CFM Data Challenge

Stock Classification from High Frequency Market Data

Sofiane Ezzehi Bastien Le Chenadec

École Normale Supérieure Paris-Saclay, Master MVA

March 20th, 2024

Table of contents

Introduction

Introduction

- **Objective:** Classify stocks based on snapshots of their respective order books.
- **Data:** Each sample is a 100-event sequence of the posted **passive** and **aggressive** orders in the order book.
- A lot of usual features are **missing**, and others are **hidden**. *Example : The Price, best bid and best ask are centered around the first event.*

	Stocks	**Events**	**Train samples**	**Test samples**
Number	24	100	160,800	80,600

Data Overview

Feature	Type	Description
Venue	Categorical	The venue where the order was placed.
Order id	Integer	A unique identifier, which can be used to retrace updates to the order.
Action	Categorical	The type of action (new, delete, update).
Side	Categorical	The side of the order (buy, sell).
Price	Float	The price of the order.
Bid	Float	The best buying price for the stock.
Ask	Float	The best selling price for the stock.
Bid size	Float	The number of shares available at the best buying price.
Ask size	Float	The number of shares available at the best selling price.
Trade	Categorical	Whether a trade occured or not.
Flux	Integer	The quantity of shares for this order.

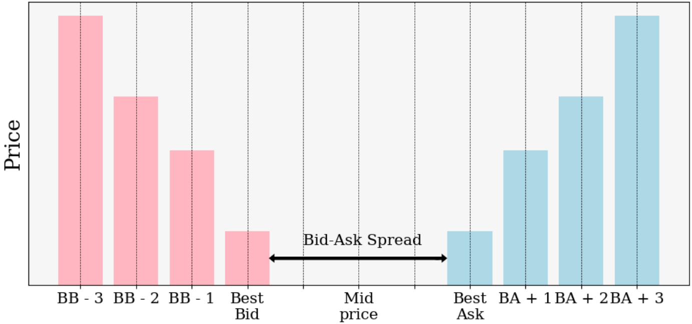
Data Visualization and feature engineering

Data Visualization and feature engineering

- **Objective:** Understand the data and find sufficiently stock-discriminative features.
- **Approach:** Visualize the data and engineer features.
- **Evaluation:** Use **feature importance** from **Random Forest classifiers** to select the most discriminative features.

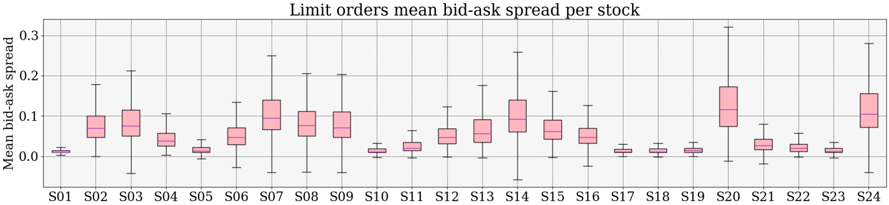
Feature 1 : Bid-Ask Spread

- **Bid-Ask Spread:** Natural feature to consider in a financial context.
- **Intepretation:** Measure of the **liquidity** and **volatility** of the stock.
  - More **liquid** stocks have a **smaller** spread.
  - More **volatile** stocks have a **larger** spread.



Feature 1 : Bid-Ask Spread

- For each stock and corresponding observations, we plot the mean spread over the 100 events.



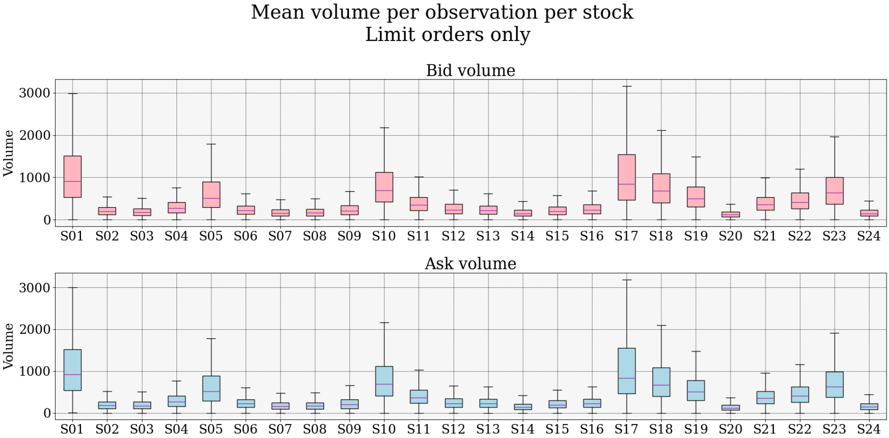
→ The spread is a **good indicator** of the stock, with a decent variety of boxplot shapes.

→ Most stocks are **fairly liquid**, with an average spread of less than 10 ticks.

→ Some stocks are strikingly **more volatile** than others.

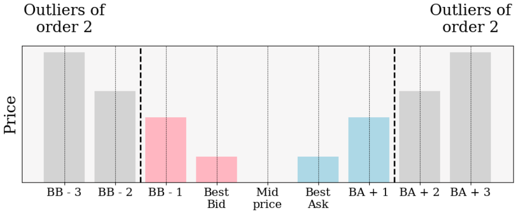
Feature 2 : Best Bid and Ask volumes

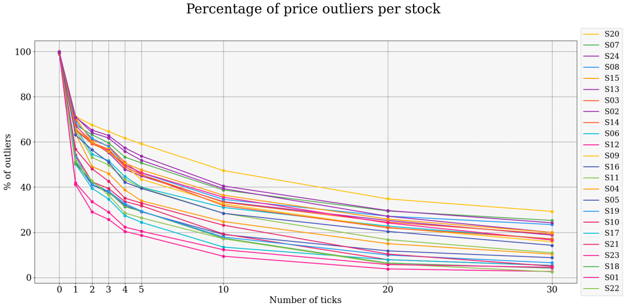
- **Best Bid and Ask volumes:** Another natural measure of **liquidity**.



Feature 3 : Price outliers (number and price value)

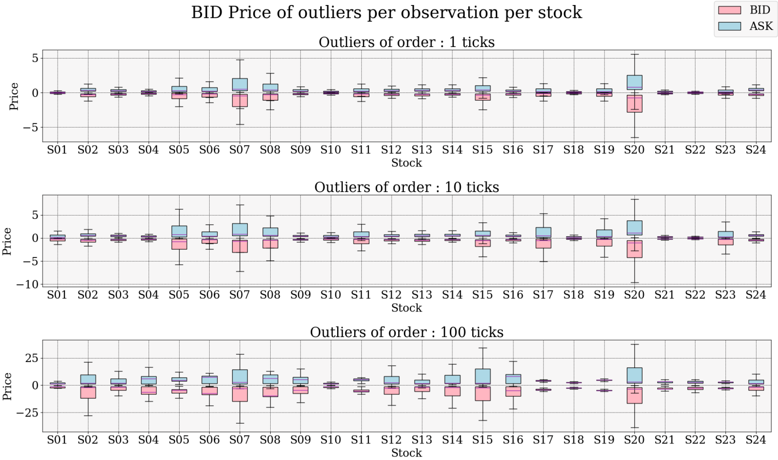
- **Definition:** A price outlier of order *i* is a price that is more than *i* ticks away from the best bid or ask.





Feature 3 : Price outliers (number and price value)

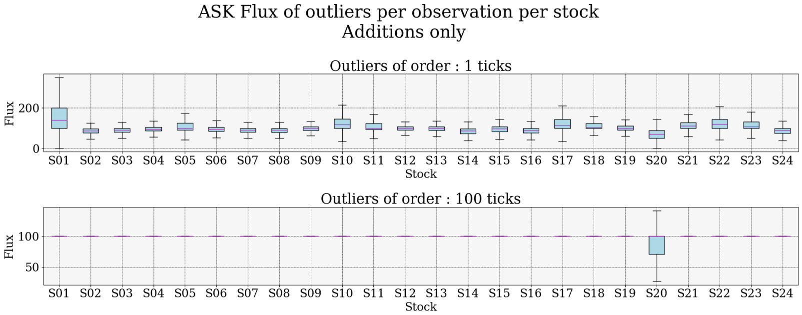
We plot for each stock and corresponding observations, the mean price value of the outliers over the 100 events.



Some stocks stand out well.

Feature 4 : Price outliers (flow)

- We plot for each stock and corresponding observations, the mean flow value of the outliers over the 100 events.
- Analysis on 4 subsets of the data: ask addition, ask update, bid update and bid addition.



Random Forest Classifier and Feature Importance Analysis

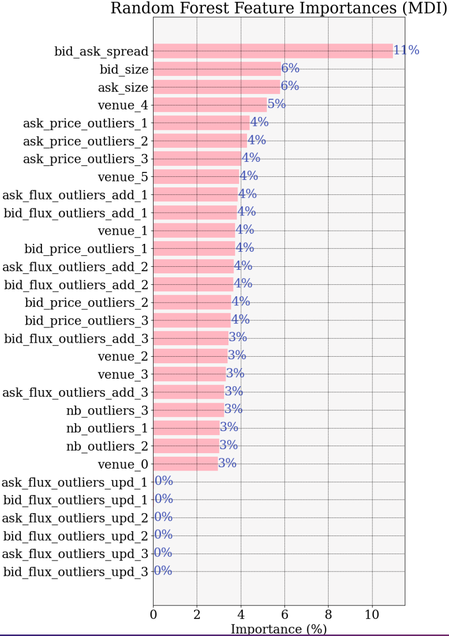
Random Forest Classifier

A first approach to the classification task is to use a **Random Forest Classifier** using the hand-crafted features.

2 **objectives:**

- **Feature importance:** Use the feature importance to unravel the most discriminative features.
- **Performance:** We tested 2 different models on different sets of features:
  1. **Model 1's features:** the Bid-Ask spread, the Bid and Ask volume, the number of price outliers, the price outliers, the flux of the price outliers, and the proportion of each venues on which the orders were placed.
  2. **Model 2's features:** Only a subset of the features of Model 1: the Bid-Ask spread, the Bid and Ask volume and the venue proportions.

Feature importance



Performance of the 2 random forest models

Model	Validation accuracy	Test accuracy
Model 1	49%	22%
Model 2	28%	20%

→ **Conclusion:** Clear indication that the "outlier" features, while being very discriminative on the training set, are almost irrelevant on the test set.

Feature-based approach

Franck Zibi's model

Model : Inspired by the winning solution of last year's challenge.

1. A random forest classifier is trained on the training set. Outputs a probability distribution over the classes.
2. Predict the residuals of the random forest classifier for each class using three models (Ridge regressor, k-nearest neighbors regressor, linear regressor).
3. Stack the three models using a linear regressor.
4. Predict the class of a sample by adding the output of the random forest classifier to the output of the stacked models, and taking the class with the highest probability.

Used features

Used features for the classification task:

1. Compute min, max, mean, median, and standard deviation over the 100 events : for each of the 11 original features, as well as the bid-ask spread, the limit order indicator, and the sum of bid and ask sizes.
2. Add the features we previously engineered:
   - The Bid-ask spread
   - The volume of the orders (bid and ask size)
   - The number of price outliers
   - The average price of the outliers
   - The average flux of the price outliers for the different types of outliers (ask addition, ask update, bid update)
   - The venue proportions

Data normalization

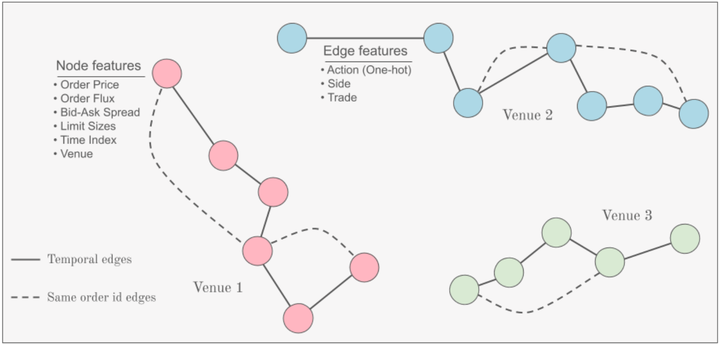
1. Approach based on features engineered with market intuition ✓
2. More generic approach :

- **Outliers removal** : Remove observations with at least one value at more than 7 standard deviations from the mean. Removed around 3.6% of the training set.
- **Log transformation** : Applied to the flux, bid size and ask size. Preserves the sign of the values.
- **Min-max scaling** : Further normalization of the bid size and ask size features.

Graph modelisation approach

Graph representation

- **Main idea:** Represent the data as a graph to capture the temporal dependencies between the events, as well as to give a structural representation of the data.



Graph Attention Network (GAT)

G : undirected graph with N nodes, node features $h_1, \ldots, h_N \in \mathbb{R}^{d_1}$, edge features $\{e_{i,j} \,|1 \leq i, j \leq N\} \in \mathbb{R}^{d_2}$

Attention weights

$$w(h_i, h_j, e_{i,j}) = a^T \, \text{LeakyReLU}(W_1 h_i + W_1 h_j + W_2 e_{i,j})$$

$$\alpha_{ij} = \frac{\exp(w(h_i, h_j, e_{i,j}))}{\sum_{k \in \mathcal{N}_i} \exp(w(h_i, h_k, e_{i,k}))}$$

with $\mathcal{N}_i$ the set of neighbors of node i, and W_1, W_2 and a learnable parameters.

Embedding update

$$h_i' = \text{LeakyReLU}\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k)} W_1^{(k)} h_j\right)$$

where K is the number of attention heads.

Recurrent Neural Networks approach

LSTM (Long-Short Term Memory)

- Categorical features embedding
- MLP on the last hidden state
- Bidirectional

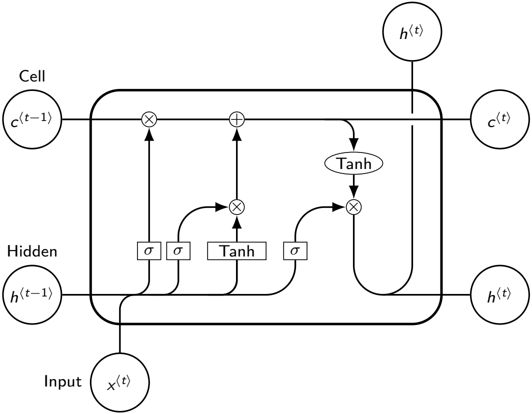


Figure: LSTM cell diagram

Training of Graph models and RNNs

Loss

- Cross-entropy loss
- Minimum class confusion (MCC) loss on the test set

MCC

Batch of samples $(X_n)_{1 \leq n \leq N}$ and model predictions $\hat{Y}_n = F(X_n) \in \mathbb{R}^{24}$:

1. $$\tilde{Y}_{n,i} = \frac{\exp\left(\hat{Y}_{n,i}/T\right)}{\sum_{j=1}^{24} \exp\left(\hat{Y}_{n,j}/T\right)}$$

2. $$H_n = -\sum_{i=1}^{24} \tilde{Y}_{n,i} \log\left(\tilde{Y}_{n,i}\right)$$

3. $$W_n = N \times \frac{1 + \exp(-H_n)}{\sum_{i=1}^{N} 1 + \exp(-H_i)}$$

4. $$C_{i,j} = \tilde{Y}_{\cdot,i}^{T} \operatorname{diag}(W_1, \ldots, W_B) \tilde{Y}_{\cdot,j}$$

Model calibration

- Calibration adjusts model output to match true probabilities.
- Calibration performed on validation data to avoid bias.
- Isotonic regression chosen for simplicity and effectiveness.
- Fits piecewise constant non-decreasing function to data.

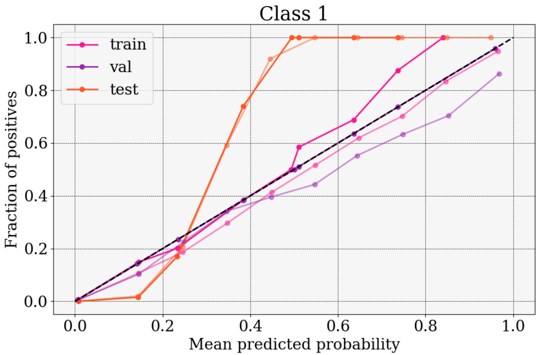
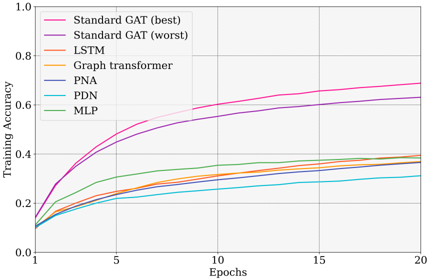


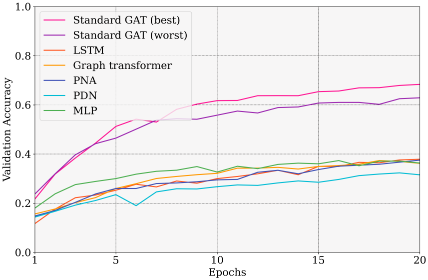
Figure: Calibration of the first class of the GAT model.

Training curves

- Training with Adam, learning rate $5 \cdot 10^{-3}$, multiplicative learning rate scheduler ($\times 0.95$ per epoch).
- Training on RTX4060 with batch size 256.



(a) Training accuracy



(b) Validation accuracy

Results

Ensemble Results

Model description	Training acc.	Validation acc.	Test acc.
Random Forest I	0.31	0.28	0.19
Random Forest II	0.50	0.48	0.22
PDN	0.42	0.43	0.23
MLP	0.42	0.41	0.24
Franck Zibi's model	0.95	0.42	0.25
Graph transformer	0.44	0.43	0.29
General GNN	0.43	0.42	0.30
PNA	0.47	0.45	0.30
LSTM	0.57	0.44	0.30
GAT (50 epochs)	0.75	0.71	0.33
GAT (20 epochs)	0.72	0.68	0.34
Generalized GNN	0.73	0.71	0.35
Final ensemble	**0.83**	**0.81**	**0.40**

Table: Main accuracy results

Conclusion

Throughout this challenge, we did:

- Perform a thorough analysis of the data (visualizations, statistical tests).
- Engineer some features that were very discriminative on the training set.
- Train a wide variety of models to diversify our ensemble (RNNs, GNNs, statistical models, Franck Zibi's model).
- Experiment with different loss functions to make the models more robust to the change of distribution between the training and the test set.
- Monitor the training of the models with multiple metrics.
- Look into the predictions of the models to try to understand why they were so bad.

Discussion:

- We achieved over 80% accuracy on the validation set with an ensemble of models.
- We achieved a 40% accuracy on the test set.
- Main reason for the low accuracy: the change of distribution between the training and the test set.
- An interesting approach would be to train a model free of all the "outlier" features, but on a granular level (i.e. by deleting events and not whole samples).
- Ninth place on the public leaderboard and a second place on the academic leaderboard.
- Some students were able to reach as high as 60% accuracy on the test set. We would be very interested in understanding how they achieved this.