforces their validity as a non-parametric tool for validating and guiding theory.

Finally, the success of global models challenges the default econometric practice of treating every financial time series as a unique, heterogeneous entity. While asset-specific idiosyncrasies certainly exist, our results suggest that the common, underlying patterns across assets are far more significant than previously assumed, at least from a forecasting

perspective. This encourages a re-evaluation of the trade-off between model specificity and the statistical power gained from pooling data.

5.2.2 Implications for Financial Practice

For practitioners, the implications are direct and actionable. The demonstrated superiority of globally trained models offers a more accurate, robust, and computationally efficient alternative to the industry practice of maintaining separate models for thousands of individual assets. A single "universal" volatility model, as developed in Chapter 2, can be trained once on a broad market cross-section and then deployed to generate high-quality forecasts for any asset, including newly listed securities with little to no historical data. This zero-shot forecasting capability is a significant practical advantage.

The findings also underscore that for financial institutions, data is a primary strategic asset. The competitive edge in the data-centric era will come not just from superior model architectures but from the scale, diversity, and granularity of the datasets used for training. This advocates for a strategy focused on aggregating diverse data sources—across asset classes, regions, and data types—to build more powerful and generalizable models.

Lastly, the models developed in this thesis offer tangible improvements for risk management. The RealRECH model from Chapter 3 and the universal model from Chapter 2 both demonstrated superior performance in forecasting tail-risk measures like Value-at-Risk and Expected Shortfall. Their enhanced robustness to outliers and rapid adaptation to market shocks, as shown in the empirical analyses, make them more reliable tools for navigating volatile market conditions.

5.3 Limitations and Future Research

While this thesis provides a comprehensive investigation, it is subject to limitations that naturally open avenues for future research. The empirical focus of this work has been on volatility forecasting and short-term price discovery in equity markets. The data-centric principles established here are likely generalizable, but their specific application

to other domains, such as long-horizon return predictability, credit risk, or other asset classes like derivatives and fixed income, requires further investigation. These areas present different data structures and signal-to-noise ratios, which may alter the specific conclusions regarding model choice and data requirements.

Furthermore, while this thesis has pushed the boundaries of model interpretation using techniques like saliency maps and news impact curves, the "black box" problem is not fully solved. The economic mechanisms learned by deep neural networks are still inferred rather than explicitly defined. Future research could benefit from applying emerging techniques in causal machine learning and advanced interpretability to move from identifying what a model has learned to understanding why it has learned it, thereby extracting more direct and testable economic hypotheses.

Finally, this work points toward several exciting directions for future inquiry.

1. **Developing Truly Universal Cross-Asset Models:** The next logical step is to extend the global training paradigm beyond equities to encompass a wide array of financial instruments, including bonds, commodities, currencies, and derivatives. A universal model trained on such a diverse dataset could uncover fundamental inter-market linkages and lead to more holistic and robust models of the entire financial ecosystem.

2. **Integrating Unstructured and Alternative Data:** The flexible architecture of deep learning is uniquely suited to integrating diverse data types. Future research should focus on incorporating unstructured data, such as financial news text, social media sentiment, and satellite imagery, directly into the global forecasting frameworks developed in this thesis. This would represent a significant step towards models that learn from the full information landscape available to market participants.

3. **Exploring the Frontiers of Price Discovery Theory:** The evidence for non-Markovian dynamics in Chapter 4 is a call to action for theorists. Future work should focus on building and empirically testing a new generation of microstructure

models that explicitly account for the persistent memory of order flow and the path-dependent nature of price formation.

In conclusion, this thesis has argued that the future of financial forecasting is data-centric. By moving beyond the limitations of traditional, localized estimation and embracing the scale of modern financial data, deep learning can be transformed from a controversial tool into an indispensable instrument for prediction, risk management, and economic discovery.

Appendix A

Appendices for Chapter 2

This appendix provides supplementary material for Chapter 2, including additional results, training configurations.

A.1 Gated Recurrent Unit

In additional to LSTM, the GRU model of Chung et al. (2014) has proven efficiency in many machine learning applications. GRU uses reset and update gates, denoted r_t and z_t respectively, to regulate the information flow. It is written as follows

$$r_t = \psi\left(W_{ry} y_t + W_{rh} h_{t-1} + b_r\right) \tag{A.1a}$$

$$z_t = \psi\left(W_{zy} y_t + W_{zh} h_{t-1} + b_z\right) \tag{A.1b}$$

$$\tilde{h}_t = \tanh\left(W_{hy} y_t + W_{hh}\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{A.1c}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{A.1d}$$

$$\sigma_{t+1} = \mathrm{ReLU}\left(W_{\sigma h} h_t + b_\sigma\right) + 1e^{-8}. \tag{A.1e}$$

where $\tanh(\cdot)$ denotes the hyperbolic tangent activation function, $\odot$ denotes the element-wise product. The matrices W and vectors b are the model parameters.

A.2 Limit of Data Scaling Effect

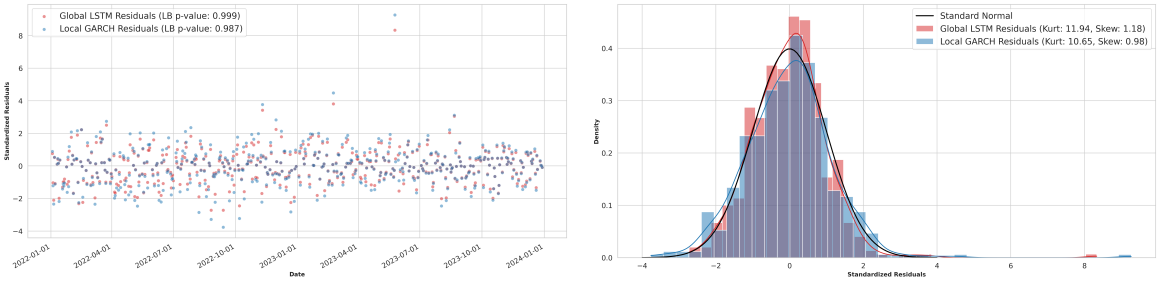
This section discuss the limit of data scaling effect. As shown in the main chapter, in our current experiment setting the performance of the global models plateaus when more than 1,280 stock series are pooled. This indicates that beyond a certain threshold, adding additional data from similar sources no longer contributes to performance improvements. This finding aligns with recent research on scaling laws in other areas of neural networks (Fernandez et al., 2024; Sorscher et al., 2022). To achieve further gains in model accuracy, it becomes necessary to shift focus from data size to data diversity. In other words, incorporating data from varied and distinct sources offers greater potential for improvement than merely increasing the volume of homogeneous data. Incorporating diverse financial instruments, such as derivatives, bonds, and foreign exchange rates, may enable the development of a more robust universal model for financial time series.

A.3 Residual Analysis

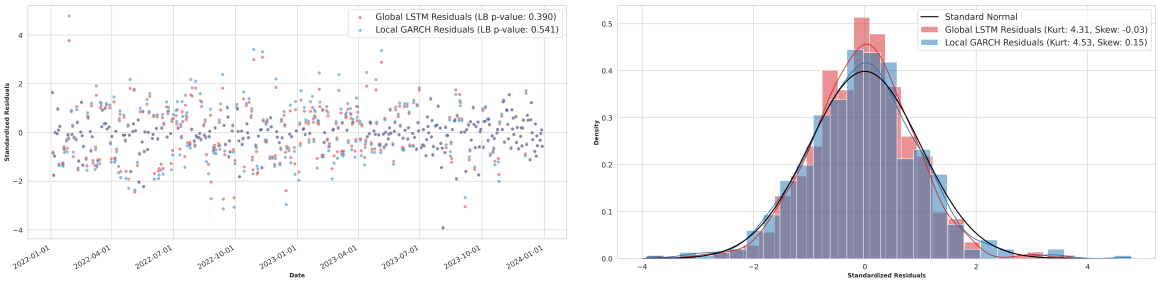
This section presents the residual analysis of the universal volatility model. Figure A.1 plots the standardized residuals, including the results of a Ljung-Box (LB) test for autocorrelation in the squared standardized residuals at lag 10, along with their distributions for the top three market-cap companies. The residual analysis highlights the model's ability to capture key stylized facts in financial time series, particularly the heterogeneity observed in volatility patterns. Similar to GARCH-family models, the universal model effectively accounts for varying degrees of persistence and clustering in volatility across different time frames, reflecting the intricate dynamics of financial markets. The standardized residuals exhibit characteristics consistent with a well-specified model, including uncorrelated but non-normally distributed series.

Figure A.1: Standardized residuals and distributions for the top three companies in the test period

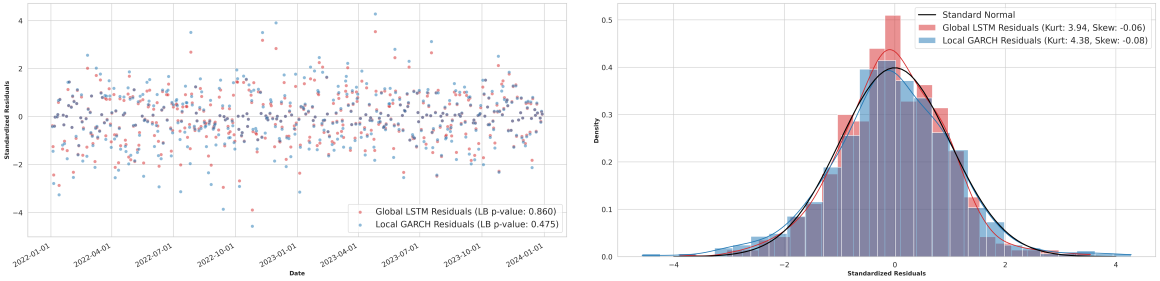
This figure shows the standardized residuals and their distributions for the test periods of the top three companies. Labels on the standardized residual plots display the results of the Ljung–Box (LB) test at lag 10, while labels on the distribution plots indicate the skewness and kurtosis values.



(a) Standardized residuals and distribution for Nvidia (NVDA)



(b) Standardized residuals and distribution for Apple (AAPL)



(c) Standardized residuals and distribution for Microsoft (MSFT)

A.4 Training Configurations

To ensure a consistent and comparable evaluation across the different NNs, we adopted standardized training configurations. This approach aligns with practices in those investigating scaling laws in natural language processing and computer vision (Kaplan et al., 2020; Zhai et al., 2022), where hyperparameter settings are kept uniform to isolate the effects of interest since our primary focus is on the impact of global training and data characteristics, rather than exhaustive hyperparameter optimization for specific NNs.

Model Size Table A.1 details the architectural hyperparameters for the four NN architecture employed. A key aspect of our configuration strategy was to ensure that all models possessed a comparable level of complexity, with approximately 500 parameters. This specific parameter count was chosen not through extensive tuning, but to be sufficiently large to avoid model capacity becoming a performance bottleneck. Our empirical work suggests that for low-frequency financial time series forecasting tasks investigated here, the choice of specific hyperparameters often plays a secondary role to the scale and diversity of the training data.

Table A.1: Architectural configurations for the neural network models

This table reports the architectural configurations for the neural network models. Hyperparameters were selected to yield approximately 500 parameters for each model type, ensuring comparable complexity. n_{layers} denotes the number of recurrent or Transformer layers; d_{hidden} is the number of units in recurrent layers; n_{head} is the number of attention heads in Transformers; d_{model} is the embedding dimension for Transformers; and d_{ff} is the dimension of the feed-forward layer within Transformers.

Model	n_{layers}	d_{hidden}	n_{head}	d_{model}	d_{ff}	Dropout
RNN	1	21	–	–	–	0.2
GRU	1	12	–	–	–	0.2
LSTM	1	10	–	–	–	0.2
Transformer	1	–	2	8	8	0.2

Optimization We optimize the in-sample NLL of all models using the standard Adam optimizer (Kingma & Ba, 2017). To achieve fast convergence and approach a near-minimum function value, we implement a cosine adaptive learning rate schedule. The learning rate begins at 1×10^{-2} and gradually decreases to a minimum of 1×10^{-4} over the course of the training. All models are trained for 10000 epochs to ensure sufficient optimization. To prevent overfitting, we apply early stopping, which terminates training if the validation loss does not improve over a specified number of epochs. The patience parameter is set to 1000 epochs, allowing training to continue for up to 1000 additional steps after the last improvement in validation loss. If no improvement is observed during this period, training stops, and the model parameters revert to those corresponding to the lowest validation loss recorded. All NN training is performed on a consumer-level

Nvidia 4090 GPU with an Intel CPU 13900K, using the PyTorch library.

Mini-batch training and Batch size To train global models, we use standard mini-batch training to optimize the objective function. At each iteration, a mini-batch $\mathcal{M}$ of m stock series is randomly selected to provide an unbiased estimate of the objective function. The training process learns the model parameters θ^* using the stochastic gradient $\nabla\widehat{\ell}(\mathbf{Y}|\theta^*)$. For example, with a batch size of $m = 10$, each optimization step involves 10 randomly selected stock series. To ensure the same number of optimization steps across models trained with different data sizes, we set the batch size to Number of Training Stocks/5. For instance, a model trained with 10 stocks uses a batch size of 2, while a model trained with 10,240 stocks uses a batch size of 2048.

Handling variable-length stock series To leverage modern parallel computing methods, each mini-batch of series must have the same length. This poses a challenge as stock series often vary significantly in length, especially across different exchanges. To address this, we use padding and masking techniques. Padding extends all series within a mini-batch to match the length of the longest series by appending placeholder values (e.g., zeros) to shorter series. Masking is then applied to identify these padded values, ensuring they are ignored during optimization, so that only valid data points are processed by NNs.

Expanding-window forecast and rolling-window forecast Our study uses both expanding-window and rolling-window forecasting methods. In expanding-window forecasting, the model takes the entire return series as input and outputs a volatility series of the same length, where each volatility represents a one-day-ahead forecast of the corresponding return. This method enables the model to leverage all historical returns for making forecasts. In rolling-window forecasting, the model takes a fixed window of return series (e.g., the most recent 252 observations) as input and output only the volatility forecast for the final time step. This approach allows us to evaluate the performance of global NNs with different lengths of historical returns (e.g., the time step importance

study in the main chapter).

Appendix B

Appendices for Chapter 4

This appendix provides the detailed empirical results for all 31 stock indices referenced in Chapter 4.

Table B.1: Datasets specification (all indices).

	In-Sample	Out-of-Sample	T_in	T_out
AEX	2004-01-02 - 2012-12-21	2012-12-24 - 2021-12-31	2302	2302
AORD	2004-01-02 - 2013-01-02	2013-01-03 - 2021-12-31	2276	2277
BFX	2004-01-02 - 2012-12-21	2012-12-24 - 2021-12-31	2302	2302
BSESN	2004-01-01 - 2012-12-20	2012-12-21 - 2021-12-31	2233	2234
BVLG	2012-10-15 - 2017-05-18	2017-05-21 - 2021-12-31	1171	1172
BVSP	2004-01-02 - 2012-12-17	2012-12-18 - 2021-12-30	2216	2216
DJI	2004-01-02 - 2012-12-26	2012-12-27 - 2021-12-31	2258	2258
FCHI	2004-01-02 - 2012-12-24	2012-12-27 - 2021-12-31	2303	2304
FTMIB	2009-06-01 - 2015-09-08	2015-09-09 - 2021-12-30	1594	1595
FTSE	2004-01-02 - 2013-01-03	2013-01-04 - 2021-12-31	2273	2273
GDAXI	2004-01-02 - 2012-12-10	2012-12-11 - 2021-12-30	2282	2282
GSPTSE	2004-01-02 - 2013-01-03	2013-01-04 - 2021-12-31	2250	2250
HSI	2004-01-02 - 2012-12-14	2012-12-17 - 2021-12-31	2213	2214
IBEX	2004-01-02 - 2013-01-04	2013-01-07 - 2021-12-30	2292	2292
IXIC	2004-01-02 - 2012-12-31	2013-01-02 - 2021-12-31	2260	2261
KS11	2004-01-02 - 2012-12-07	2012-12-10 - 2021-12-30	2223	2223
KSE	2004-01-01 - 2013-02-11	2013-02-12 - 2021-12-31	2196	2196
MXX	2004-01-02 - 2012-12-20	2012-12-21 - 2021-12-31	2261	2261
N225	2004-01-05 - 2012-12-14	2012-12-17 - 2021-12-30	2198	2199
NSEI	2004-01-01 - 2012-12-21	2012-12-24 - 2021-12-31	2232	2233
OMXC20	2005-10-03 - 2013-11-14	2013-11-15 - 2021-12-30	2018	2018
OMXHPI	2005-10-03 - 2013-11-07	2013-11-08 - 2021-12-30	2037	2037
OMXSPI	2005-10-03 - 2013-11-08	2013-11-11 - 2021-12-30	2037	2037
OSEAX	2004-01-02 - 2012-12-06	2012-12-07 - 2021-12-30	2247	2247
RUT	2004-01-02 - 2012-12-28	2012-12-31 - 2021-12-31	2259	2259

SMSI	2005-07-04 - 2013-09-29	2013-09-30 - 2021-12-30	2101	2102
SPX	2004-01-02 - 2012-12-27	2012-12-28 - 2021-12-31	2259	2259
SSEC	2004-01-02 - 2012-12-31	2013-01-04 - 2021-12-31	2184	2185
SSMI	2004-01-05 - 2012-12-13	2012-12-14 - 2021-12-30	2258	2258
STI	2004-01-02 - 2016-11-03	2016-11-04 - 2021-12-31	1289	1289
STOXX50E	2004-01-02 - 2012-12-24	2012-12-27 - 2021-12-31	2292	2292

References

Admati, A. R., & Pfleiderer, P. (1988). A theory of intraday patterns: Volume and price variability. *The Review of Financial Studies*, *1*(1), 3–40.

Almeida, C., Fan, J., Freire, G., & Tang, F. (2023). Can a Machine Correct Option Pricing Models? *Journal of Business & Economic Statistics*, *41*(3), 995–1009.

Amihud, Y. (2002). Illiquidity and stock returns: Cross-section and time-series effects. *Journal of Financial Markets*, *5*(1), 31–56.

Andersen, T. G., & Bollerslev, T. (1998). Answering the Skeptics: Yes, Standard Volatility Models do Provide Accurate Forecasts. *International Economic Review*, *39*(4), 885–905.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2001). The distribution of realized exchange rate volatility. *Journal of the American Statistical Association*, *96*(453), 42–55.

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and Forecasting Realized Volatility. *Econometrica*, *71*(2), 579–625.

Andersen, T. G., & Bondarenko, O. (2014). VPIN and the flash crash. *Journal of Financial Markets*, *17*, 1–46.

Atkins, A., Niranjan, M., & Gerding, E. (2018). Financial news predicts stock market volatility better than close price. *The Journal of Finance and Data Science*, *4*(2), 120–137.

Bali, T. G., Beckmeyer, H., Mörke, M., & Weigert, F. (2023). Option Return Predictability with Machine Learning and Big Data. *The Review of Financial Studies*, *36*(9), 3548–3602.

Barndorff-Nielsen, O. E., Hansen, P. R., Lunde, A., & Shephard, N. (2008). Designing Realized Kernels to Measure the ex post Variation of Equity Prices in the Presence of Noise. *Econometrica*, *76*(6), 1481–1536.

Barndorff-Nielsen, O. E., & Shephard, N. (2004). Power and Bipower Variation with Stochastic Volatility and Jumps. *Journal of Financial Econometrics*, *2*(1), 1–37.

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854.

Bessembinder, H., & Venkataraman, K. (2010). Bid–Ask Spreads. In *Encyclopedia of Quantitative Finance*. John Wiley & Sons, Ltd.

Bogousslavsky, V., Fos, V., & Muravyev, D. (2024). Informed Trading Intensity. *The Journal of Finance*, *79*(2), 903–948.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327.

Bollerslev, T. (2008). Glossary to ARCH (GARCH). *CREATES Research Paper*, *2008*(49).

Bollerslev, T., Patton, A. J., & Quaedvlieg, R. (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, *192*(1), 1–18.

Bouchaud, J.-P., Farmer, J. D., & Lillo, F. (2009, January). CHAPTER 2 - How Markets Slowly Digest Changes in Supply and Demand. In T. Hens & K. R. Schenk-Hoppé (Eds.), *Handbook of Financial Markets: Dynamics and Evolution* (pp. 57–160). North-Holland.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Bucci, A. (2020). Realized Volatility Forecasting with Neural Networks. *Journal of Financial Econometrics*, *18*(3), 502–531.

Capponi, A., & Yu, S. (n.d.). Price Discovery in the Machine Learning Age.

Cartea, Á., Jaimungal, S., & Penalva, J. (2015). *Algorithmic and high-frequency trading*. Cambridge University Press.

Chamberlain, G. (1982). Multivariate regression models for panel data. *Journal of econometrics*, *18*(1), 5–46.

Chevalier, G. (2022). Supervised portfolios. *Quantitative Finance*, *22*(12), 2275–2295.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, *89*(3), 539–552.

Chordia, T., Roll, R., & Subrahmanyam, A. (2002). Order imbalance, liquidity, and market returns. *Journal of Financial Economics*, *65*(1), 111–130.

Christensen, K., Siggaard, M., & Veliyev, B. (2023). A Machine Learning Approach to Volatility Forecasting*. *Journal of Financial Econometrics*, *21*(5), 1680–1727.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014, December). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

Cont, R., Kukanov, A., & Stoikov, S. (2014). The Price Impact of Order Book Events. *Journal of Financial Econometrics*, *12*(1), 47–88.

Cont, R., Stoikov, S., & Talreja, R. (2010). A Stochastic Model for Order Book Dynamics. *Operations Research*, *58*(3), 549–563.

Corsi, F. (2009). A Simple Approximate Long-Memory Model of Realized Volatility. *Journal of Financial Econometrics*, *7*(2), 174–196.

Das, A., Kong, W., Sen, R., & Zhou, Y. (2024, April 17). A decoder-only foundation model for time-series forecasting.

Del Moral, P., Doucet, A., & Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, *22*(5), 1009–1020.

Easley, D., López de Prado, M., O'Hara, M., & Zhang, Z. (2021). Microstructure in the Machine Age. *The Review of Financial Studies*, *34*(7), 3316–3363.

Easley, D., López de Prado, M. M., & O'Hara, M. (2012). Flow Toxicity and Liquidity in a High-frequency World. *The Review of Financial Studies*, *25*(5), 1457–1493.

Easley, D., & O'Hara, M. (1987). Price, trade size, and information in securities markets. *Journal of Financial Economics*, *19*(1), 69–90.

Ellis, K., Michaely, R., & O'Hara, M. (2000). When the Underwriter Is the Market Maker: An Examination of Trading in the IPO Aftermarket. *The Journal of Finance*, *55*(3), 1039–1074.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Engle, R. (2002). New frontiers for arch models. *Journal of Applied Econometrics*, *17*(5), 425–446.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, *50*(4), 987–1007.

Engle, R. F., & Gallo, G. M. (2006). A multiple indicators model for volatility using intra-daily data. *Journal of Econometrics*, *131*(1), 3–27.

Engle, R. F., Hong, C.-H., & Kane, A. (1990, May). Valuation of Variance Forecast with Simulated Option Markets.

Engle, R. F., & Ng, V. K. (1993). Measuring and Testing the Impact of News on Volatility. *The Journal of Finance*, *48*(5), 1749–1778.

Engle, R. F., & Russell, J. R. (1998). Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data. *Econometrica*, *66*(5), 1127–1162.

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383–417.

Fama, E. F. (1991). Efficient Capital Markets: II. *The Journal of Finance*, *46*(5), 1575–1617.

Feng, G., Giglio, S., & Xiu, D. (2020). Taming the Factor Zoo: A Test of New Factors. *The Journal of Finance*, *75*(3), 1327–1370.

Fernandez, J., Wehrstedt, L., Shamis, L., Elhoushi, M., Saladi, K., Bisk, Y., Strubell, E., & Kahn, J. (2024, November). Hardware Scaling Trends and Diminishing Returns in Large-Scale Distributed Training.

Fissler, T., & Ziegel, J. F. (2016). Higher order elicitability and Osband's principle. *The Annals of Statistics*, *44*(4), 1680–1707.

Forsberg, L., & Bollerslev, T. (2002). Bridging the gap between the distribution of realized (ECU) volatility and ARCH modelling (of the Euro): The GARCH-NIG model. *Journal of Applied Econometrics*, *17*(5), 535–548.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*(1), 1–58.

Gerlach, R., & Wang, C. (2016). Forecasting risk via realized GARCH, incorporating the realized range. *Quantitative Finance*, *16*(4), 501–511.

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *The Journal of Finance*, *48*(5), 1779–1801.

Glosten, L. R., & Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, *14*(1), 71–100.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Gunawan, D., Kohn, R., & Tran, M. N. (2022). Flexible and Robust Particle Tempering for State Space Models. *Econometrics and Statistics*.

Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, *20*(7), 873–889.

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The Model Confidence Set. *Econometrica*, *79*(2), 453–497.

Hansen, P. R., & Huang, Z. (2016). Exponential GARCH Modeling With Realized Measures of Volatility. *Journal of Business & Economic Statistics*, *34*(2), 269–287.

Hansen, P. R., Huang, Z., & Shek, H. H. (2012). Realized GARCH: A joint model for returns and realized measures of volatility. *Journal of Applied Econometrics*, *27*(6), 877–906.

Hasbrouck, J. (1991). Measuring the Information Content of Stock Trades. *The Journal of Finance*, *46*(1), 179–207.

Hendershott, T., Jones, C. M., & Menkveld, A. J. (2011). Does Algorithmic Trading Improve Liquidity? *The Journal of Finance*, *66*(1), 1–33.

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

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Hyup Roh, T. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications*, *33*(4), 916–922.

Jeffreys, H. (1998, August). *The Theory of Probability*. OUP Oxford.

Jiang, F., Ma, T., & Zhu, F. (2024). Fundamental characteristics, machine learning, and stock price crash risk. *Journal of Financial Markets*, *69*, 100908.

Jiang, W., Ruan, Q., Li, J., & Li, Y. (2018). Modeling returns volatility: Realized GARCH incorporating realized risk measure. *Physica A: Statistical Mechanics and its Applications*, *500*, 249–258.

Kaniel, R., Lin, Z., Pelger, M., & Van Nieuwerburgh, S. (2023). Machine-learning the skill of mutual fund managers. *Journal of Financial Economics*, *150*(1), 94–138.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020, January). Scaling Laws for Neural Language Models.

Karpman, K., Basu, S., Easley, D., & Kim, S. (2023). Learning Financial Networks with High-Frequency Trade Data. *Data Science in Science*, *2*(1), 2166624.

Kass, R. E., Carlin, B. P., Gelman, A., & Neal, R. M. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. *The American Statistician*, *52*(2), 93–100.

Kass, R. E., & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, *90*(430), 773–795.

Kelly, B., Malamud, S., & Zhou, K. (2024). The Virtue of Complexity in Return Prediction. *The Journal of Finance*, *79*(1), 459–503.

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, *103*, 25–37.

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization.

Koenker, R., & Bassett, G. (1978). Regression Quantiles. *Econometrica*, *46*(1), 33–50.

Kwan, A., Philip, R., & Shkilko, A. (2024, October). The Conduits of Price Discovery: A Machine Learning Approach.

Kyle, A. S. (1985). Continuous Auctions and Insider Trading. *Econometrica*, *53*(6), 1315–1335.

Lee, C. M. C., & Ready, M. J. (1991). Inferring Trade Direction from Intraday Data. *The Journal of Finance*, *46*(2), 733–746.

Li, D., Clements, A., & Drovandi, C. (2021). Efficient Bayesian estimation for GARCH-type models via Sequential Monte Carlo. *Econometrics and Statistics*, *19*, 22–46.

Lillo, F., & Farmer, J. D. (2004, July). The long memory of the efficient market.

Lin, J.-C., Sanger, G. C., & Booth, G. G. (1995). Trade Size and Components of the Bid-Ask Spread. *The Review of Financial Studies*, *8*(4), 1153–1183.

Liu, L. Y., Patton, A. J., & Sheppard, K. (2015). Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes. *Journal of Econometrics*, *187*(1), 293–311.

Liu, Y. (2019). Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. *Expert Systems with Applications*, *132*, 99–109.

Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from the width. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6232–6240.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, *13*(3), e0194889.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, *11*(2), 125–139.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, *59*(2), 347–370.

Nguyen, T.-N., Tran, M.-N., & Kohn, R. (2022). Recurrent conditional heteroskedasticity. *Journal of Applied Econometrics*, *37*(5), 1031–1054.

Nguyen, T.-N., Tran, M.-N., Gunawan, D., & Kohn, R. (2023). A Statistical Recurrent Stochastic Volatility Model for Stock Markets. *Journal of Business & Economic Statistics*, *41*(2), 414–428.

Ni, C., Li, Y., & Forsyth, P. (2024). Neural network approach to portfolio optimization with leverage constraints: A case study on high inflation investment. *Quantitative Finance*, *24*(6), 753–777.

O'Hara, M. (1998, March). *Market Microstructure Theory*. John Wiley & Sons.

Rahimikia, E., Zohren, S., & Poon, S.-H. (2021). Realised Volatility Forecasting: Machine Learning via Financial Word Embedding. *SSRN Electronic Journal*.

Riccò, R., Rindi, B., & Seppi, D. J. (2023, April). Non-Stationary vs. Stationary Equilibrium in Dynamic Limit Order Markets.

Roşu, I. (2009). A Dynamic Model of the Limit Order Book. *The Review of Financial Studies*, *22*(11), 4601–4641.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2017). Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(11), 2660–2673.

Shephard, N., & Sheppard, K. (2010). Realising the future: Forecasting with high-frequency-based volatility (HEAVY) models. *Journal of Applied Econometrics*, *25*(2), 197–231.

Shi, X., Wang, S., Nie, Y., Li, D., Ye, Z., Wen, Q., & Jin, M. (2025, February 27). Time-MoE: Billion-scale time series foundation models with mixture of experts.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014, April). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., & Morcos, A. (2022). Beyond neural scaling laws: Beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, *35*, 19523–19536.

Taylor, J. W. (2019). Forecasting Value at Risk and Expected Shortfall Using a Semiparametric Approach Based on the Asymmetric Laplace Distribution. *Journal of Business & Economic Statistics*, *37*(1), 121–133.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

Upson, J., McInish, T., & Iv, B. H. J. (2021). Order based versus level book trade reporting: An empirical analysis. *Journal of Banking & Finance*, *125*, 106074.

van Kervel, V. (2015). Competition for Order Flow with Fast and Slow Traders. *The Review of Financial Studies*, *28*(7), 2094–2127.

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*, *30*.

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*, *34*, 22419–22430.

Xie, H., & Yu, C. (2020). Realized GARCH models: Simpler is better. *Finance Research Letters*, *33*, 101221.

Xing, F. Z., Cambria, E., & Zhang, Y. (2019). Sentiment-aware volatility forecasting. *Knowledge-Based Systems*, *176*, 68–76.

Zhai, X., Kolesnikov, A., Houlsby, N., & Beyer, L. (2022, June). Scaling Vision Transformers.

Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2023). Volatility Forecasting with Machine Learning and Intraday Commonality*. *Journal of Financial Econometrics*, nbad005.

Zhang, Z., Lim, B., & Zohren, S. (2021). Deep Learning for Market by Order Data. *Applied Mathematical Finance*, *28*(1), 79–95.

Zhang, Z., Zohren, S., & Roberts, S. (2019). DeepLOB: Deep Convolutional Neural Networks for Limit Order Books. *IEEE Transactions on Signal Processing*, *67*(11), 3001–3012.

Zhao, P., Zhu, H., Ng, W. S. H., & Lee, D. L. (2024). From GARCH to neural network for volatility forecast. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(15), 16998–17006.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115.