Course Project Report

High Frequency Price Prediction of Index Futures

Submitted By

Abhaysingh Rajput (221AI002)
Rohil Sharma (221AI033)

as part of the requirements of the course

Data Science (IT258) [Dec 2023 - Apr 2024]

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in Artificial Intelligence

under the guidance of

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

undergone at

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

DEC 2023 - APR 2024

High Frequency Price Prediction of Index Futures

Abhaysingh Rajput[1], Rohil Sharma[2]

***Abstract*— Accurately forecasting short-term price movements in financial markets through the use of high-frequency trading data requires careful data preparation. This paper describes a comprehensive procedure for preparing high-frequency order book data for 1-second price change prediction modeling in futures contracts.**

To comprehend the properties of the data, such as distributions, patterns, and correlations between variables, it is first examined and visualized. Techniques such as correlation analysis aid in determining the most significant factors. Methods are used to identify and manage outlier values that have the potential to skew the data. To improve the accuracy of price changes predictions, new variables such as liquidity indicators are created using the available data. Advanced algorithms are used to fill in the missing data points by estimating the missing values from similar data points. The initial data distributions are maintained in this way. Prior to modeling, the data is lastly normalized to place all variables on the same scale. Through these processes of rigorous processing, the raw order book data is optimized to feed into machine learning models such as gradient boosting and random forests. Building reliable forecasting models from high-frequency trade data requires meticulous data preparation.

Keywords: **keyword 1, keyword 2**

link of Overleaf project: `https://www.overleaf.com/6598131753xhtxfjdbybbv#451935`

I. INTRODUCTION

Forecasting future price fluctuations in the financial markets is a highly sought-after goal. The emergence of high-frequency trading and the accessibility of detailed order book information have created new opportunities to investigate the complex dynamics and patterns that influence market behavior. This paper explores the challenging problem of predicting 1-second price fluctuations for a futures contract by utilizing the abundance of data seen in high-frequency order book snapshots.

II. LITERATURE SURVEY

A. Mostafa Goudarzi / 2023

We aim to identify High-Frequency Trading (HFT) using selected features and a suitable machine learning algorithm. Our goal is to achieve Automated Market-Making (AMM) classification with fuzzy logic. After identifying the optimal algorithm, we test it on different trading day data and interpret the results.

1) Advantages: The empirical results from the evaluation of the models established that the ensemble model combined with random undersampling among several supervised learning approaches is the most effective method for detecting HFTs in order book data.

2) Limitation: Comparitive study to identify optimal algorithm

B. Rajan Lakshmi A/2017

Explores how Low latency trading impact on Indian capital markets and explores the merits and demerits of High frequency trading. A Study on the Impact of HFT on Institutional Investors, retail investors and small traders.

1) Advantages: Benefits of HFT algorithms include increased liquidity, better efficiency, increased returns for investors,lower volitality.

2) Limitation: Harmful HFT Algorithms that might cause market disruption such as high Intra-Day volatility and high Order-to-Trade Ratio

C. Spyros Makridakis, Evangelos Spiliotis/ March 27, 2018

The methodology involves data preprocessing to enhance forecasting accuracy through techniques like seasonal adjustments, transformations, and trend removal. It also explores multiple-horizon forecasting approaches, including iterative, direct, and multi-neural network methods, evaluating their accuracy and computational complexity across various forecasting horizons. Challenges such as overfitting and computational complexity are addressed, with future research directions proposed for improvement.

1) Advantages: The methodology improves forecasting accuracy through diverse preprocessing techniques and multiple-horizon forecasting approaches while addressing challenges like overfitting and computational complexity, laying the groundwork for further advancements in forecasting methodologies.

2) Limitation: The methodology faces challenges including computational complexity, potential overfitting, manual preprocessing requirements, and lack of uncertainty estimation in ML forecasts.

III. OUTCOME OF LITERATURE SURVEY

1) Paper-1: The empirical results from the evaluation of the models established that the ensemble model combined with random undersampling among several supervised learning approaches is the most effective method for

detecting HFTs in order book data.
The model showed a very robust performance in classifying data from five other trading days during a week and performed very well on all those days.

2) Paper-2: With the rise of algo trading and high-frequency trading (HFT), there have been notable improvements in various aspects of market dynamics. These include reduced transaction costs, volatility, and a more balanced buy-sell ratio. Market efficiency has improved, aiding in better price discovery. Colocation services have also contributed by minimizing latency and creating a level playing field for HFT market participants.

However, leveraging advanced technology for algo trading and HFT requires substantial technical expertise and resources. The lack of proper controls has introduced systemic risks, with the potential for significant deviations from healthy market prices due to errors or faulty algorithms. This poses a challenge for traditional investors who prioritize fundamental analysis.

HFT firms often utilize specialized services such as colocation facilities and direct access to raw market data feeds. While HFT can contribute to price fluctuations and short-term volatility, it also brings efficiencies to market operations.

3) Paper-3: The literature survey highlights the comparative forecasting performance of machine learning (ML) and statistical methods. It reveals that while ML methods offer computational advantages, they often fall short in accuracy compared to simpler statistical models. The survey underscores the need for ML methods to improve accuracy, reduce computational complexity, automate preprocessing, and provide uncertainty estimates for practical forecasting applications.

IV. Problem Statement

To Employ sophisticated data preprocessing methodologies and to meticulously analyze, visualize, refine data,hence rendering it optimal for predictive modeling purposes and choosing suitable model for predictions.

V. Objectives

•Perform exploratory data analysis (EDA) to understand the distribution, patterns, and relationships within the order book data, and visualize key insights using descriptive statistics and data visualization techniques.

•Preprocess the data by handling outliers, missing values, and standardizing features, while also exploring feature engineering methods to enhance predictive power and ensure data quality.

•Develop and evaluate machine learning models to predict the probability of future price changes, using appropriate algorithms and hyperparameter tuning, while validating the effectiveness of preprocessing techniques through cross-validation and performance metrics.

VI. Existing Methodology

Lee and Mykland developed the LM estimator, a tool in financial econometrics for estimating integrated volatility of asset prices. It's particularly useful in noisy market conditions. Here's a simplified overview:

1. Input:
- High-frequency price data (e.g., stock prices).
- Sampling frequency (e.g., tick-by-tick data).

2. Preprocessing:
- Remove outliers.
- Ensure evenly sampled data

3.Initialization:
- Set parameters like observation count and time window length.
- Decide on handling irregularities like price jumps.

4. Estimation Algorithm:
- Calculate realized volatility for each observation.
- Smooth volatility using a kernel function to reduce noise.
- Aggregate smoothed estimates over a time window.

5. Output:
- Integrated volatility estimates for each time period.

The LM estimator involves complex math, but these steps give a basic understanding of how it works.

VII. Proposed enhancements / novelty

VIII. DATASET

A. About The Data

The raw dataset used in this paper can be accessed at: **https://www.kaggle.com/competitions/caltech-cs155-2020**

The dataset under consideration has a distinct problem, as each row represents a single point in time and the condition of the order book at that particular 5ms interval. Creating a predictive framework that can precisely predict the direction of the "mid" price—a critical statistic that denotes the middle point between the best bid and ask prices—is the main objective. In particular, each timestep must be categorized as either a possible increasing trajectory (designated as "1") or a static or falling trajectory (designated as "0").

B. Terminologies in Dataset

•**id** - The timestep ID of the order book features.

last_price - the price at which the most recent order fill occurred.

•**mid** - the "mid" price, which is the average of the best bid (bid1) and the best ask (ask1) prices.

•**opened_position_qty** - In the past 500ms, how many buy orders were filled?

•**closed_position_qty** - In the past 500ms, how many sell orders were filled?

•**transacted_qty** - In the past 500ms, how many buy+sell orders were filled?

•**d_open_interest** - In the past 500ms, what is (#buy orders filled)- (#sell orders filled)?

•**bid1** - What is the 1st bid price (the best/highest one)?

•**bid[2,3,4,5]** - What is the [2nd, 3rd, 4th, 5th] best/highest bid price?

•**ask1** - What is the 1st ask price (the best/lowest/cheapest one)?

•**ask[2,3,4,5]** - What is the [2nd, 3rd, 4th, 5th] best/lowest/cheapest ask price?

•**bid1vol** - What is the quantity of contracts in the order book at the 1st bid price (the best/highest one)?

•**bid[2,3,4,5]vol** - What is the quantity of contracts in the order book at the [2,3,4,5]th bid price (the [2,3,4,5]th best/highest one)?

•**ask1vol** - What is the quantity of contracts in the order book at the 1st ask price (the best/lowest/cheapest one)?

•**ask[2,3,4,5]vol** - What is the quantity of contracts in the order book at the [2,3,4,5]th ask price (the [2,3,4,5]th best/lowest/cheapest one)?

•**y** (unique to training data) - What is the change in the mid price from now to 2 timesteps (approx. 1 second) in the future? "1" means this change is strictly positive, and "0" means the change is 0 or negative.

IX. Methodology

A. Roadmap

To guarantee the quality and usefulness of the data for analysis, a thorough data pretreatment pipeline is necessary before starting the modeling step. This complex procedure includes feature engineering, data cleansing, addressing missing values, and possible scaling or transformations to improve the features' predictive power. Furthermore, correcting any imbalances in the target variable and investigating resampling strategies would be required to lessen biases and enhance model functionality. The study will use a range of machine learning algorithms to train and assess predictive models once the data has undergone thorough preprocessing. During this iterative process, essential metrics including precision, recall, F1-score, and accuracy are carefully chosen, hyperparameters are adjusted, and model performance is rigorously assessed. The final objective is to create a strong predictive framework that, using the complex patterns seen in the order book data, can reliably predict future price changes.

B. Data Preprocessing

1) Exploratory Data Analysis (EDA): We started by calculating central tendency measures like mean, median, and mode to understand the core values of our features. To gauge data spread, we computed variance and standard deviation. Visual aids such as histograms and box plots helped us grasp the data's shape and identify potential outliers. We used Hexbin plots which are merged scatter plot and histogram principles to visualize bivariate data distributions in the dataset . Density contour plots offered another view by outlining dense regions with contour lines. Heatmaps displayed feature correlations visually , while we used violin plots depicted density profiles, emphasizing areas with more data points and skewness of each features in the dataset.

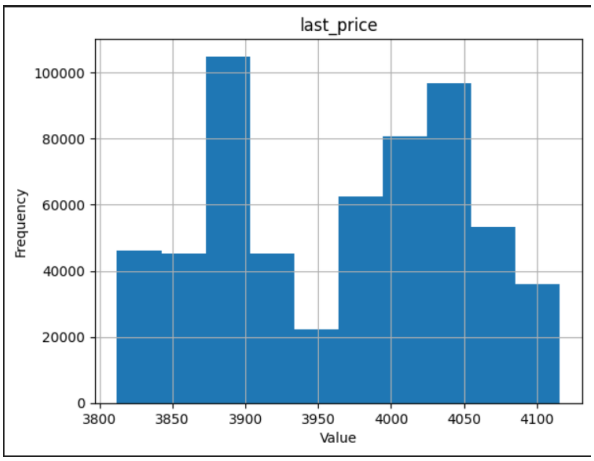

Fig. 1: Spread

2) Outlier Detection and Handli: From the previous eda we had got idea that data is highly skewed and there were lot of oultiers and missing values in the data.For detecting the outliers we use IQR method from the box plot analysis .

- IQR=Q3-Q1
- Lower bound: Q1 - 1.5 * IQR
- Upper bound: Q3 + 1.5 * IQR
    - where-
    - IQR-Inter Quartile Range
    - Q1-25th percentile
    - Q3-75th percentile

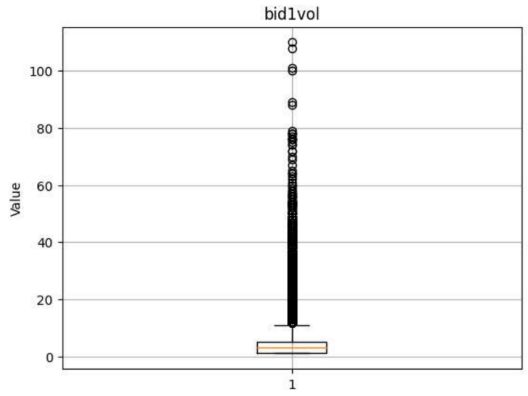

Fig. 2: Outlier detection using Box Plot

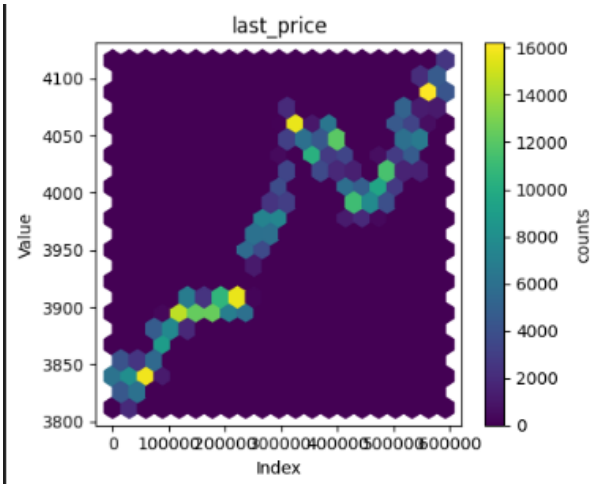

Fig. 3: Hexabin Plot

After identifying the outliers are Initially approach was trimming outliers and this was fatal as the significant amount of data was lost in this process. Instead,adopted a robust approach, capping outliers by the replacing it with the values at Q3 from , preserving most data while minimizing the impact of extreme values.

3) Feature Construction: We introduced a new feature, the "Liquidity Indicator," which Utilize bid-ask spread as a feature, calculated as the difference between the best bid and ask prices and Incorporate volume at the best bid and ask prices as additional liquidity indicators and visually examined its distribution. By capturing nuanced aspects like liquidity it improves the model's ability to forecast price changes accurately

4) Feature Scaling: Initially, we explored robust scaling methods like RobustScaler but as we already had handled the outlier and based on the techniques of imputation we are going to use in this paper we ultimately used StandardScaler for standardization of all the features before missing value imputation.

```
In [ ]: mad =(df-df.mean()).abs()
        print("Mean Absolute Deviation (MAD):\n", mad.mean())
```

```
Mean Absolute Deviation (MAD):
 last_price              32.544473
mid                     32.542544
opened_position_qty      1.336979
closed_position_qty      1.521899
transacted_qty           2.411397
d_open_interest          1.657635
bid1                    32.543140
bid2                    32.543774
bid3                    32.544062
bid4                    32.544710
bid5                    32.545165
ask1                    32.542233
ask2                    32.540981
ask3                    32.539835
ask4                    32.538554
ask5                    32.537638
bid1vol                  2.987776
bid2vol                  3.673262
bid3vol                  3.874921
bid4vol                  4.141199
bid5vol                  4.464485
ask1vol                  3.125307
ask2vol                  3.888302
```

Fig. 4: Conducting EDA on the features

```
for col in X.columns:
    print( col,"      ",X[col].skew())
```

```
last_price       -0.1318266914887565
mid       -0.13181556203389655
opened_position_qty        1.3158249657575323
closed_position_qty       0.4945496630812634
transacted_qty       1.190513498903171 8
d_open_interest       -0.09549288011632044
bid1       -0.1319289874260657
bid2       -0.1320183498414334 3
bid3       -0.1320958672592881 8
bid4       -0.1321834293872104 8
bid5       -0.1322734302838220 7
ask1       -0.1317004558218221 3
ask2       -0.1316105917262429 5
ask3       -0.1315290539612179
ask4       -0.1314498362297554 8
ask5       -0.1313743815795252
bid1vol       1.3130071058410309
bid2vol       1.013611199828579
bid3vol       1.1575818931885964
bid4vol       1.0708423658650026
bid5vol       0.9884410725149758
ask1vol       1.302075593470738
```

Fig. 5: Skewness Reduction

5) Missing Value Imputation:: Initially Techniques such as deletion and feature disregarding didn't yield satisfactory results as expected and also lead to significant data lose and while in the eda part we had studied the correlation between the features and found out that the feature containing the missing value had a weak positive correlation. After we come to imputing the missing value instead.Various methods which were used for imputations are Mean/Median/Mode Imputation,Regression Imputation,K-Nearest Neighbors (KNN) Imputation.With the knn imputation we could get the most satisfactory results and the total dispersion of the features where maintained.Hence, we implemented the K-Nearest Neighbors (KNN) algorithm for missing value imputation, even in knn we tried both distance based and uniform based averaging for the imputing values,theoretically better results should be found in distance based averaging and so was when implemented

practically .

6) Advanced Exploratory Data Analysis: Before the final model development Advanced Exploratory Data Analysis (EDA) techniques can provide deeper insights into the data, uncover complex relationships, and aid in the selection and engineering of relevant features for modeling. Here are some advanced EDA methods:

Correlation and Covariance Analysis:

- Compute correlation matrices to identify linear relationships between pairs of features.
- Visualize correlation matrices using heatmaps or similar techniques for easy interpretation.
- Analyze covariance matrices to understand the spread and direction of feature relationships.
- Identify and potentially remove highly correlated or redundant features to avoid multicollinearity issues.

Principal Component Analysis (PCA) is a powerful dimensionality reduction technique that we used ig. It aims to transform a high-dimensional dataset into a lower-dimensional subspace while retaining as much of the original information or variance as possible. Here's a more detailed explanation of PCA and its implementation:

1) Motivation and Rationale:
   - High-dimensional datasets often contain redundant or highly correlated features, leading to the curse of dimensionality and computational inefficiencies.
   - PCA identifies the directions of maximum variance in the data, known as principal components, and projects the data onto a lower-dimensional subspace spanned by these components.
   - This projection captures the most important patterns and structures in the data while reducing noise and redundancy.
2) Mathematical Formulation:
   - Let X be an n x p data matrix, where n is the number of observations and p is the number of features.
   - PCA seeks to find a linear transformation that maps the original p-dimensional data onto a k-dimensional subspace (k < p) while minimizing the reconstruction error.
   - The principal components are the orthogonal directions (eigenvectors) of the covariance or correlation matrix of X, corresponding to the largest eigenvalues.
3) Implementation Steps:
   a. Data Preprocessing: Center the data by subtracting the mean from each feature, or standardize (scale) the data if features have different units or scales.

   b. Compute the Covariance or Correlation Matrix: Calculate the covariance or correlation matrix of the preprocessed data.

   c. Eigendecomposition: Perform eigendecomposition on the covariance or correlation matrix to obtain the eigenvectors and corresponding eigenvalues.

   d. Select Principal Components: Sort the eigenvectors by their corresponding eigenvalues in descending order. The top k eigenvectors with the largest eigenvalues are chosen as the principal components.

   e. Project Data onto Principal Components: Project the original data onto the k-dimensional subspace spanned by the selected principal components by multiplying the data matrix X with the matrix of k principal component vectors.
4) Determining the Number of Principal Components:
   - The choice of k (the number of principal components to retain) is crucial and depends on the desired trade-off between dimensionality reduction and information preservation.
   - Common approaches include:
     - Explained Variance Ratio: Retain enough components to account for a certain percentage (e.g., 90% or 95%) of the total variance in the data.
     - Scree Plot: Plot the eigenvalues in descending order and look for an "elbow" or sudden flattening, indicating that subsequent components contribute little to the overall variance.
     - Domain Knowledge: Choose k based on prior knowledge or interpretability requirements of the specific problem domain.

C. Model Development and Evaluation:

Model Selection: We extensively evaluated two different machine learning algorithms to choose the best approach for forecasting 1-second price changes in futures contracts.

Random Forest: We opted for Random Forest due to its ability to handle high-dimensional data and capture complex non-linear relationships. Random Forest is an ensemble of decision trees. It uses bootstrap sampling and random feature subsets to build diverse trees. For prediction, it aggregates

the outputs of all trees: majority vote for classification. The ensemble reduces variance and overfitting. Hyperparameters like maxdepth and nestimators were fine-tuned using techniques like GridSearchCV.
C(x) = majority vote(C1(x), C2(x), ..., CB(x)) Where C(x) is the predicted class, f(x) is the predicted value, Cb(x) and fb(x) are predictions from the b-th tree, and B is the total number of trees.

Gradient Boosting: Models like CatBoost were explored for their proficiency in capturing intricate patterns in HFT data. Hyperparameters such as learningrate and maxdepth were optimized through randomized search

Model Evaluation: Stratified k-fold cross-validation assessed model generalization. We computed accuracy, precision, recall, and F1-score to measure predictive capabilities.

Hyperparameter Tuning: For Random Forest, Logistic Regression Grid/RandomizedSearchCV evaluated hyperparameter combinations systematically. For gradient boosting models, randomized search strategies maximized validation metrics.

X. Results Obtained

The results obtained in both the techniques are discussed below and the performance metrics are also specified

Models	Accuracy
Logistic Regression	0.663
Random Forest Classifier	0.667
CatBoost	0.671

REFERENCES

[1] Jonathan Brogaard, Allen Carrion, Thibaut Moyaert, Ryan Riordan, Andriy Shkilko, Konstantin Sokolov. (2018). High frequency trading and extreme price movements.

[2] A. Lakshmi and Sailaja Vedala. (2017). A study on low latency trading in Indian stock markets. *International Journal of Civil Engineering and Technology*, **8**(12), 733-743.

[3] Peter Gomber, Björn Arndt, Marco Lutat, and Tim Uhle. (2011). High-Frequency Trading. *SSRN Electronic Journal*. doi: 10.2139/ssrn.1858626.

[4] C. Dutta, K. Karpman, S. Basu, et al. (2023). Review of Statistical Approaches for Modeling High-Frequency Trading Data. *Sankhya B*, 85(Suppl 1), 1–48.

[5] G.P.M. Virgilio. (2019). High-frequency trading: a literature review. *Financ Mark Portf Manag*, 33, 183–208. doi: 10.1007/s11408-019-00331-6.

APPENDIX

Add the first page of Plagiarism Report here, after I provide the report to you (must have less than 15% similarity). Each team member should add their signature on the report page

project_report .pdf

ORIGINALITY REPORT

7%	4%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Aston University Student Paper	1%
2	Submitted to Imperial College of Science, Technology and Medicine Student Paper	1%
3	www.dellemc.com Internet Source	1%
4	docs.oracle.com Internet Source	1%
5	Submitted to CSU, San Jose State University Student Paper	1%
6	Submitted to UNITEC Institute of Technology Student Paper	1%
7	docplayer.net Internet Source	<1%
8	www.i-scholar.in Internet Source	<1%