Hybrid Neural Network Models for Predicting Limit Order Executions in High-Frequency Trading

Prayash Joshi
Computer Science
Virginia Tech
Blacksburg, VA 24061-0002
prayash@vt.edu

Abstract

In this paper, we investigates various hybrid neural network models for predicting limit order executions in high-frequency trading. We focus on logistic regression, weighted logistic regression, and XGBoost with artificial neural networks (ANNs) to classify executed vs. non-executed orders. The study shows that LOB(limit order books) are complex and have a very severe class imbalance comparing executed orders to non-executed orders. Thus, the best methodologies used class weights, regularization penalties, and other optimizations to account for imbalance. The best model in the paper perfectly predicts executed status for LOB using an ensemble of XGBoost and ANN. However, this method has its own set of limitations in the real world.

1 Introduction

High-frequency trading (HFT) is a hot topic in the current finance research space. HFT research focuses on low-latency systems and high-frequency data analytics, where nanoseconds determine if an order is set to be executed. Thus, The ability to process and analyze large financial data in real-time has become increasingly crucial for trading firms to gain a competitive edge. This study aims to contribute to the growing body of research in this field by developing a hybrid neural network classification model to predict the execution of limit orders in high-frequency trading environments. While building a low latency model to update the trading algorithm is a very viable research topic, financial institutions trade in the scale of billions of dollars. There need to be some human intervention to interpret the model's output and insights. Thus, this paper will dive deeper into gaining insight from LOB(limit order books) about the factors that lead to orders being executed.

1.1 Problem Relevance

AI has been growing at an astronomic pace and the amount of data we deal with doubles every year. The rapid growth of electronic trading has led to an explosion of available market data. Among the is High Frequency Trading, where the challenge is to build the fastest and best algorithms for executing orders at nanosecond speeds. The ability to utilize this data and extract meaningful insights provides an advantage in this field. By accurately predicting the likelihood of a limit order being executed, traders can optimize their strategies, reduce latency, and potentially improve their overall profitability. While there is a common belief that being first to market is the key to successful order execution, the reality is more complex. A multitude of factors, such as the bid price, timing, and size of the order, play crucial roles in determining whether an order will be filled. This complexity highlights the need for sophisticated models that can capture both linear and non-linear relationships among these variables. Our study aims to draw insights on what factors lead to orders being executed.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

1.2 Proposed Approach

In this study, we propose a hybrid neural network model that combines the strengths of logistic regression and artificial neural networks (ANNs). Logistic regression is suited for classification tasks where we want to capturing linear relationships between factors such as order size, price, and timing. ANNs are proficient at modeling non-linear interactions and can uncover complex hidden patterns in the data. However, the ANNs need to be optimized across a multitude of hyper-parameters and can be computationally expensive as the amout of data increases.

By leveraging the coverage of these two approaches, we aim to develop a robust classification model that can accurately predict the probability of a limit order being executed. Before we perform our analysis, its essential to carefully consider the characteristics of the dataset and the limitations of the chosen models and parameters. This study explores the considerations in detail, providing insights into the development and evaluation of the proposed hybrid model. The paper is structured as follows: Section 2 provides an in-depth exploration of the datasets, highlighting key features and patterns. Section 3 provides a comprehensive literature review, discussing relevant research in the field of high-frequency trading and machine learning applications. Section 4 outlines the methodology employed. Section 5 outlines the architecture of the proposed hybrid model. Section 6 presents the results of our experiments, comparing the performance of the hybrid model against standalone logistic regression and ANN models. Finally, Section 7 concludes the paper, summarizing our findings and discussing potential avenues for future research.

2 Data Exploration

The data used in this study are two high-frequency limit order book datasets sourced from LOBSTER. We will specifically look at Microsoft (MSFT) and Apple (AAPL) stocks. The MSFT dataset contains 141,506 rows and 6 columns and the AAPL dataset has 91,996 rows and 6 columns. One of the key limitations was the dataset size. LOBSTER's paid plan would provide access to millions of rows of data. However, I opted with the sample data used, which still consists of a lot of information to analyze at a high level.

2.1 Data Structure

Each dataset consists of the following six columns:

- **Time**: Seconds after midnight with decimal precision.
- **Type**: Indicates the type of order event. 1: Submission of a new limit order; 2: Cancellation (partial deletion of a limit order); 3: Deletion (total deletion of a limit order); 4: Execution of a visible limit order; 5: Execution of a hidden limit order; 7: Trading halt indicator.
- **Order ID**: Unique order reference number assigned in order flow.
- **Size**: Number of shares in the order.
- **Price**: Dollar price times 10,000 (i.e., a stock price of $91.14 is given by 911400).
- **Direction**: Indicates whether the order is a buy or sell limit order. -1: Sell limit order; 1: Buy limit order.

Note: that the execution of a sell (buy) limit order directly relates to a buyer (seller) initiated trade, in other words, a buy (sell) trade. Also the *StartTime* and *EndTime* variables in the dataset are limited because the theoretical beginning and end time of the output file in milliseconds after midnight and *LEVEL* refers to the number of levels of the requested limit order book.

2.2 MSFT - Microsoft LOB Data

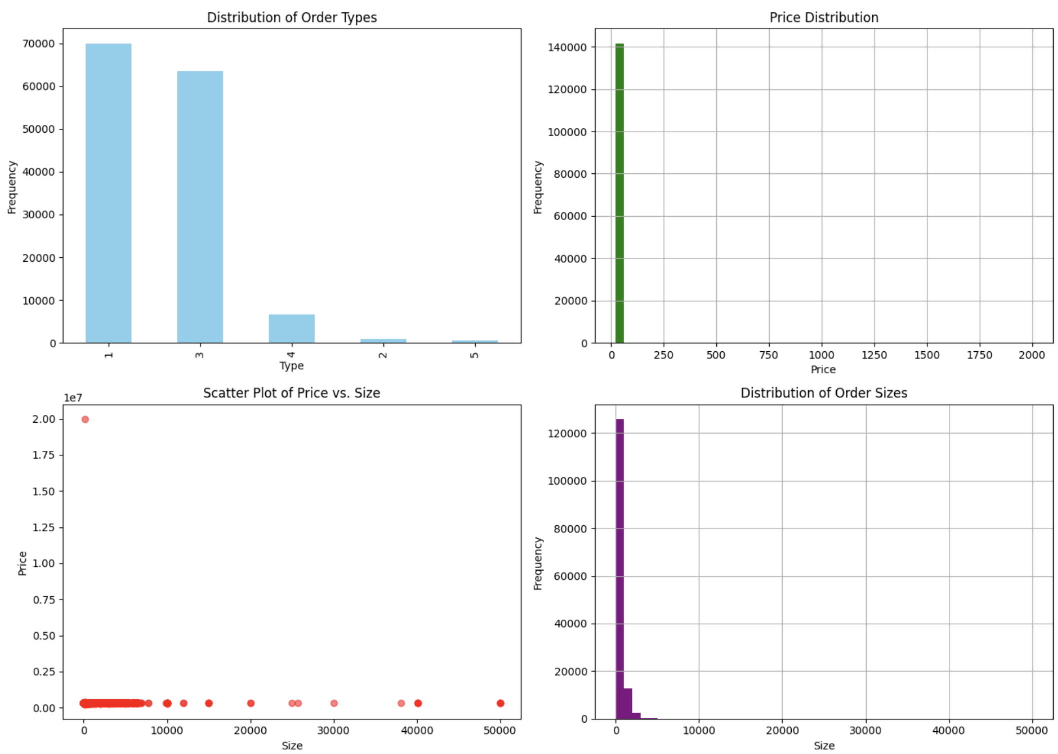

Figure 1: Exploratory Data Analysis for Microsoft LOB Data

Figure 14 presents an exploratory analysis of the MSFT dataset. The visualizations reveal several key characteristics of the data. First, there is a high frequency of small-sized orders, indicating that most of the trading activity involves relatively low volumes. Second, the price variance is not substantial, suggesting that the stock price remains relatively stable within the observed time frame. Finally, Type 1 (submission of new limit orders) and Type 3 (total deletion of limit orders) events constitute the majority of the dataset, implying that a significant portion of the order flow involves the placement and cancellation of limit orders.

2.3 AAPL - Apple LOB Data

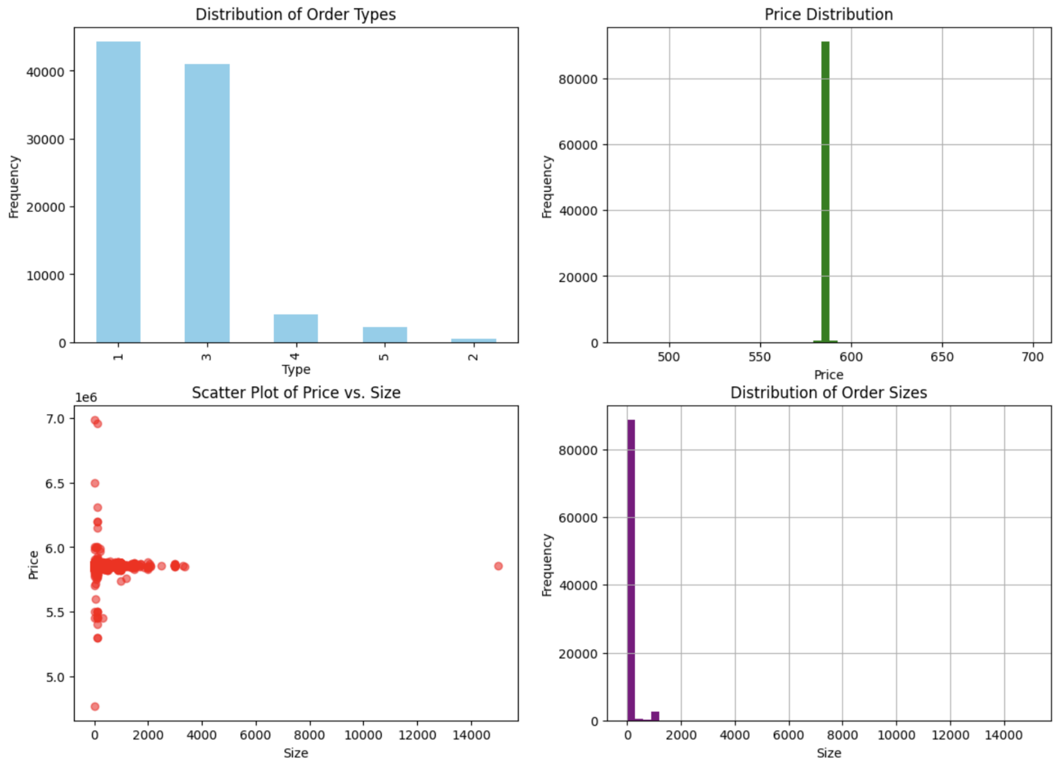

Figure 2: Exploratory Data Analysis for Apple LOB Data

The exploratory analysis of the AAPL dataset, as shown in Figure 2, reveals similar patterns to those observed in the MSFT dataset. The AAPL data also exhibits a high frequency of small-sized orders, relatively stable prices, and a dominance of Type 1 and Type 3 order events. These similarities suggest that the two datasets share common characteristics, which may be representative of high-frequency limit order book dynamics in general.

3 Literature Review

3.1 Limit Order Books: Mechanics and Challenges

Limit Order Books (LOBs) serve as the primary data format for facilitating trade in many of the world's major financial markets. This includes the NYSE, NASDAQ, LSE, and Tokyo Stock Exchange [2]. This paper introduces the mechanics and challenges of using limit orderbooks. By definition, an LOB is a system that matches buyers and sellers of an asset by maintaining a record of unexecuted limit orders waiting to be filled. In the paper, Gould et al. provide a precise mathematical formulation of the LOB mechanism, defining key terms such as order size, price, priority, bid and ask prices, and market depth.

The authors also discuss the economic benefits of LOBs. The ability to facilitate trade between patient and impatient traders introduces an interesting dynamic of what drives an order to be executed and how price, time and size can be leveraged to meet that demand. The authors describe how limit orders can be viewed as free options offered to the market. However, studying LOBs is a challenge due to its high dimensionality and complex dependence between order flows and LOB states(submitted, deleted, executed, etc.). There are presences of hidden liquidity and its hard to estimate volatility based on these factors. But, traders and alogirhtm builders are interested in studying the impact of LOB resolution parameters on trading behavior and market dynamics.

3.2 Machine Learning and Data Science Applications in LOBs

In the past decade, AI advancements have been powered by the availability of faster, cheaper computing power have opened up new possibilities for modeling and simulating LOBs using modern machine learning techniques [4]. Chip manufactures like Nvidia are pushing the limits of what is possible, to an extent where experts are worried about energy. These models are build around big data collected across the internet protocals and online traffic. However, its not feasable to have completely labeled datasets of all fields and subfields, which is why researchers use simulated datasets that exhibit properties of real datasets. Jain et al. emphasize the importance of these simulations for calibrating and fine-tuning automated trading strategies, which aligns with our goal of predicting limit order executions.

In the paper, authors classify LOB simulation models based on their methodology, including Point Processes, Agent-Based Modeling, Deep Learning, and Stochastic Differential Equations. These approaches guide the design of our hybrid models we are interested in this study. Additionally, the paper discusses "stylized facts", in other words, empirically observed LOB statistics that to improve their realism and performance of complex models.

Jain et al. also highlight the importance of responsiveness to exogenous trades or Market/Price Impact. He suggests that practically applicable LOB simulator needs to be Market Impact aware because we want to avoid poor out-of-sample performance. This means market impact could be a crucial variable in our models for the real-world effectiveness index of LOB predictions.

Jain et al. provides a comparative analysis of various LOB models' goodness of fit and performance against empirical data. They identify that recent novel LOB simulators that leverage generative modeling techniques like GANs open an avenue for future research in integrating these advanced models into our hybrid approach and considering the complex mechanics of limit order books.

3.3 Logistic Regression and Neural Networks in Related Fields

For various clasificaion and predictions tasks, Logistic Regression (LR) and Artificial Neural Networks (ANNs) have been widely applied to various fields. There is a plethora of reserch done on the effectiveness, limitations and benefits of each approach. Dreiseitl and Ohno-Machado [1] provide a comprehensive review of these two methodologies in the field of biomedical data, discussing the similarities, differences, and key considerations for training and evaluation.

In the paper, the authors notes that a neural network without a hidden layer is comparable to logistic regression model if using a logistic activation function. Adding hidden layers makes the ANN flexible and nonlinear compared to LR. This understanding is important for developing our hybrid models that combine these two approaches.

Dreiseitl and Ohno-Machado also discuss variable selection methods for LR and techniques to avoid overfitting in ANNs, such as regularization, early stopping, and Bayesian approaches. These considerations are likely relevant for training the component models in our hybrid approach.

In analyzing a sample of papers comparing LR and ANN models on medical data, the authors find that both models often perform at a similar level, with ANNs outperforming LR in some cases likely due to their greater flexibility. Thus, ANNs likely provide better performance reulsts in some cases but they are coparable. The flexibility of ANNs may be more advantageous for our complex LOB data, but it will be important to compare hybrid models against LR and ANN baselines to demonstrate the benefit of our approach.

3.4 Hybrid Models Combining Logistic Regression and Neural Networks

Tunç [5] proposes a hybrid model that combines Logistic Regression and feedforward neural networks (FNNs) for lung cancer classification. The study shows that there exists a potential benefits of integrating these two methods into a hybrid model. The proposed approach consists of two stages: first, an LR model is used to obtain initial classification results and determine significant covariates, and a FNN is trained using significant covariates identified by the LR model.

In the disucssions, the author attributes the strong performance of the hybrid model to the effective combination of LR for covariate selection and FNN for nonlinear modeling. By using only the significant covariates identified by LR, the hybrid approach reduces the input dimensionality for the

FNN. This improved the accuracy and generalization. Although the domain differs from financial LOB data, the application of methods and success of this hybrid LR-FNN model in improving classification accuracy compared to just LR and FNN standalone models is a good sign. We can utilize a similar approach, which could be beneficial for our limit order predictions. Based on the nature of LOB data, I do believe using Logistic Regression for feature selection and dimensionality reduction can be helpful before passing it to a neural network.

3.5 XGBoost for Imbalanced Data

Zhang et al. [6] investigate the application of XGBoost. XGBoost is a well known powerful gradient boosting algorithm. It can be used for classification and regression tasks. The authors highlight XGBoost's advantages, such as high flexibility, strong predictability, generalization ability, scalability, efficiency, and robustness. Considering the nature of LOB data, XGBoost is promising for our hybrid models, given the large-scale and complex nature of LOB data. But this is also one of the downsides, XGBoost is not feasable for extremely large datasets.

Despite the optimizations in its objective functions, the authors states that XGBoost's performance can suffer on imbalanced datasets. To address this issue, the authors propose combining data-level resampling methods like SVM-SMOTE oversampling and EasyEnsemble undersampling with XGBoost, as well as optimizing XGBoost's regularization term and using Bayesian optimization for hyperparameter tuning.

In their experiments, Zhang et al. find that the proposed XGBoost method outperforms other leading boosting algorithms on imbalanced data. The results suggest that optimized XGBoost can improve our model greatly. They also show that the combination of resampling techniques and XGBoost optimizations is more effective than using mixed sampling alone. This means both resampling and optimizations play a key role in achieving the improved performance. Thus, this paper highlight XGBoost's potential for handling the class imbalance problem, which we can apply to our LOB data. The authors also propose and discuss the benefits of incorporating techniques like resampling, regularization, and hyperparameter tuning when integrating XGBoost into our hybrid models.

3.6 Research Gap and Proposed Approach

From the literature review, it is clear that research on modeling and simulating LOBs involves using various machine learning techniques. Standalone models are likely to perform sub optimally. Thus, many researchers and papers have proposed hybrid combination of models that incorporate regualrization, oversampling, weighting, penalties etc.

The challenges posed by the high dimensionality, non-stationarity, and class imbalance of LOB data call for a hybrid approach that combines the strengths of different methodologies. By integrating logistic regression for feature selection and interpretability, XGBoost for handling class imbalance and capturing complex relationships, and artificial neural networks for learning non-linear patterns, we aim to develop a powerful and flexible framework for predicting limit order executions in high-frequency trading environments. Not all models need to be in the same archtecture. It would be very inefficient, as we know ANNs will have similar results to logistic regressino if we arent carefull.

Our proposed approach builds upon the findings and insights from previous studies, incorporating techniques such as regularization, hyperparameter tuning, and class weights to address the specific challenges of LOB data. The success of hybrid models in other domains, like the LR-FNN model for lung cancer classification [5], can give us confidence in our architecture. Firstly, we will start with standalone models, and then move on to the neural network integration.

4 Methodology

4.1 Logistic Regression

Logistic regression is a widely used statistical method for binary classification problems. It models the probability p of an event occurring based on a set of independent variables x. In the context of predicting limit order executions, logistic regression captures the linear relationships between factors such as order size, price, and timing, and the probability of an order being executed.

The logistic regression model estimates the parameters of a logistic function, which is defined as:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

where $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients.

The model is trained by minimizing the logistic loss function, typically using optimization algorithms such as gradient descent. The logistic loss function is given by:

$$L(\beta) = -\sum_{i=1}^{N} [y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))]$$

where y_i are the class labels.

In the financial domain, logistic regression has been applied to various problems such as credit scoring [1], bankruptcy prediction [5], and fraud detection. For high-frequency trading, logistic regression can be used to model the linear relationships between order characteristics and execution probabilities. However, as noted by [3], financial time series often exhibit non-linear dynamics, which may limit the effectiveness of logistic regression in capturing complex patterns in limit order data.

To address this limitation, researchers have explored techniques to enhance the flexibility of logistic regression models. One approach is to include interaction terms between the independent variables, allowing the model to capture some non-linear relationships [1]. Another strategy is to combine logistic regression with more advanced non-linear models, such as artificial neural networks, in a hybrid approach [5].

4.2 XGBoost

XGBoost (Extreme Gradient Boosting) is an ensemble method that combines multiple weak learners, typically decision trees, to create a strong predictive model. It is known for its performance and efficiency in various types of data.

Algorithm 1 XGBoost Algorithm

1: [t] Initialize model with a constant value: $f_0(x) = \arg\min_\gamma \sum_{i=1}^{n} l(y_i, \gamma)$
2: **for** $t = 1$ to T **do**
3: [t] Compute residuals: $r_{it} = -\left[\frac{\partial l(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{t-1}(x)}$
4: [t] Fit a new model $h_t(x)$ to predict the residuals r_{it}
5: [t] Update the model: $f_t(x) = f_{t-1}(x) + \eta h_t(x)$
6: **end for**

In recent years, XGBoost has gained popularity in the financial industry due to its ability to handle large-scale, high-dimensional data and its strong predictive performance. [6] highlight the advantages of XGBoost, including its flexibility, scalability, and robustness, making it well-suited for complex financial modeling tasks.

XGBoost can capture complex non-linear interactions among the input features, such as order size, price, and timing on our LOB dataset. By leveraging its ability to handle high-dimensional data and its built-in regularization techniques, XGBoost is both simiple and great for handling data like our financial LOB dataset

4.2.1 Features

- **Regularization:** Includes L1 and L2 regularization to prevent overfitting.
- **Handling Missing Values:** Learns the best direction to move when a missing value is encountered.
- **Scalability:** Efficiently utilizes multiple CPU cores during training.

4.3 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are inspired by biological neural networks. ANNs consist of layers of interconnected nodes, or neurons, each of which applies a non-linear transformation to its input and passes the output to the next layer.

The standard feedforward network, or multi-layer perceptron (MLP), involves layers structured as follows:

- **Input Layer:** Receives the input data.
- **Hidden Layers:** One or more layers that learn abstract data representations.
- **Output Layer:** Produces the final predictions.

The learning process involves backpropagation, where the network's weights are adjusted based on the gradient of the loss function. For binary classification, the binary cross-entropy loss is used:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where $\hat{y}_i$ is the predicted probability and y_i is the actual class label.

ANNs have been widely applied in finance for tasks such as stock price prediction, portfolio optimization, and risk assessment. In the context of limit order books, ANNs can learn complex, non-linear relationships between order characteristics and execution probabilities. [4] discuss the recent advancements in using deep learning techniques, including ANNs, for limit order book modeling and simulation. They highlight the ability of ANNs to capture intricate patterns and dynamics in high-frequency trading data.

However, training ANNs on limit order book data presents several challenges. The high dimensionality and non-stationarity of the data require careful feature engineering and model design. Additionally, the class imbalance problem necessitates the use of appropriate techniques, such as cost-sensitive learning or data resampling, to ensure the model learns to predict both executed and non-executed orders effectively.

5 Proposed Hybrid Models

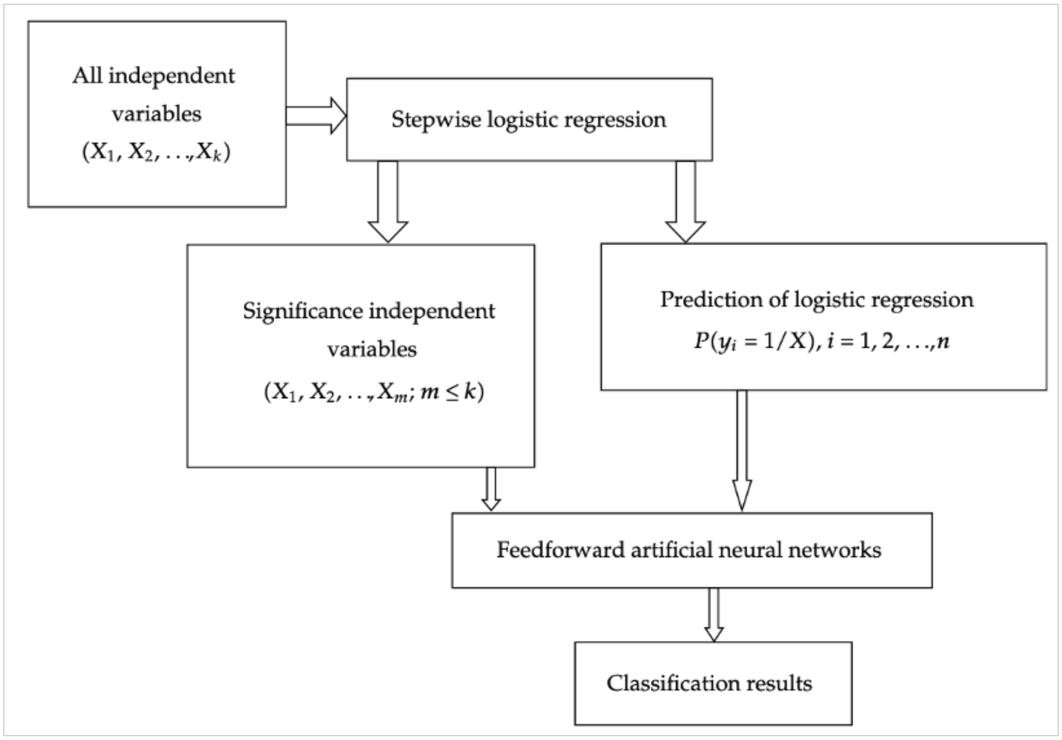

Figure 3: Initial Proposed Architecture

6 Results

6.1 MSFT - Microsoft LOB Data

Table 1: Performance Metrics for MSFT Data

Method	AUC	Accuracy	Precision	Recall
Logistic Regression	0.6779	0.95	0.08	0.00
Weighted Logistic Regression	0.6779	0.58	0.08	0.66
XGBoost	0.9275	0.85	0.23	0.87

For the MSFT dataset, XGBoost outperforms both logistic regression and weighted logistic regression across all metrics. XGBoost achieves an AUC of 0.9275, accuracy of 0.85, precision of 0.23, and recall of 0.87. Weighted logistic regression improves upon the recall of standard logistic regression (0.66 vs. 0.00) but at the cost of lower accuracy and precision.

6.2 AAPL - Apple LOB Data

Table 2: Performance Metrics for AAPL Data

Method	AUC	Accuracy	Precision	Recall
Logistic Regression	0.5872	0.93	0.00	0.00
Weighted Logistic Regression	-	0.58	0.09	0.57
XGBoost	0.9263	0.87	0.32	0.81

Similar to the MSFT dataset, XGBoost demonstrates the best overall performance on the AAPL dataset, with an AUC of 0.9263, accuracy of 0.87, precision of 0.32, and recall of 0.81. Weighted logistic regression again trades off accuracy and precision for improved recall compared to standard logistic regression.

6.3 Combined LOB Data

Table 3: Performance Metrics for Combined Data

Method	AUC	Accuracy	Precision	Recall
Logistic Regression	0.6336	0.94	0.00	0.00
Weighted Logistic Regression	-	0.61	0.08	0.57
XGBoost	0.8992	0.82	0.22	0.81
ANN	0.7278	0.9429	0.4663	0.0471
Weighted LogReg + ANN	0.7144	-	0.1203	0.5592
XGBoost + ANN	1.0000	0.9999	1.0000	0.9985

For the combined dataset, the performance metrics show the advantages of hybrid and ensemble methods over simpler models. The basic Logistic Regression model, while achieving a high accuracy of 0.94, fails to predict any executed orders correctly, as indicated by zero precision and recall.

Weighted Logistic Regression shows a significant improvement in recall (0.57), suggesting it is better at identifying executed orders. However, the precision remains low (0.08). This means a indicating a high number of false positives, which is very costly and dangerous in trading.

XGBoost continues to perform robustly, with substantial gains in both precision (0.22) and recall (0.81) compared to logistic regression models. Its ability to handle diverse data characteristics and complex interactions between features makes it well-suited for this task.

The standalone ANN model, despite high accuracy (0.9429), shows limited success in precision (0.4663) and very low recall (0.0471). This indicates that ANN can correctly label most non-executed orders but struggles to identify the executed orders. So its level with logistic regression.

The hybrid model combining Weighted Logistic Regression and ANN shows an improvement in recall (0.5592) compared to standalone models, but precision is still very low (0.1203). In this scenario, we want precision to be fairly high. The hybrid Weighted Logistic Regression and ANN model has some success in capturing the executed orders.

The XGBoost + ANN model shows excellent performance, with nearly perfect scores across all metrics (AUC, Accuracy, Precision, Recall all approaching 1). This model effectively uses the strengths of both XGBoost's robust classification abilities and ANN's pattern recognition capabilities.

As discussed, the computational cost of this model makes it less feasible for real-time high-frequency trading applications.

These results suggest that while advanced hybrid models like XGBoost + ANN offer exceptional accuracy, with near perfect precision and recall.

7 Discussion and Future Work

Insights and Model Performance : The methods and literature review in this study has provided interesting insights into the prediction of limit order executions in a simulation of a high frequncy environment. We looked various standalone and hybrid models. The main goal was to determine what factors were significant contributors of orders being executed. The result of this study is intended to be used for interpretation and future research in this field. Our data had significant class imbalance in executed and non-executed orders. This heavily influenced our initial logistic regression model. This issue is reflective of the challenges of doing a study on LOB data. Thus, dealing with class imbalance and feature processing were integral.

Advancements in Model Techniques: Since our weighted logistic regression had poor precision and recall, it was not great at predicting executed orders despite having a high accuracy. Using class-weighted logistic regression and XGBoost with class weights and regularization penalties did show improvements and captured come more complex feature interactions. However, our weighted logistic regression had poor precision and recall, it was not great at predicting executed orders despite having a high accuracy. Comparing it to the performance of XGBoost, its clear that XGBoost was the better standalone model for its robustness in classifying on LOB data. Its precision and recall values were high enough to be considered reasonable and it boasted a high AUC and accuracy scores. This lead into our investigation for hybrid model. Among the hybrid models we looked at, XGBoost + artificial neural networks (ANN), achieves near-perfect classification accuracy.

Computational Considerations and Practical Implications: XGBoost + ANN model had great success with classifying executed orders but tradeoff of computational complexity and resource demands of the hybrid model vs. the accuracy of alternative models needs to be considered. This introduce challenges in real-time high-frequency trading settings, where decision-making speeds are more important. Considering the methods developed thus far in this paper, the tradeoff for computational complexity makes sense but I believe there is space for improvement in our logistic regression hybrid model.

Future Research Directions: This study highlights several areas for future improvements. Firstly, including domain-specific data like market sentiment, news events, and cross-asset correlations could potentially improve the predictive capabilities of our models and relatability of our models to the real world. Secondly, investigating the parts of XGBoost and ANN that give us a perfect classification can be very beneficial. This can help us optimize and create less computationally expensive models. I would like the future work to focus on enabling traders to understand the underlying drivers of limit order executions more clearly.

However, market dynamics continue to evolve, stressing the need for real time data injections into our model to update every week. One of the suggestions made my the audience at the final report presentation was to use more simpler data which is more accessible and free. I agree with my colleague as this LOB dataset was interesting to study but accessibility was an issue. Future studies should also focus on the interpretability of models to facilitate deeper insights into the decision-making processes in high-frequency trading, building a interactive sandbox for simulating LOB algorithms as I acquire more information on its complex structure.

In conclusion, this study has made strides in understanding and predicting limit order executions in high-frequency trading environments. By tackling class imbalance, utilizing advanced modeling techniques, and proposing computationally efficient solutions, we have laid a strong foundation for further research in this field.

8 Figures and Supplemental Work

8.1 Data Exploration

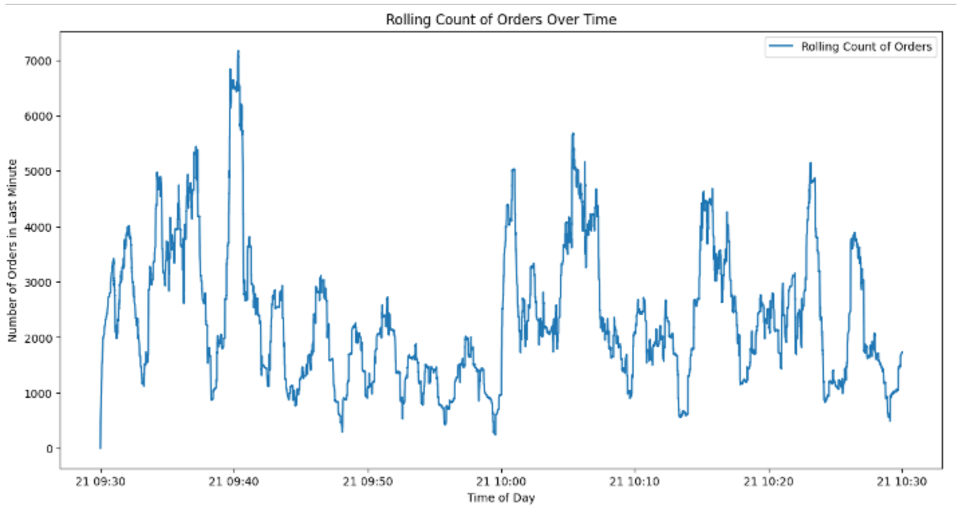

Figure 4: Rolling Count of Orders Over Time

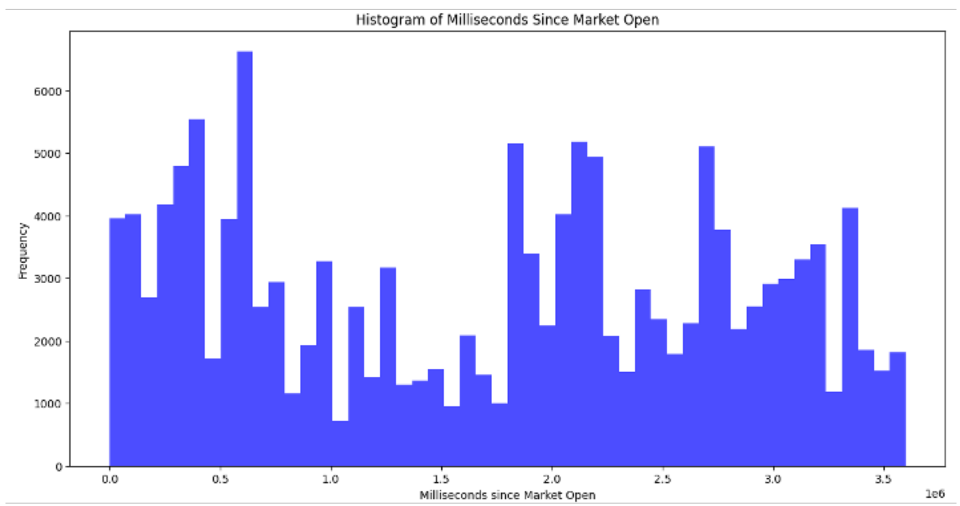

Figure 5: Histogram of Milliseconds Since Market Open

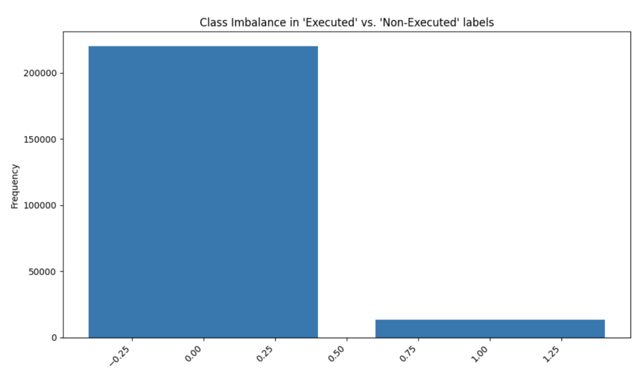

Figure 6: Histogram of Milliseconds Since Market Open

8.2 MSFT - Summary Plots of XGBoost

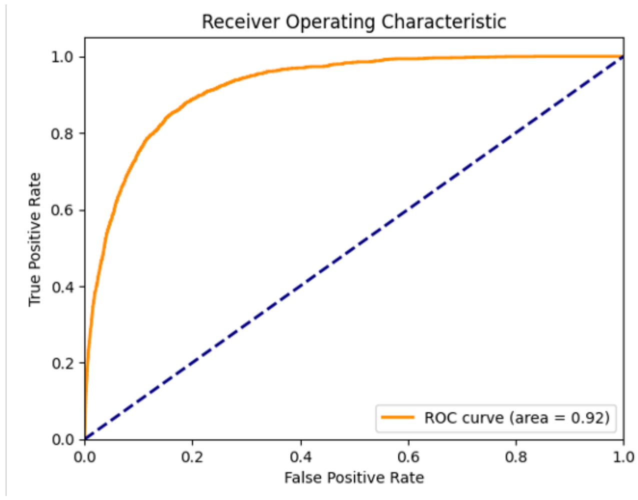

Figure 7: ROC Curve for MSFT

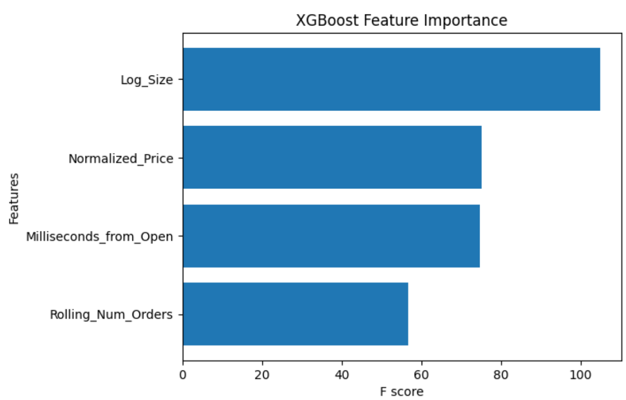

Figure 8: Feature Importance for MSFT

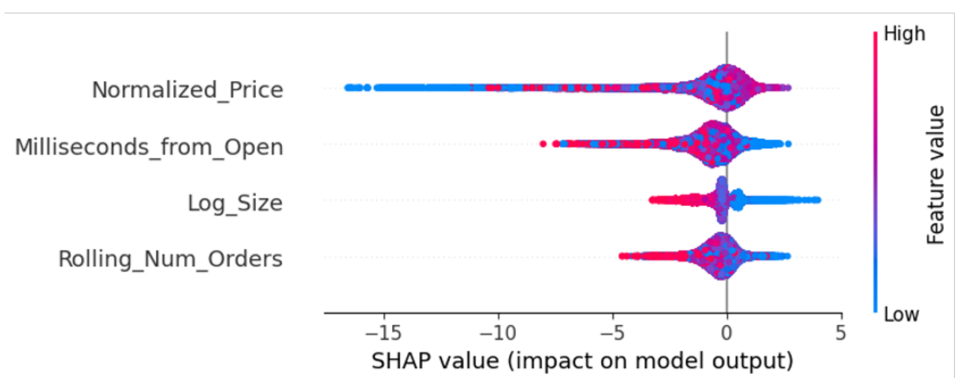

Figure 9: SHAP Values for MSFT

8.3 AAPL - Summary Plots of XGBoost

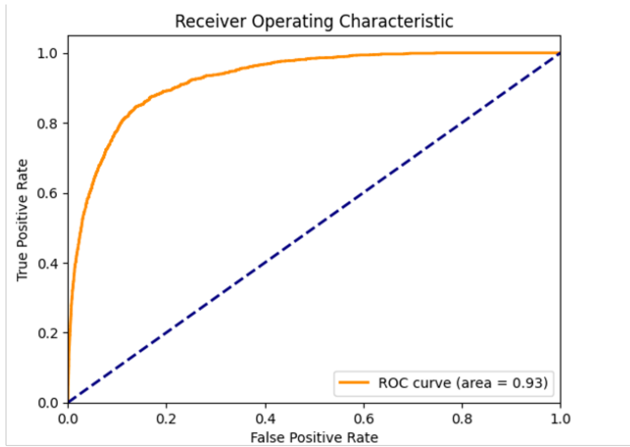

Figure 10: ROC Curve for AAPL

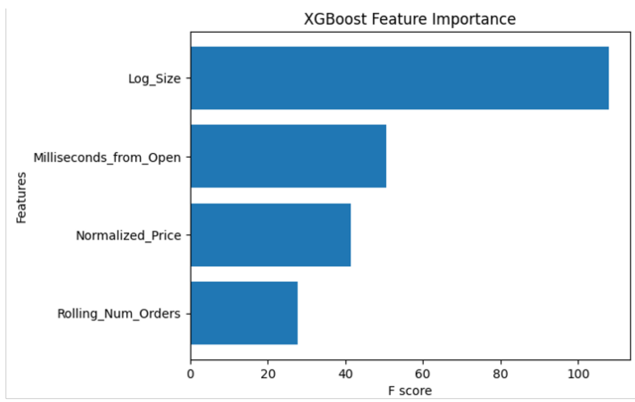

Figure 11: Feature Importance for AAPL

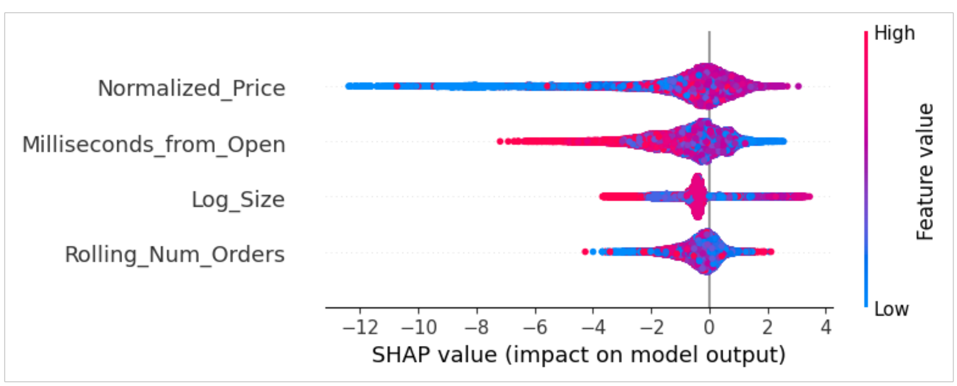

Figure 12: SHAP Values for AAPL

8.4 Standalone ANN - Summary Statistics

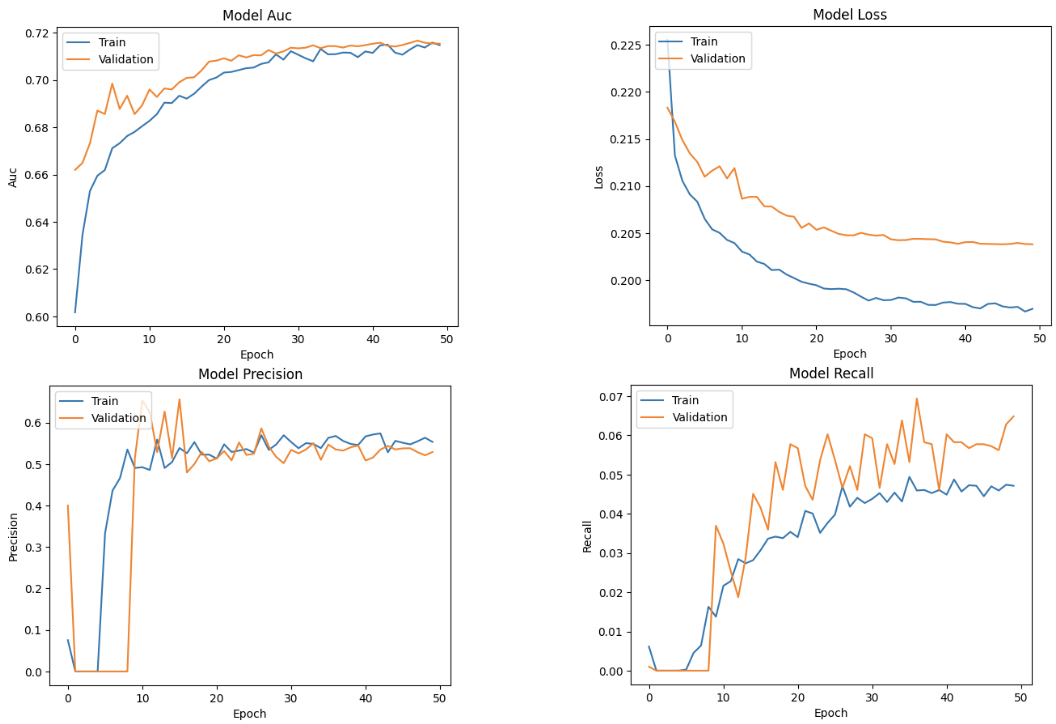

Figure 13: ANN Model Recorded Metrics across Epochs

8.5 XGBoost + ANN - Summary Statistics

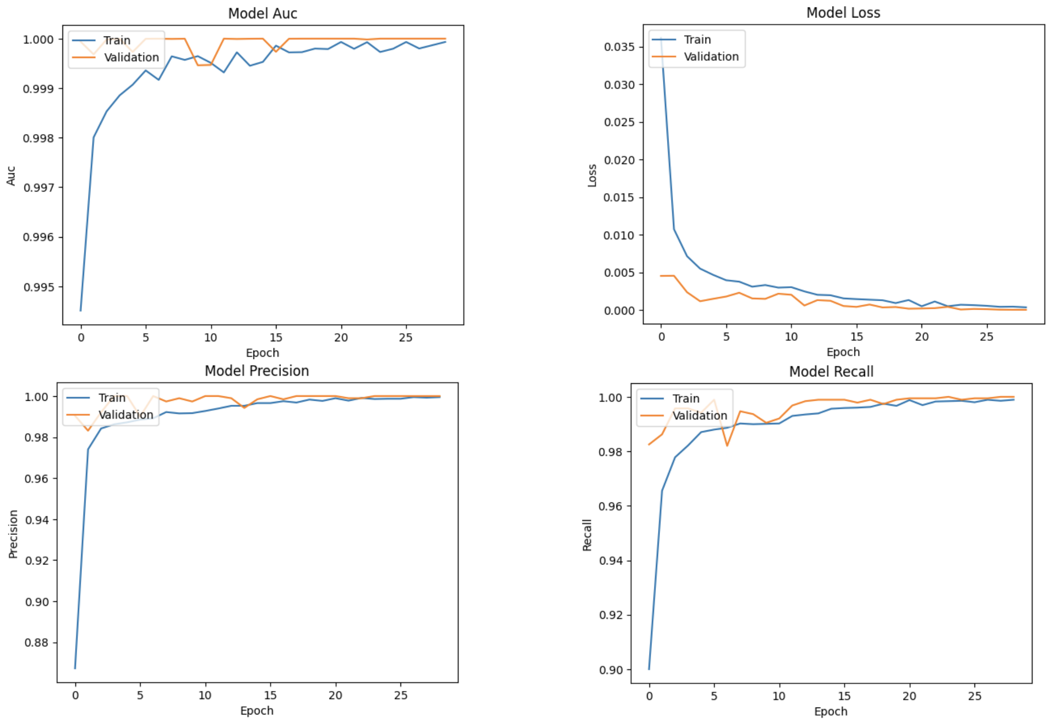

Figure 14: XGBoost + ANN Model Recorded Metrics across Epochs

9 Code Availability

The source code and additional resources used in this study are available on GitHub below:

https://github.com/PrayashJoshi/LOB-Order-Execution-Classifier

References

[1] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, 35(5):352–359, October 2002.

[2] Martin D. Gould, Mason A. Porter, Stacy Williams, Mark McDonald, Daniel J. Fenn, and Sam D. Howison. Limit order books. *Quantitative Finance*, 13(11):1709–1742, November 2013.

[3] Clive Granger and Timo Teräsvirta. *Modelling Non-Linear Economic Relationships*. Oxford University Press, 1993.

[4] Konark Jain, Nick Firoozye, Jonathan Kochems, and Philip Treleaven. Limit order book simulations: A review. *arXiv*, March 2024.

[5] Taner Tunç. A new hybrid method logistic regression and feedforward neural network for lung cancer data. *Mathematical Problems in Engineering*, 2012:1–10, 2012.

[6] Ping Zhang, Yiqiao Yia, and Youlin Shang. Research and application of xgboost in imbalanced data. 2022.