Deep Limit Order Book (DeepLOB) Forecasting Applied to Crypto Currencies

Remi Peyrot[a]

[a]*Department of Computer Science, Monroe College, NY, United State*

Abstract

The paper delves into the application of deep learning for predicting cryptocurrency price movements using Limit Order Book (LOB) data. By examining five cryptocurrency pairs across the liquidity spectrum on Binance, the study explores the impact of liquidity variations on forecasting efficacy. A variety of deep learning models, including DeepLOB and Transformers, are trained and evaluated across three prediction horizons (5, 10, and 50 LOB updates) using LOB snapshots sampled at intervals as short as 100 milliseconds. Model performances are assessed through traditional metrics (accuracy, F1-score, and Matthews Correlation Coefficient) and a trading-oriented metric that measures the probability of executing correct transactions [8]. Results demonstrate a direct relationship between liquidity and predictive power, where higher liquidity pairs yield more reliable signals across horizons, supporting effective high-frequency trading. The findings underscore the importance of considering asset-specific microstructural characteristics in model selection and strategy design for cryptocurrency markets.

Keywords: Cryptocurrency, Limit Order Book, High-Frequency Trading, Deep Learning, Market Microstructure.

1. Introduction

Cryptocurrency markets, like their traditional counterparts, operate in complex, dynamic environments where the high volume of transactions and diverse participant strategies contribute to their stochastic nature. These markets are characterized by a low signal-to-noise ratio, as traders act at multiple time scales and possess varying levels of access to information and trading expertise. To efficiently manage this complexity, exchanges rely on electronic Limit Order Books (LOBs) [1], which record and organize all outstanding buy and sell orders. Unlike traditional quote-driven markets, the LOB allows participants to directly observe the evolving supply and demand dynamics at different price levels. This transparency, combined with FIFO (first-in, first-out) [2] execution priority, forms the backbone of modern cryptocurrency trading systems.

The structure and dynamics of the LOB present a high-dimensional and temporally evolving problem, where interactions between prices, order sizes, and execution rates must be modeled effectively. Traditional methods, such as autoregressive models [3, 4], often struggle to capture these complexities due to the vast scale and intricacies of the data. Recent advancements in machine learning, particularly deep learning, have shown promise in overcoming these limitations by leveraging LOB data to forecast mid-price movements or other market signals [5, 9, 13]. High-frequency trading (HFT) strategies, which operate on similar data at extremely high speed, further complicate the environment by exploiting minor inefficiencies not yet identified by other actors, adding noise to the system, and introducing asymmetries in the information available to traders [10].

Despite significant research in applying deep learning to LOB data, the computational cost of processing high-frequency data and latency of deep learning models at inference time, coupled with the unique microstructural properties of cryptocurrency markets [12], limits the transition of these techniques from research to production in the industry. Nonetheless, the study of LOB microstructure continues to reveal valuable insights into liquidity patterns, transaction costs, and price formation processes, which may vary across different assets even within the same asset class [8, 11].

The structure of this paper is inspired from Briola et al. [8]: Section 2 lays out the technical background required to comprehend the mechanics of the Limit Order Book (LOB). In Section 3, we provide a comprehensive description of the dataset employed in our experiments. Section 4 delves into the technical framework used to train and validate the different models we used. Section 5 focuses on the microstructural characteristics of the cryptocurrency pairs examined in our study, while Section 6 presents the outcomes of our forecasting experiments. Lastly, Section 7 wraps up the paper by providing a cohesive view of deep learning approaches informed by market microstructure for LOB forecasting and highlights the remaining challenges in this field.

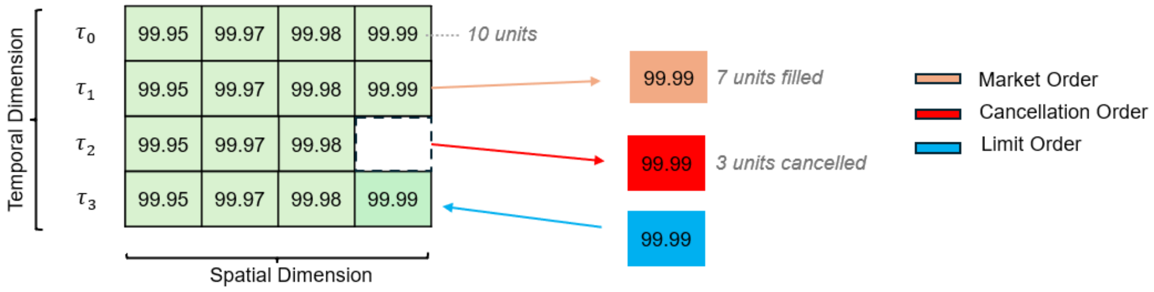
2. Limit Order Book

Modern exchanges utilize an electronic system to log and match the trading intentions of market participants, structured through what is known as the Limit Order Book (LOB). Each asset has a dedicated LOB that provides traders with a real-time view of the visible supply and demand in the market. This system enables price formation through a decentralized process driven by the submission, modification, and cancellation of orders, allowing participants to act on price signals as they evolve over time.

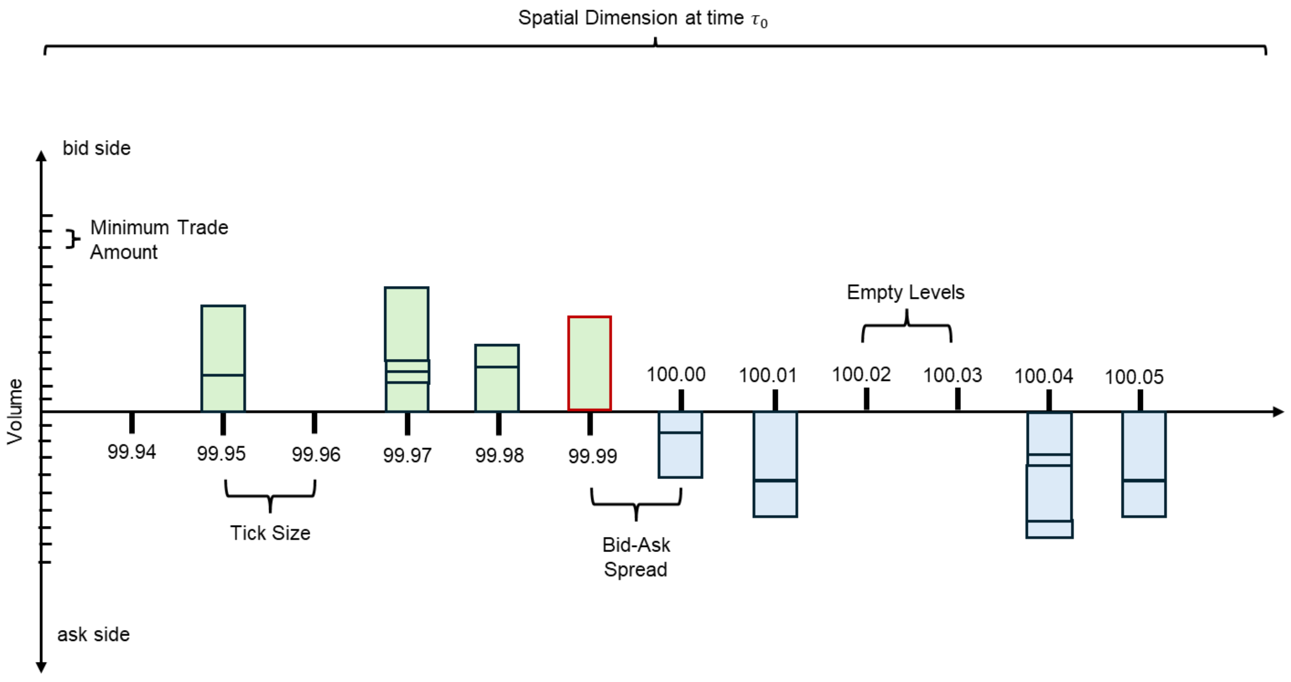
An order is essentially a public declaration of a trader's intent to buy or sell a specific quantity of an asset at a certain price. Formally, an order can be represented as a tuple $(\epsilon_0, p_0, v_0, \tau_0)$, where ϵ_0 denotes the order direction (e.g., $\varepsilon = +1$ for buy orders and $\varepsilon = -1$ for sell orders), p_0 indicates the price level, v_0 the volume, and τ_0 the submission time [14]. Price levels in an LOB are discrete, determined by the tick size (θ), while volume levels are also discrete, specified by the lot size. In equity markets, price levels in an LOB are discrete, typically denominated in dollars, with uniform tick sizes (e.g., $0.01 on Nasdaq) and standardized lot sizes across securities. However, in currency markets, including cryptocurrency, each trading pair may have its own base currency (e.g., BTC/USD or BTC/USDT), and the tick and lot sizes are therefore specific to each pair and exchange.

Orders can be classified into three main types: (i) limit orders, (ii) market orders, and (iii) cancellation orders. Limit orders specify a price different from the current best price, providing liquidity but without guaranteed execution. Market orders seek immediate execution at the best available price, typically incurring higher transaction costs due to their liquidity-taking nature. Lastly, cancellation orders enable traders to withdraw or reduce limit orders without transaction costs.

Figure 1: Illustration of the Limit Order Book (LOB). The top section depicts the dynamic evolution of LOB price levels in response to market orders from the opposite side, and cancellation and limit orders on the same side. The bottom section provides a static snapshot of the LOB ($\mathbb{L}(\tau_0)$), showing both price levels and corresponding volumes.



τ_0	99.95	99.97	99.98	99.99
τ_1	99.95	99.97	99.98	99.99
τ_2	99.95	99.97	99.98	
τ_3	99.95	99.97	99.98	99.99



The LOB is thus a repository of unmatched (active) limit orders for each asset at time τ and can be represented as a multivariate time series $\mathbb{L}$, $\mathbb{L}(\tau) \in \mathbb{R}^{4L}$, capturing price and volume information across *L* levels for both the bid and ask sides. Each record *L(τ)* consists of price-volume pairs of the form $\{\boldsymbol{P}^s(\tau), \boldsymbol{V}^s(\tau)\}_{s\in\{ask,bid\}}$, where $\boldsymbol{P}^{ask}(\tau), \boldsymbol{P}^{bid}(\tau) \in \mathbb{R}^L$ are the prices on the ask and bid side, and $\boldsymbol{V}^{ask}(\tau), \boldsymbol{V}^{bid}(\tau) \in \mathbb{R}^L$ are the volumes on the ask and bid side, respectively:

$$\mathbb{L}(\tau) = \{p_\ell^{ask}(\tau), v_\ell^{ask}(\tau), p_\ell^{bid}(\tau), v_\ell^{bid}(\tau)\}_{\ell=1}^{L} \tag{1}$$

It means that, $\forall\ \tau \in \{1, \dots, N\}\ and\ \forall\ \ell \in \{1, \dots, L\}$ on the *s* side, $v_\ell^s(\tau)$ shares can be traded at price $p_\ell^s(\tau)$.

The mid-price $m(\tau)$of the stock at time τ is defined as the average of the best ask price $p_1^{ask}(\tau)$ and the best bid price $p_1^{bid}(\tau)$:

$$m(\tau) = \frac{p_1^{ask}(\tau) + p_1^{bid}(\tau)}{2}.$$

The bid-ask spread $\sigma(\tau)$ at time τ is the difference between the best ask and bid prices:

$$\sigma(\tau) = p_1^{ask}(\tau) - p_1^{bid}(\tau).$$

3. Data

In this study, we analyze five cryptocurrency pairs — BTC/USDT, ETH/USDT, SOL/USDT, DOGE/USDT, and ACM/USDT — all traded on the Binance exchange. Each pair is quoted in USDT (Tether), a stablecoin pegged to the U.S. dollar, providing the stability of a fiat equivalent within the cryptocurrency market. Pairing against USDT, rather than USD, offers traders and investors the advantage of executing trades entirely within the crypto ecosystem, which reduces transfer fees and speeds up transaction times by avoiding external fiat transactions. USDT's blockchain-based infrastructure also provides transparency and security, making it a reliable medium of exchange in high-frequency crypto trading.

These pairs cover a wide range of liquidity, from BTC/USDT, the most liquid pair, to ACM/USDT, representing a lower-liquidity, speculative asset. This selection provides a small but robust sample that includes both high- and low-capitalization assets, with BTC and ETH as large-cap "blue-chip" cryptocurrencies, while SOL and DOGE would represent mid-cap equivalents, highly traded altcoins. DOGE, sometimes classified as a "meme" coin, along with SHIBA, sees its trading activity heavily influenced by the popularity of related memes circulating on social media platforms like Reddit and X. Unlike assets driven by fundamental factors, meme coins often experience trading surges tied to viral content or endorsements, with DOGE notably spiking in response to high-profile posts from figures like Elon Musk. ACM, a fan token often classified as a so-called "junk coin" due to its low liquidity and limited market depth, illustrates the volatility and price sensitivity typical of niche cryptocurrencies. Concentrated on specialized platforms and driven by AC Milan fan engagement, ACM experiences sharp demand fluctuations tied to club events, making it more susceptible to price discrepancies and swings than mainstream assets.

The data for this analysis was sourced from the data provider called Crypto Lake [15], a comprehensive data provider specializing in high-frequency cryptocurrency market data. Crypto Lake offers in-depth, real-time insights into market dynamics across a variety of cryptocurrency exchanges. It provides access to order book data, trades, and aggregated market indicators, serving as a valuable resource for quantitative analysis and trading model development.

For this study, we utilized data from the "book" table of Crypto Lake, which captures detailed snapshots of market depth with updates at intervals as frequent as every 100 milliseconds, depending on exchange support. Each snapshot includes up to 10 price levels on both the bid and ask sides, allowing us to examine fine-grained trading patterns and liquidity distribution. This level of granularity facilitates a precise understanding of market behaviors, particularly useful in assessing volatility, order flow, and liquidity across a diverse set of crypto assets.

Table 1: Overview of the average market capitalizations and average daily trading volumes on Binance for 2024 are as follows:

Ticker	Crypto Name	Market Capitalization (2024)	Adv on Binance (quoted ccy)
BTC	Bitcoin	\$1.2 T	0.2 M
ETH	Ethereum	\$300 B	2.5 M
SOL	Solana	\$80 B	35 M
DOGE	Dogecoin	\$15 B	3 M
ACM	AC Milan Fan Token	\$10 M	3 M

Tables 2 and 3 present the minimum order sizes and the smallest price increment (tick size) for each pair, both expressed in USDT. Unlike equity markets, where trades are denominated in whole shares with standardized minimum trading increments, cryptocurrency pairs allow flexibility in the minimum trading amount for both the base and quote currencies. In this case, the base currency (USDT) has a minimum trading size of 1 USDT across all pairs, while the minimum tradable amount of each cryptocurrency can be a fractional unit of the asset.

This flexibility highlights the distinct liquidity structure in crypto markets, where assets range from highly liquid pairs (such as BTC/USDT) to more niche and speculative low-liquidity assets (such as ACM/USDT). Analyzing this selection of pairs provides a broad view of the crypto liquidity landscape, from stable, high-liquidity assets to highly volatile, illiquid assets, thereby enhancing the robustness and predictive power of models across the liquidity spectrum.

Table 2: Tick-size on Binance

	BTC	ETH	SOL	DOGE	ACM
Θ (USDT)	0.01	0.01	0.01	1e-5	0.001

Table 3: Minimum Order Size on Binance

	BTC	ETH	SOL	DOGE	ACM
Min Trade Amount	1e-5	1e-4	0.001	1	1
Max Limit Order Amount	9e3	9e3	9e4	1e6	9e4

In our experiments, the training, validation, and testing datasets were structured in sequential consecutive blocks to ensure no forward-looking bias in the model training—a critical requirement for time-series modelling. Consequently, for certain pairs, the rolling mean and standard deviation used in z-score normalization for the validation and test sets may extend into the training set. This approach is acceptable and follows standard practice for rolling estimators among practicians.

Table 4: Structure of the datasets used across training, validation, and testing stages. The sets include the specified start and end dates. Notably, weekends and public holidays are not excluded from any datasets, as trading occurs on these days in the crypto market.

	training						validation		test	
	H = 5		H = 10		H = 50					
pairs	from	to	from	to	from	to	from	to	from	to
BTC	9/18	9/29	9/20	9/29	9/4	9/29	9/30	10/2	10/3	10/5
ETH	9/18	9/27	9/18	9/27	8/27	9/27	9/28	10/1	10/2	10/5
SOL	4/29	5/27	5/5	5/27	3/1	5/27	5/28	6/5	6/6	6/15
DOGE	7/10	9/18	8/12	9/18	7/1	9/5	9/19	9/26	9/27	10/5
ACM	3/10	6/17	1/15	6/17	2023-09-14	6/17	6/18	8/10	8/11	10/5

Data collection was managed by downloading the order book snapshots from Crypto Lake through their Python API, populating a SQLite database hosted locally on my machine. Unlike equity markets, which restrict analysis to a subset of the trading day (e.g., 9:30 AM to 4 PM EST in the U.S.) to avoid the auction-led volatility at market open and close [14], cryptocurrency markets operate continuously. Thus, I retained all trading hours, including weekends and holidays, for each dataset, capturing uninterrupted, full-cycle market dynamics.

In this work, we aim to explore the predictability of mid-price direction changes at various time horizons, focusing on cases where these changes exceed a specified threshold (θ). For clarity, we will refer to these mid-price differences simply as "returns," though we are not referencing relative or logarithmic returns. By using the direct difference in mid-prices, we achieve greater control over the magnitude of changes across different time horizons, while also preserving the stationarity of the resulting time series, as per Briola et al. [8].

For this study, we consider three distinct horizons $H\Delta\tau \in \{5, 10, 50\}$ and the labeling process for each is defined as per Briola et al. [8] as follows:

$$\begin{cases} (m_{t+\Delta t} - m_t) \leq -\theta \rightarrow -1 \rightarrow Down\,, \\ -\theta < (m_{t+\Delta t} - m_t) < +\theta \rightarrow 0 \rightarrow Stable\,, \\ (m_{t+\Delta t} - m_t) \geq +\theta \rightarrow 1 \rightarrow Up\,, \end{cases} \tag{2}$$

where θ is set to the tick size and m_t is the mid-price at time t. Note that time horizons are defined by LOB updates, which occur at uneven intervals, rather than by fixed time steps. Tables 5, 6, and 7 display the average daily class distribution for each stock within the training, validation, and test sets, aggregated over the analysis period for $H\Delta\tau \in \{5, 10, 50\}$.

Table 5: Stocks' average daily class distribution for the training set, computed for $H\Delta\tau \in \{5, 10, 50\}$.

Ticker	H5			H10			H50		
	Down	Stable	Up	Down	Stable	Up	Down	Stable	Up
BTC-USDT	0.13	0.75	0.13	0.18	0.63	0.18	0.38	0.23	0.38
ETH-USDT	0.19	0.61	0.19	0.28	0.44	0.28	0.45	0.10	0.45
SOL-USDT	0.28	0.44	0.28	0.34	0.31	0.35	0.46	0.08	0.46
DOGE-USDT	0.13	0.73	0.13	0.21	0.59	0.21	0.39	0.22	0.39
ACM-USDT	0.16	0.67	0.16	0.22	0.55	0.22	0.34	0.33	0.33

Table 6: Stocks’ average daily class distribution for the validation set, computed for $H\Delta\tau \in \{5, 10, 50\}$.

Ticker	H5			H10			H50		
	Down	Stable	Up	Down	Stable	Up	Down	Stable	Up
BTC-USDT	0.19	0.64	0.18	0.27	0.48	0.26	0.44	0.13	0.42
ETH-USDT	0.15	0.71	0.15	0.23	0.54	0.23	0.43	0.16	0.42
SOL-USDT	0.24	0.52	0.24	0.32	0.36	0.32	0.43	0.12	0.44
DOGE-USDT	0.15	0.70	0.15	0.23	0.55	0.22	0.39	0.23	0.39
ACM-USDT	0.06	0.88	0.06	0.10	0.80	0.10	0.26	0.48	0.26

Table 7: Stocks’ average daily class distribution for the test set, computed for $H\Delta\tau \in \{5, 10, 50\}$.

Ticker	H5			H10			H50		
	Down	Stable	Up	Down	Stable	Up	Down	Stable	Up
BTC-USDT	0.17	0.66	0.17	0.25	0.50	0.25	0.42	0.15	0.43
ETH-USDT	0.27	0.47	0.27	0.35	0.30	0.35	0.48	0.05	0.48
SOL-USDT	0.24	0.52	0.24	0.33	0.36	0.32	0.44	0.12	0.44
DOGE-USDT	0.23	0.54	0.23	0.31	0.38	0.31	0.43	0.14	0.43
ACM-USDT	0.04	0.93	0.03	0.07	0.87	0.07	0.22	0.57	0.22

Generally, the data reveals distinct patterns in class distributions across horizons, with specific assets exhibiting unique directional tendencies that do not strictly correlate with traditional liquidity-based groupings.

At $H\Delta\tau = 50$ in the training set, SOL and ETH exhibit high proportions of the 'Down' and 'Up' classes, each reaching approximately 45-46%, while the 'Stable' class drops to a very low 8-10%. This alignment between SOL and ETH, despite differences in liquidity, suggests that they share similar directional behaviors at extended horizons, potentially due to comparable volatility profiles.

In the validation set at $H\Delta\tau = 10$, BTC and DOGE demonstrate a moderate balance between stability and movement, with the 'Down' and 'Up' classes each making up 22-27% of observations, while the 'Stable' class remains substantial at 48-55%. This pattern indicates that BTC and DOGE experience a mix of stability and moderate directional shifts at medium horizons, distinguishing them from the more pronounced imbalances observed at $H\Delta\tau = 50$.

At $H\Delta\tau = 50$ in the test set, BTC and ETH display a markedly high representation of the 'Down' and 'Up' classes, with each comprising around 42-48% of observations, while the 'Stable' class remains low at 10-15%. This shift towards active classes reflects a strong directional movement away from stability at longer horizons, a characteristic that is consistent across all dataset types.

Lastly, ACM, the lowest-liquidity asset, consistently displays a very high proportion of the 'Stable' class across all horizons. In the validation and test sets at shorter horizons ($H\Delta\tau = 10$), the 'Stable' class accounts for 80-93% of observations, highlighting ACM's limited responsiveness to short-term market fluctuations. Even at $H\Delta\tau = 50$, ACM maintains a high level of stability, with the 'Stable' class constituting 48-57% of observations, suggesting that low-liquidity assets like ACM exhibit sustained stability across extended horizons.

These patterns highlight that while liquidity factors such as the trading volume and tick size may influence class distributions to some extent, other latent liquidity factors such as volatility contribute significantly to the observed class imbalance across horizons. Indeed, in all symbols the Stable class is the most represented at $H\Delta\tau = 5$ and is much lower relative to Up/Down classes as the prediction horizon increases. This just means that the price is more likely to move as time passes which is consistent with volatility being proportional to the square-root of time.

Overall, the target class distributions for each asset remain roughly consistent across the training, validation, and test sets within each horizon. Although some assets will display a slight change in the imbalance of the target class distribution such as ETH, DOGE and ACM at $H\Delta\tau = 5$ and $H\Delta\tau = 10$, the inter-dataset shift within the same class is mild when compared to the Stable proportion to Down and Up classes. This confirms that the dataset preparation process preserved the underlying class distributions across different dataset types.

Given the imbalance of the target class is pronounced for most securities across different prediction horizon, the model will learn more the Stable class at $H\Delta\tau = 5$ and learn it less at $H\Delta\tau = 50$, however we want the model to perform well on each class for any prediction horizon.

The sampling process is handled differently for the training, validation, and test sets. In the training set, sub-sampling is performed as inspired by Biola et al. [8, 9] in a random and balanced manner. For each trading day, we identify the number of samples N in the least represented class. If this count is at least 5000, we randomly sample N representatives for the other 2 classes, as defined in Equation (2). However, if the count is less than 5000, that date must display some uncommon behavior (consistently high/low activity, trending upward or downward all day) and is discarded from the training set.

There are significant differences in the liquidity of the assets in this study, which is impacting the size of the dataset by symbol, measured in number of observations. Indeed, a single day's number of observations can vary widely from one symbol to the next, with BTC-USDT reaching approximately 600,000 observations per day on average, while for ACM-USDT one day of data would represent about 15,000 observations. Furthermore, for a same crypto the distribution of the predicted classes for different predicted horizons can differ vastly therefore my under-sampling in the training set could create training set with very different training set sizes for different time horizons within a same security. Since I am applying the same models with the same number of parameters across different crypto pairs, it would create challenges of over-parameterization in less liquid assets such as ACM-USDT, introducing an unintended bias due to the disparity in the observation-to-parameter ratio. This discrepancy could skew the cross-asset comparison of the results.

The 2 largest models that we experimented with in this paper (DeepLOB and Transformer) have from 100,000 to 150,000 parameters. The Chinchilla paper [17] introduces the scaling law for LLMs defining a proportional relationship between the number of tokens and the model size, in order to achieve compute-optimal training. They set this factor at about 20, that I used for this use-case with much smaller models than LLMs to roughly 2 million observations for a model with 100k parameters. To address this, I established a uniform dataset size, aiming for 2 to 2.5 million observations in the training set for each asset (I did not want to cut the dataset intraday). Validation and test sets were set consistently across all prediction horizons for comparison purpose within the same crypto, containing around 1 million observations each. This approach maintains comparability across assets, ensuring a balanced training set and avoiding biases stemming from liquidity disparities. For BTC/USDT, however, the substantial daily observation counts sometimes exceeded my memory limits, requiring me to sample a smaller daily subset to prevent crashes.

To maintain consistency despite the variability in daily observation counts across trading pairs, we applied rolling window z-score normalization using a fixed number of observations (150,000) rather than a fixed time window. The rational for that was to have a stable statistic for even the least liquid asset. This approach typically covers less than a day for BTC/USDT but can extend to over a week for a lower-liquidity pair like ACM/USDT. Due to ACM/USDT's illiquidity, the historical data provided did not extend far enough to achieve a sufficient training size when under-sampling the least represented class, as was feasible with the other pairs. Therefore, we had to oversample with replacement the under-represented class and under-sample without replacement the over-represented class.

4. Methods

This paper aims to provide an accessible method for estimating the predictability of cryptocurrencies based on the microstructural characteristics of their limit order books (LOB). The research is structured in two stages, as inspired by Briola et al. [8]: (i) we analyze liquidity features of a diverse set of cryptocurrencies to identify clusters for classification (see Section 6), and (ii) we conduct a forecasting analysis on each cryptocurrency, comparing the results against insights from the initial classification.

For the forecasting, I open-source my codebase that is designed to handle cryptocurrency LOB data with limited compute and RAM – 52Gb (CPU) and 24Gb (GPU) – resources, enabling use by anyone with access to platforms like SageMaker or Google Colab. This framework integrates recent scientific advancements into a cohesive system. One of my codebase's key benefits is its end-to-end pipeline, which includes data transformation, high-speed implementations for training, validation, and testing, and comprehensive model evaluation through trading simulations. Additionally, this codebase allows for the integration of new models. The findings in this study focus on the performance of major state-of-the-art models commonly picked for performance comparison [5, 8, 9]: DeepLOB, Transformers, B(TABL), C(TABL), providing a benchmark to guide future improvements and innovations.

The models chosen for this study—DeepLOB, Transformers, and both variants of BiNTABL (B-TABL and C-TABL)—represent state-of-the-art approaches in financial time-series forecasting, each bringing unique strengths to the prediction of short-term price movements. DeepLOB, originally developed by Zihao Zhang et al. [5], employs a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) layers to extract spatial and temporal patterns from Limit Order Book (LOB) data. Its hierarchical architecture effectively reduces the complexity of high-frequency LOB data, enabling robust feature extraction and sequential processing. The established Transformer model [23], leveraging self-attention mechanisms, excels in capturing long-term dependencies and relationships across the temporal dimension of the data, making them highly effective for tasks requiring detailed temporal context, such as LOB forecasting. Finally, the TABL models [22], which integrate bilinear and temporal attention mechanisms, provide a novel approach to understanding interdependencies between features and temporal slices. The B-TABL and C-TABL have a similar architecture, C-TABL has an extra Bilinear Layer to refine features and time interactions further. These models' diversity in design ensures comprehensive coverage of the spatiotemporal dynamics inherent in LOB data, offering insights into market microstructure that are essential for improving predictive performance in this high-frequency domain.

The entire framework is built in Python using the PyTorch library. We initially experimented with DeepLOB from an alternate framework in TensorFlow, guided by Zihao Zhang's GitHub repository which includes implementations in both PyTorch and TensorFlow. Using the same random seed and hyperparameters as in my Pytorch implementation — such as Kaiming initialization for CNN and Dense layers (instead of Glorot) and a Batch Normalization momentum of 0.1 (instead of 0.99) — the TensorFlow implementation produced notably poorer validation performance across various learning rates and models. Uncertain about the source of this performance disparity between TensorFlow and PyTorch defaults, we chose to proceed solely with PyTorch. For the other models we used the implementation provided by Antonio Briola on his GitHub repo, he also used PyTorch [8, 9].

To train, validate, and test the models, we implemented an efficient data loader that samples mini-batches of size 128, differing from the original model's setup of 32. Increasing the batch size offers several efficiency benefits. First, processing larger batches reduces the frequency of data transfers from CPU to GPU, maximizing computational throughput. Additionally, a larger batch size produces smoother gradient estimates, which helps to offset the high noise in cryptocurrency data and leads to more stable parameter updates. This approach prioritizes faster, steadier convergence by sacrificing a bit of the generalization benefits that smaller, noisier batches might offer through extended training. Given the compute and RAM constraints we faced, this trade-off — slightly reduced generalization in favor of efficient, reliable convergence — was considered worthwhile.

We initially trained the models with a high maximum number of epochs and an early-stopping patience of 15 epochs, as suggested in established literature [8]. However, as we monitored the training process, we began adjusting the learning rate manually whenever we observed the model plateauing after 5 to 10 epochs. In these cases, we experimented with both significantly higher learning rates to try escaping potential local minima, and, if unsuccessful, progressively lowered the learning rate to fine-tune performance gains. This approach continued until the learning rate dropped below $1 \text{ x } 10^{-6}$, yielding better results than the initial fixed schedule. While we recognize that this approach was time-consuming, my limited compute resources made it essential to maximize efficiency by halting training early whenever convergence seemed unlikely after a few epochs, rather than letting it continue to the full 15-epoch patience limit. The training uses a modified Adam optimizer with decoupled weight decay (AdamW), configured with an initial learning rate specific to each pair, ranging from $1 \text{ x } 10^{-6}$ to

$1 \text{ x } 10^{-3}$, along with decay rates of 0.9 for β1 and 0.99 for β2, the default settings in PyTorch's torch.optim library.

For training, the dataloader samples mini-batches randomly (shuffle=True) to increase data diversity, helping the model avoid overfitting to specific sequences and learn more generalizable patterns over the training set. For the validation and test phases, the batches are sequential, covering the entire dataset for each respective phase. Each mini-batch consists of input samples with dimensions T×40, $T\Delta\tau \in \{20, 50, 100\}$, where T represents the temporal sequence length (i.e., the number of consecutive LOB updates used as backward-looking history for each sample), and 40 represents the spatial components of each LOB snapshot (see Equation (1)). The values of T were selected using a simple heuristic aimed at maintaining approximately a 5-to-1 ratio between the forward-looking and backward-looking windows. This ratio is based on the rapid decay observed in auto-regressive relationships in financial markets, where signal-to-noise ratio diminishes quickly over extended historical windows. However, this rough 5-to-1 ratio could not be maintained for the tuple (H50, T100) with prediction horizon $H\Delta\tau = 50$ and sequence length $T\Delta\tau = 100$ due to memory limitations, as a T value exceeding $T\Delta\tau = 100$ resulted in system crashes.

To manage training datasets that exceed available CPU/GPU RAM, we developed two versions of the training code. For $T\Delta\tau \in \{20, 50\}$, a single implementation loads the entire training and validation set into one data loader. However, for $T\Delta\tau = 100$, a different approach was necessary to stay within memory limits. In this version, we first mapped the number of observations by date, then selected dates incrementally through a random sampling process (without replacement) to build a set ranging from 500,000 to 800,000 observations—this upper threshold was set to avoid RAM crashes. Each chunk undergoes additional shuffling during conversion into a DataLoader object to create mini-batches, and the model trains sequentially over these chunks until completing each epoch. While this approach successfully handles larger datasets, it significantly increased training time due to the repeated overhead loading.

In total, we conducted 60 training experiments on a private Google Colab Pro+ account, accumulating approximately 325 hours of GPU runtime. We initially tested the setup on my machine with an NVIDIA GeForce MX250, but the compute was insufficient. As a result, the experiments were run on Google Colab's GPUs: (i) NVIDIA T4 and (ii) NVIDIA A100.

5. Microstructural Properties

As an initial microstructural property, we examine the relationship between the cryptos' average spread, $\langle\sigma\rangle$, and the tick size [6, 18], θ, over the analysis period. In the literature [2], stocks are typically classified based on a general rule: if $\langle\sigma\rangle \gg \theta$, the asset is considered a small-tick stock, whereas if $\langle\sigma\rangle \approx \theta$, it is classified as a large-tick stock. While commonly used, this classification lacks a quantitative basis and may be overly restrictive [8], potentially overlooking the more nuanced behavior of crypto pairs traded on the Binance exchange.

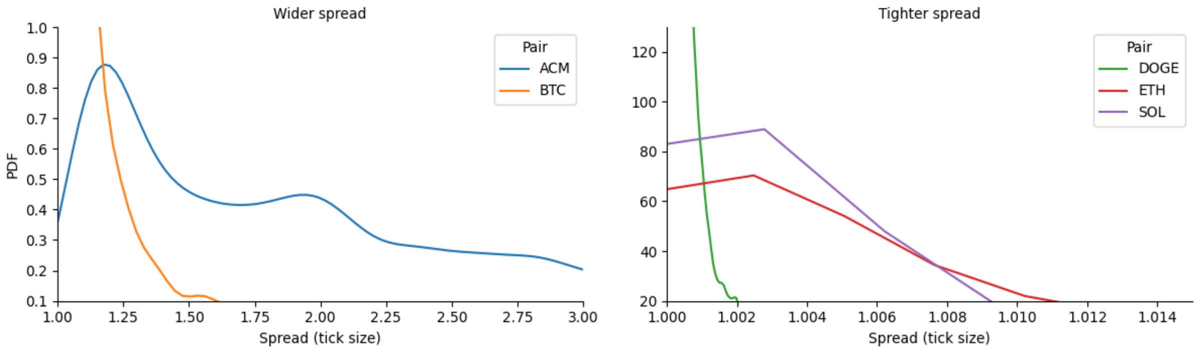
Table 8: The 5 small, medium and large tick crypto pairs compose the security universe of this study, with the average mid-price, spread and the tick-size.

pair	mean spread ($)	mean spread (tick size)	mean price	tick size	group
ETH	0.01	1.01	2513.31	0.01	large
SOL	0.01	1.00	158.55	0.01	large
DOGE	0.00001	1.00	0.10494	1.00E-05	large
BTC	0.01	1.15	63398.61	0.01	medium
ACM	0.002	2.02	1.998	0.001	small

In this paper we classify the 5 crypto pairs in 3 groups as follows: if (i) $\langle\sigma\rangle = \theta$, we observe so-called large-tick pairs, if (ii) $\theta \leq \langle\sigma\rangle \leq 1.5\theta$ then we are dealing with so-called medium-tick pairs, if (iii) $1.5\theta \leq \langle\sigma\rangle$ then the pair is classified as a small-tick pair.

Looking at Table 8: we have 3 representatives of large-tick pairs (ETH, DOGE, SOL) presenting a spread of almost always 1 tick which is the tightest possible spread, 1 representative of medium-tick spread (BTC) and 1 representative of small-tick spread (ACM). The yearly average spread is subject to non-negligible fluctuations, specifically for the medium and small-tick pairs, as illustrated with the PDF of the spread measured in tick size Figure 3. Notably, unlike equity market studies in machine learning, which often focus on assets with mid-prices in a similar range, the mean mid-prices here display vast disparities. For instance, BTC's mean price of about 60,000 USDT contrasts sharply with DOGE's rough 0.1 USDT magnitude, showing a factor difference of about 1.10^{5}, a range that is not uncommon in currency markets (e.g., USDJPY rate is in the range of 1.10^{2} ; USDIDR rate is in the range of 1.10^{4} for Indonesian Rupiah) and by extension with cryptocurrencies.

Figure 2: PDF of the spread (expressed in number of ticks) over the 5 crypto pairs of interest, in the period of analysis.



The distributions reveal distinct behavioral clusters among small-, medium-, and large-tick pairs. For large-tick stocks, distributions sharply peaked around an average of 1 tick, almost exclusively the minimum allowable spread of 1 tick. From a practical standpoint, tighter spreads benefit traders by allowing quick entry and exit from positions with minimal transaction costs for liquidity takers. The medium-tick pair BTC presents a distribution peaking slightly above the minimum possible spread, with an average peak value of 1.15 tick, indicating more variation than is seen with large-tick pairs. This outcome aligns with expectations, borrowing the rational from the equity markets: as medium-tick securities are inherently ‘borderline’ assets with behavioral patterns that do not distinctly align with either small or large-tick equivalents. The small-tick pair (ACM), in contrast, display consistently broader distributions. Similar to observations drawn from the equity markets, the small-tick securities exhibit greater variance than large-tick ones, consistent with liquidity characteristics such as less frequent trading or larger sparse orders that may influence the market, leading to increased market impact for liquidity takers.

Figure 3: CCDF of the volumes available at the best quotes for the 5 crypto pairs during the period of analysis.

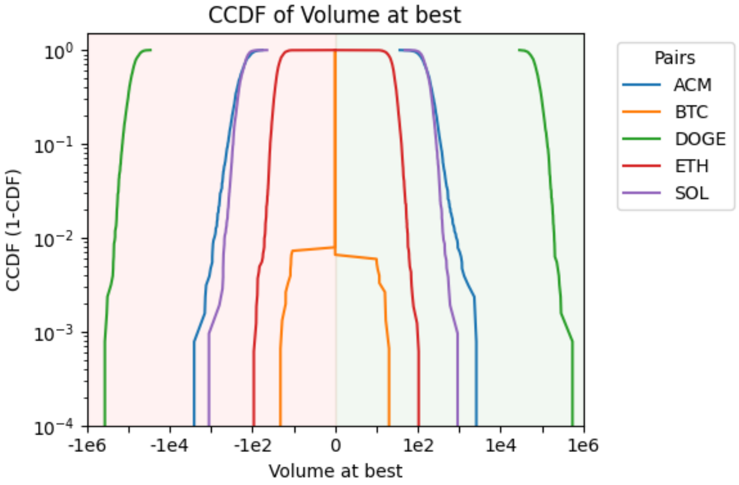
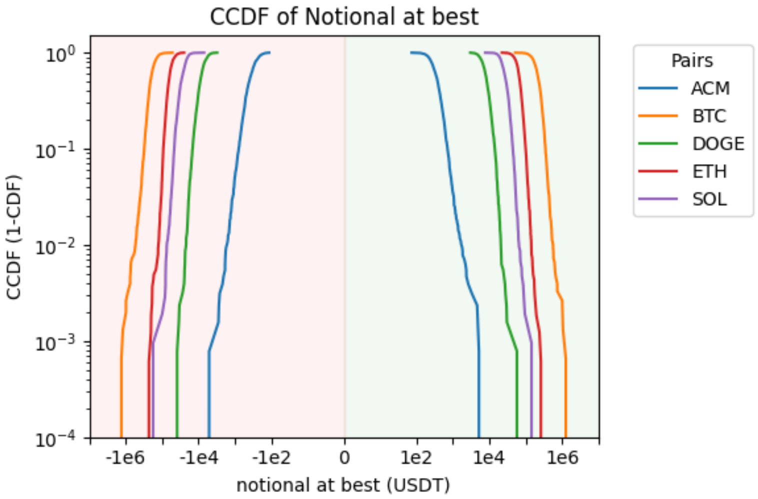


Figure 3 presents the analysis, using a symmetric log scale on the x-axis, as per Briola and al. [8], to examine both the bid side (negative part, red area) and the ask side (positive part, green area) of the LOB, emphasizing the distribution spread. The distributions appear generally symmetric across both sides, indicating no significant imbalance in price movement during the analyzed period. Unlike in equity markets, where clustering by tick size aligns with other liquidity measures such as spread normalized by tick size, the substantial differences in exchange rates across the pairs disrupt the consistency of the relationship between tick size and volume observed in equities. Another key distinction is the presence of lot sizes in U.S. equity markets, which are traditionally used by institutional investors handling larger orders. These lot sizes create a de facto lower bound for displayed volumes, likely causing small-tick stocks’ minimum displayed volumes to approach the same order of magnitude as those of larger-tick stocks. In crypto markets, however, this conventional volume threshold is absent which can explain the finer granularity in quoted volumes, hence complicating direct volume-based comparisons that we can find in the equity markets-based LOB literature.

Figure 4: CCDF of the notional available at the best quotes for the 5 crypto pairs along the period of analysis.



To account for these differences, it is more appropriate to compare the notional values (dollar terms) displayed at the top of the book across various pairs, as this provides a more robust metric for cross-asset comparisons. Figure 5, which illustrates the notional values displayed, reveals a clustering pattern among the large-tick pairs (ETH, SOL, and DOGE) and shows that the medium-tick pair BTC does not deviate significantly from this group, while the small-tick pair ACM stands out as notably dissimilar. This notional-based liquidity measure aligns closely with the market cap hierarchy presented earlier, offering a consistent order of liquidity across pairs. In equities, the steepest distributions often belong to the most volatile securities, which aligns with the pronounced steepness observed in both volume and notional distributions for cryptocurrencies in Figure 5, highlighting their inherently higher volatility compared to stocks. In contrast, comparisons that can be found in equity markets literature regarding the wideness of the distribution based on tick size fail to hold here, likely due again to the absence of a traditional use of the lot size. While round notional amounts are also common in currency and crypto markets, their values are based on individual traders' heuristics rather than the SEC-mandated standards applied in U.S. equity markets. The only exception is Binance's minimum order size, which is specific to each currency but remains extremely low (see Table 3). This lack of uniformity in crypto markets is likely the primary factor of the absence of consistency found in equity markets.

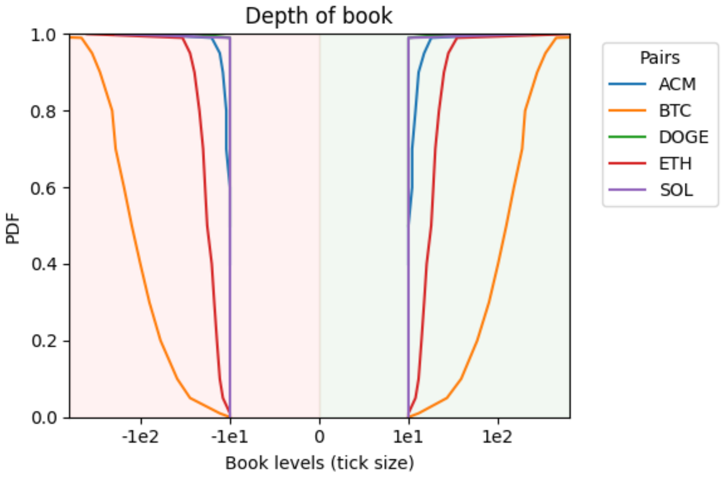
As highlighted by Wu et al. [19], the LOB representation used in this paper (see Equation (1)), known as the "compressed representation", has a significant limitation: its spatial structure is non-homogeneous (see Figure 1). This is because there is no assumption that adjacent price levels are evenly spaced; only a monotonic order is maintained. Such representation is susceptible to sudden changes due to shifts in price levels, which can greatly affect predictability when used as input for deep learning models. Wu et al. emphasize that a core assumption in deep learning is that signals within the same channel (or input dimension) originate from a consistent source, and they reveal the fragility of these feature representations under adversarial perturbations. We acknowledge this limitation in their findings and recognize that our results may be influenced by the arbitrary order chosen for features to represent the order book — a factor that we do not investigate further in this study. Here, each "level" in the LOB is specific to a single snapshot and lacks association with a stable source, especially as aggressive orders can shift this information. To evaluate a stock's sensitivity to this issue, we calculate the metric introduced by Briola et al. [8, 9] termed "actual LOB depth" (Ξ). For a given snapshot $\mathbb{L}(\tau)$, this measure is calculated for both sides of the market as follows:

$$\Xi^{Ask} = \frac{p_{10}^{ask} - p_1^{ask}}{\theta} \quad \text{ask side,}$$

$$\Xi^{Bid} = \frac{p_{10}^{bid} - p_1^{bid}}{\theta} \quad \text{bid side.}$$

This 'actual LOB depth' metric was originally developed for equity markets and may be influenced by the extremely low minimum quoted volumes in crypto markets. Regardless, we will use this metric with an understanding of its limitations, since we believe that imposing a minimum volume threshold could introduce an inductive bias given the price-dependent nature of crypto volumes.

Figure 5: PDF of the 'actual LOB depth' for the 5 crypto pairs along the period of analysis.



In Figure 5, we present the PDF for Ξ^{Bid} and Ξ^{Ask} across the period of analysis for each cryptocurrency pair, using a symmetric log scale on the x-axis. A distribution starting at 10 (or -10) indicates that the first nine consecutive price levels immediately below the top of the book (level 1), have each quoted volumes spaced at intervals of exactly one tick size. Occasionally, price levels with quoted volume are separated by more than one tick-size interval, resulting in an "actual LOB depth" greater than 10.

Figure 5 shows that BTC has the sparsest liquidity across these actual depth levels, with intervals between consecutive quoted levels often reaching multiples of tens or even hundreds of tick sizes within the first 10 levels. BTC exhibits the widest occasional intervals, and those wide intervals are occurring more frequently than in the other pairs. Compared to BTC, ETH displays a steeper distribution, reflecting a more compact book with fewer liquidity gaps and smaller intervals between consecutive quoted levels. ACM has an even more compact structure, while SOL and DOGE—represented below the SOL curve—show the tightest book, with the first 10 price levels almost always quoting some volume.

This ordering aligns closely with the CCDF of volume at the best, except for ACM. BTC has the smallest quoted volume in currency terms, contrasting with DOGE, which shows substantially larger volumes due to the price difference between the two (around 60,000 USDT for BTC vs. 0.1 USDT for DOGE). This difference is reflected in the book's sparsity within the first 10 quoted levels, with BTC having the sparsest book and DOGE (along with SOL) the densest.

Starting with our earlier observation about the actual depth formula's limitations—specifically, its lack of a minimum volume threshold and the much smaller minimum tradable volume in crypto compared to equities—we can further validate the observed patterns by examining the volume relative to the top of the book volume, as well as the notional quoted, which provides a robust metric for cross-asset comparison. Unlike the previous actual depth measure, the depth of the book here is averaged over the analysis period for each security and side, with cumulative averages at each level calculated as multiples of tick size from the top, regardless of whether there is volume quoted.

Figure 6: Cumulative volume across depth levels.

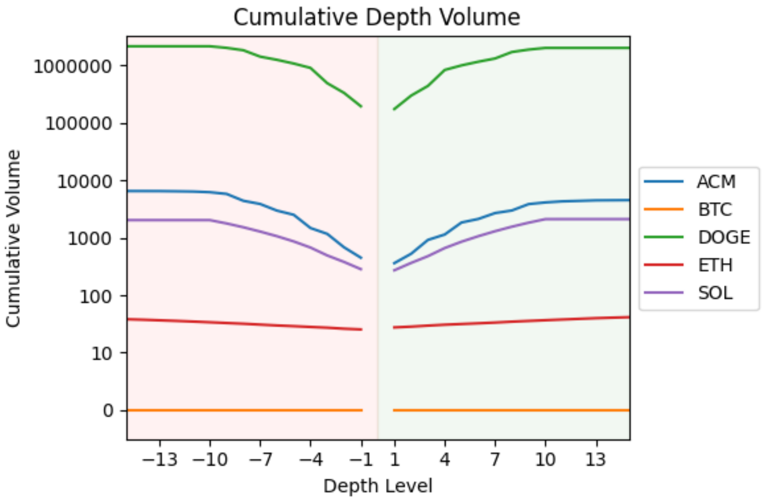


Figure 7: Cumulative volume normalized by top of the book volume across depth levels.

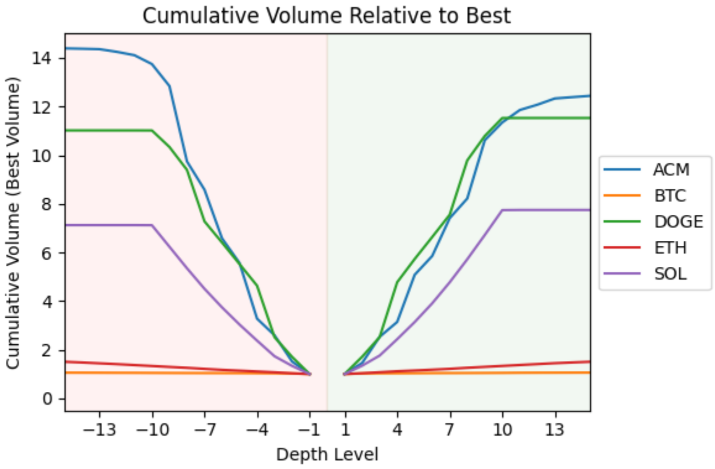
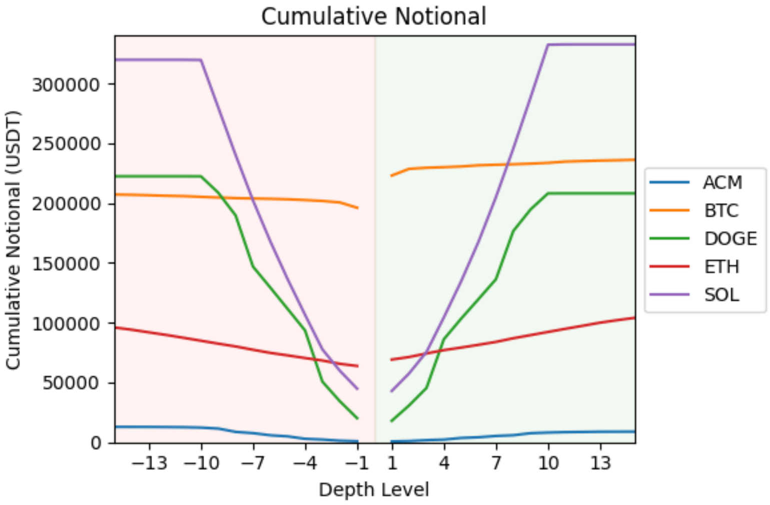


Figure 8: Cumulative volume across depth levels.



In Figure 6, we observe a similar hierarchy aligned with market capitalization in the Cumulative Volume Relative to Best. Consistent with the actual depth measure, two clusters emerge: altcoins ACM, DOGE, and SOL display a dense book up to level 10, but with scarce liquidity beyond that point. In contrast, BTC and ETH, the "blue-chip" cryptocurrencies, show comparatively smaller volumes quoted near the top but maintain liquidity further into depth compared to the altcoin cluster. This alignment with the actual depth measure confirms the earlier findings from Figure 6.

Figure 7 further highlights this distinction, with BTC showing notably smaller quoted volumes compared to other cryptocurrencies. This difference stems from Bitcoin’s widespread adoption as an investment vehicle and its substantial price disparity with altcoins, in part due to the lack of price-reduction mechanisms like stock splits in equities. ETH exhibits a similar but less pronounced pattern.

Figure 8 supports this interpretation: the value of the BTC volumes quoted at on top-of-the-book is comparable to the entire cumulative worth of DOGE’s liquidity within the first 10 levels of depth. Additionally, the shape of the cumulative notional distribution, both in terms of slope and curvature, offers further insights. BTC stands out, with substantial amounts quoted within the first three levels, such that large block trades would need to be handled over-the-counter (OTC) to avoid moving through multiple layers. At the opposite end, ACM, the least liquid currency, does not show significant volume increases at deeper levels, indicating that a critical level of liquidity may be needed before agents adopt posting strategies similar to those seen in SOL and DOGE.

Lastly, the structural differences in the books of ETH versus SOL and DOGE are notable, even though the notional values posted are within a similar range. This likely reflects the higher volatility of SOL and DOGE, prompting market participants to position liquidity deeper in the book to capture potential trading opportunities arising from their greater price fluctuations. A potential factor contributing to this dynamic is the greater maturity of BTC and ETH, which benefits from earlier and more comprehensive regulatory frameworks. These frameworks paved the way for the introduction of Exchange-Traded Funds (ETFs) tied to BTC and ETH, attracting a more diverse investor base that includes retail investors and institutions. In contrast, DOGE and SOL tend to appeal more to niche participants such as retail crypto enthusiasts, arb funds, and market makers, resulting in LOB profiles resembling an inverse butterfly payoff at maturity that reflect their higher volatility.

Table 9: Average probability that the number of updates characterizing the three horizons $H\Delta\tau \in \{5, 10, 50\}$, happens in a physical time (i) < 1 second; (ii) ≥ 1 second and < 10 seconds; or (iii) ≥ 10 seconds.

Ticker	<1s			>=1s & <10s			>= 10s		
	H5	**H10**	**H50**	**H5**	**H10**	**H50**	**H5**	**H10**	**H50**
BTC	0.26	0.00	0.00	0.74	1.00	0.92	0.00	0.00	0.08
ETH	0.09	0.00	0.00	0.91	1.00	0.31	0.00	0.00	0.69
SOL	0.00	0.00	0.00	0.99	0.77	0.00	0.01	0.23	1.00
DOGE	0.01	0.00	0.00	0.98	0.95	0.00	0.00	0.05	1.00
ACM	0.00	0.00	0.00	0.33	0.09	0.00	0.67	0.91	1.00

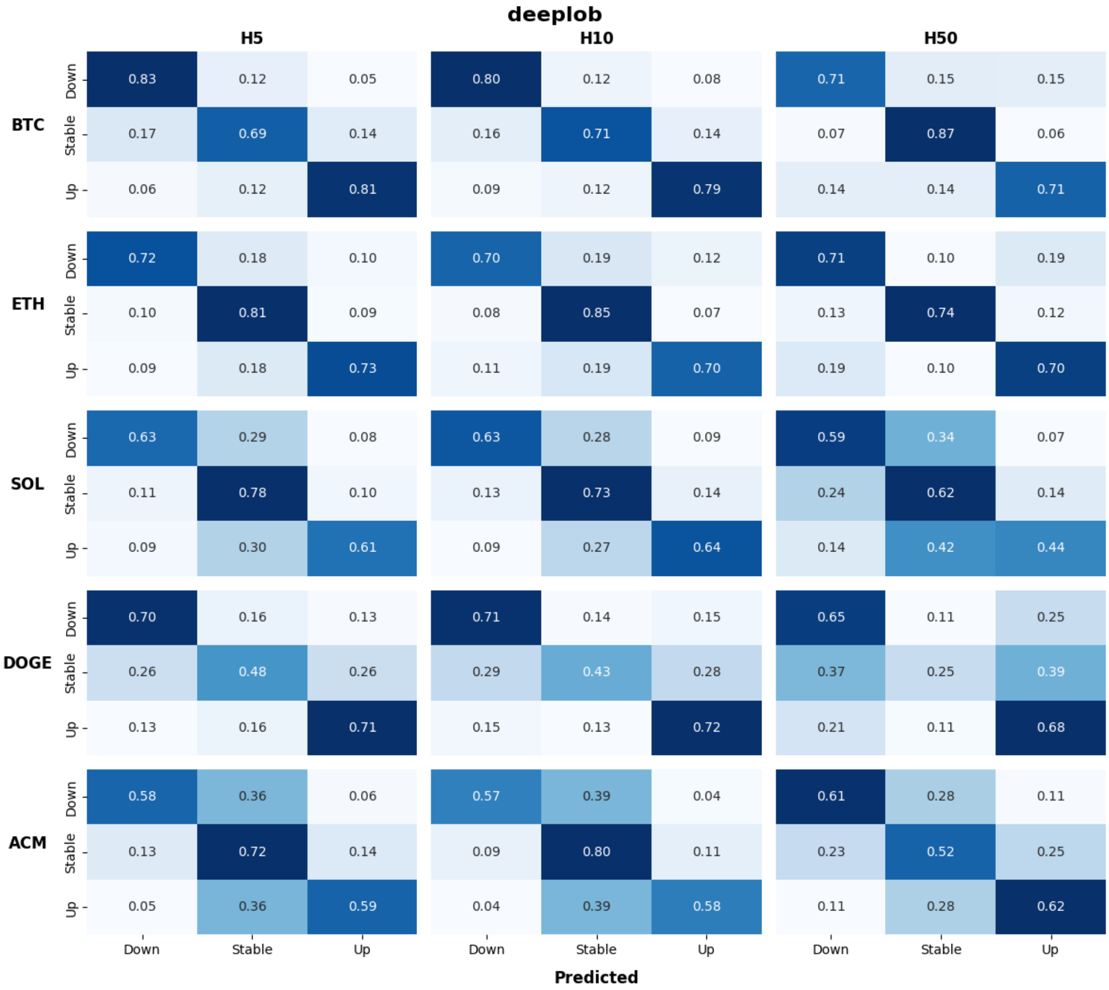
In previous analyses, we have consistently defined time based on the number of LOB updates (i.e., "tick time"). This approach results in different mappings between physical time and tick time across stocks, presenting a challenge for practitioners who prioritize real-world applicability over academic exercises. Table 9 displays the average probability (computed over a 3-year period) that updates for the three horizons $H\Delta\tau \in \{5, 10, 50\}$ occur within physical times of (i) < 1 second, (ii) ≥ 1 and < 10 seconds, or (iii) ≥ 10 seconds. For each horizon, these three probabilities sum to 1.

Across most cryptocurrencies (excluding ACM, a small-tick pair), 5 LOB updates are more likely to happen within ≥ 1 and < 10 seconds. For 10 updates, the majority occur within ≥ 1 and < 10 seconds, with at the exception of ACM (small-tick). Lastly, the 50-update horizon reveals clear behavioral clusters among cryptocurrencies classes: for large-tick and small-tick, 50 updates are more likely to occur over ≥ 10 seconds, with BTC as exception due to its higher trading activity.

6. Results

6.1 Assing models' forecast performances using Confusion Matrixes

Figure 9: Confusion Matrices for model DeepLOB across the three prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. To create these summarized representations, we first calculate individual confusion matrices for each pair, each time horizon over the test set. These matrixes are subsequently normalized row-wise to transform raw counts into proportions, providing insight into predictive accuracy and class-specific performance. The final normalized matrix effectively visualizes the model's capability in classifying mid-price movement directions throughout the testing period.



		H5			H10			H50		
		Down	Stable	Up	Down	Stable	Up	Down	Stable	Up
BTC	Down	0.83	0.12	0.05	0.80	0.12	0.08	0.71	0.15	0.15
	Stable	0.17	0.69	0.14	0.16	0.71	0.14	0.07	0.87	0.06
	Up	0.06	0.12	0.81	0.09	0.12	0.79	0.14	0.14	0.71
ETH	Down	0.72	0.18	0.10	0.70	0.19	0.12	0.71	0.10	0.19
	Stable	0.10	0.81	0.09	0.08	0.85	0.07	0.13	0.74	0.12
	Up	0.09	0.18	0.73	0.11	0.19	0.70	0.19	0.10	0.70
SOL	Down	0.63	0.29	0.08	0.63	0.28	0.09	0.59	0.34	0.07
	Stable	0.11	0.78	0.10	0.13	0.73	0.14	0.24	0.62	0.14
	Up	0.09	0.30	0.61	0.09	0.27	0.64	0.14	0.42	0.44
DOGE	Down	0.70	0.16	0.13	0.71	0.14	0.15	0.65	0.11	0.25
	Stable	0.26	0.48	0.26	0.29	0.43	0.28	0.37	0.25	0.39
	Up	0.13	0.16	0.71	0.15	0.13	0.72	0.21	0.11	0.68
ACM	Down	0.58	0.36	0.06	0.57	0.39	0.04	0.61	0.28	0.11
	Stable	0.13	0.72	0.14	0.09	0.80	0.11	0.23	0.52	0.25
	Up	0.05	0.36	0.59	0.04	0.39	0.58	0.11	0.28	0.62

In Figure 9, we present the confusion matrix of the DeepLOB model (as a representative example; confusion matrices for other models are provided in the appendix) for each asset across prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. The performance exhibits a distinct trend of increasing misclassifications with both decreasing liquidity and extended prediction horizons. Many trading strategies, including those to which we will apply these models, rely on accurate classification of extreme classes (-1 and 1), which correspond to anticipated 'Down' and 'Up' movements, respectively. Consequently, frequent reciprocal misclassifications between these extreme classes are particularly problematic, as they are more impactful than misclassifications between an extreme class and the 'Stable' class.

Specifically, at $H\Delta\tau = 5$, for the most liquid pair BTC, there are only 5% of extreme classes mutual misclassification (in each class) and 12% misclassification from the extreme class into the Stable class. For comparison on the other end of the liquidity dimension with the least liquid pair ACM, 36% of actual class 1 and -1 instances are misclassified as class 0. At $H\Delta\tau = 50$ for BTC, extreme classes mutual misclassification jumps from 5% to 15%. Overall, the most liquid pairs (BTC and ETH) have the lowest combined misclassification rates (highest scores on the diagonal compared to other securities). In this case DOGE presents similar predicting performances as ETH, although slightly higher extreme class misclassification. In contrast, for the least liquid securities (SOL, ACM), the model's predictive performance significantly worsens, with a relatively slightly stronger score vs BTC in classifying the extreme classes correctly but much larger misclassification from extreme to Stable class.

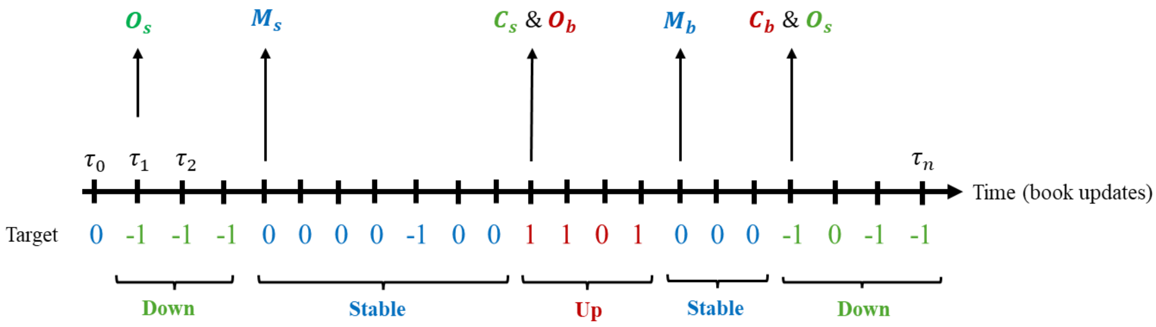
Similar results can be observed across the other models, in different absolute and relative proportions across these same two criteria for decreasing performances: decreasing liquidity, increasing time horizon. You can find the details of the confusion matrixes for these other models in Appendix A. The coherence of the results bolsters the robustness of the findings discussed earlier in this Section, highlighting that liquid pairs exhibit a significant predictability rate across all the considered horizons and this characteristic is consistent regardless of which model is used.

6.2 Assing models' forecast performances using metrics along different probability thresholds

Confusion matrices provide a foundational framework for analyzing the behavior of predictive models across a variety of contexts. These matrices detail the distribution of model forecasts, offering a valuable perspective on performance in diverse scenarios. However, to fully grasp a model's effectiveness and to enable more refined evaluations, derived metrics are indispensable. Such metrics provide deeper insights into the model's predictive accuracy and patterns of error, allowing for a thorough examination of its capabilities and limitations. Through these metrics, researchers and practitioners can better understand each model's potential, guiding informed decisions for application and improvement. To evaluate the predictive performance of the different models covered in this study, we employ the Matthews Correlation Coefficient (MCC). MCC generalizes Pearson's correlation, measuring the relationship between actual and predicted classes. Ranging from −1 (indicating inverse prediction) to +1 (indicating perfect prediction), with 0 representing random predictions, MCC is a balanced metric that is particularly useful when class sizes vary significantly [20, 21].

To support a strategy-focused analysis of the models' forecasts, we use a metric specifically designed to be assumption-free and robust to class imbalances introduced by Briola et al. [8]. In scenarios where mid-price direction forecasts are chronologically sorted (as illustrated in Figure 10), this approach evaluates predictive accuracy in a manner that aligns closely with real-world trading scenarios.

Figure 10: Illustration of a chronologically ordered forecast vector. Using the mapping outlined in Equation (2), we establish a simplified strategy where $O_{p \in \{s/b\}}$ indicates 'opening a new selling/buying position', $M_{p \in \{s/b\}}$ represents 'maintaining an existing selling/buying position', and $C_{p \in \{s/b\}}$ signifies 'closing an existing selling/buying position.'



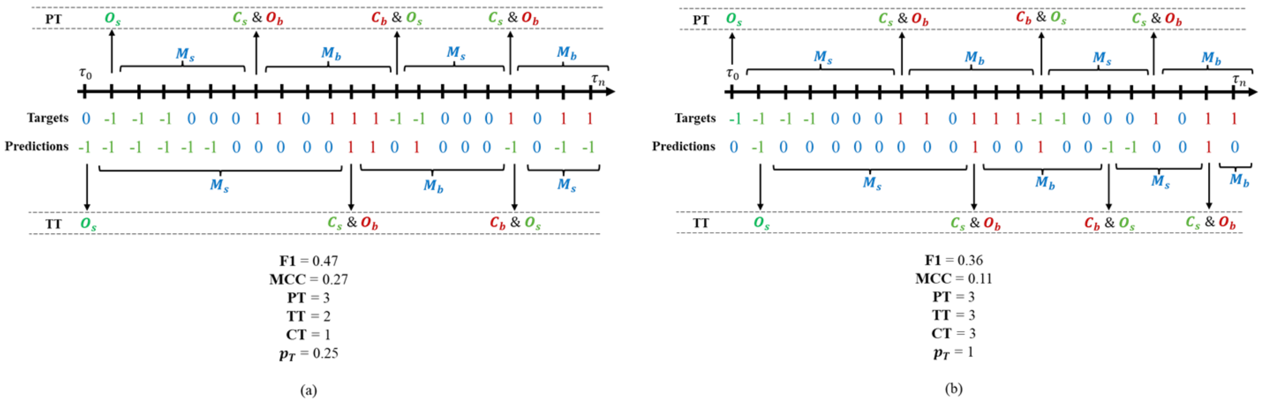
Here, trading actions are directly mapped to predictions, where (i) a selling position is opened with a predicted ‘Down’ movement (O_s); (ii) the position is maintained if the forecast indicates a ‘Stable’ period (M_s); (iii) the selling position is closed, and a buying position is opened with a predicted ‘Up’ movement (C_s for closing and O_b for opening); (iv) the buying position is maintained during predicted ‘Stable’ periods (M_b); and, (v) the buying position is closed while a new selling position is opened with a predicted ‘Down’ movement (C_b and O_s, respectively). Following this basic strategy, three positions are opened and two are closed, resulting in two completed transactions (a position being opened and later closed). Using forecasts inherently depends on their accuracy, so it’s essential to contextualize this reliance. In Figure 11, two examples of forecast sequences are presented, with their MCC, F1 score, and the following transaction-related metrics:

- Potential Transactions (PT): In Figure 10, a transaction occurs when a position is opened and closed (e.g., $O_s \rightarrow C_s$ or $O_b \rightarrow C_b$). PT is counting transactions in the target set.
- Total Executed Transactions (TT): This metric is calculated similarly to PT, but applied to the prediction set.
- Correctly Executed Transactions (CT): This metric counts transactions in the prediction set that align with the target set. In Figure 11a, CT is 1, as there are misalignments between positions in the two sets.
- Probability (p_T) of Executing a Correct Transaction: This is computed as:

$$p_T = \frac{CT}{PT+TT-CT} \tag{3}$$

In this approach, 'opening' and 'closing' a position, or 'executing' a transaction, refer to the model’s ability to identify optimal entry or exit points for trades accurately. The examples in Figure 11 illustrate the limitations of traditional metrics in assessing out-of-sample practicality for Limit Order Book (LOB) forecasting. In particular, they highlight scenarios where standard machine learning metrics diverge significantly from p_T, underscoring the gap between academic metrics and actionable forecasts.

Figure 11: Transaction-related metrics (PT, TT, CT, p_T) and machine learning metrics (MCC, F1) calculated on two chronologically ordered vectors of forecasts and their corresponding targets.


PT O_s C_s & O_b C_b & O_s C_s & O_b
M_s M_b M_s M_b
τ_0 τ_n
Targets 0 -1 -1 -1 0 0 0 1 1 0 1 1 1 -1 -1 0 0 0 1 0 1 1
Predictions -1 -1 -1 -1 -1 -1 0 0 0 0 0 1 1 0 1 0 0 0 -1 0 -1 -1
M_s M_b M_s
TT O_s C_s & O_b C_b & O_s
F1 = 0.47
MCC = 0.27
PT = 3
TT = 2
CT = 1
p_T = 0.25
PT O_s C_s & O_b C_b & O_s C_s & O_b
M_s M_b M_s M_b
τ_0 τ_n
Targets -1 -1 -1 -1 0 0 0 1 1 0 1 1 1 -1 -1 0 0 0 1 0 1 1
Predictions 0 -1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 -1 -1 0 0 1 0
M_s M_b M_s M_b
TT O_s C_s & O_b C_b & O_s C_s & O_b
F1 = 0.36
MCC = 0.11
PT = 3
TT = 3
CT = 3
p_T = 1

(a)

(b)

Specifically, this domain values the timing of prediction errors more than their frequency. The key objectives are (i) achieving at least one correct prediction for each ‘Down’ or ‘Up’ movement, and (ii) avoiding premature closing signals for open positions. Other errors are tolerable when these two conditions are met. In real-world applications, probabilities associated with predictions also play a role, as they can determine the decision to enter or exit a position based on signal strength.

Figure 12, shows for each asset the values of p_T and MCC across probability thresholds (0.3, 0.5, 0.7, 0.9), revealing two consistent trends: (i) MCC and p_T decreases as thresholds increase (clearer at $H\Delta\tau \in \{10, 50\}$) and rises from less-liquid (ACM, DOGE, SOL) to highly-liquid (BTC and ETH) assets; (ii) at $H\Delta\tau = 5$, MCC and p_T increases with higher thresholds for the most liquid group (BTC and ETH), observed consistently across all models. This is well shown in Figure 13 because as the thresholds increase, the model will miss more extreme mid-price moves but for those predicted it will be more often accurate, as illustrated by the increasing precision and the decreasing recall. These patterns emphasize the importance of signal positioning; applying higher thresholds can disrupt the sequence, potentially improving classical metrics while impairing the model’s ability to manage positions effectively. Conversely, these findings underscore the impact of asset microstructural characteristics on signal reliability, with highly-liquid assets generally offering more automated trading feasibility than the less-liquid ones. The confusion matrices (Figure 9) further illustrate the error distributions affecting transaction management, where highly liquid securities exhibit reciprocal misclassifications in extreme classes, impacting transaction timing. Illiquid asset errors, however, tend to misclassify more extreme predictions as neutral, minimizing impacts on transaction management in relative terms compared to liquid securities.

In a closer analysis by liquidity group, Table 10 reveals that without thresholds for less liquid stocks at $H\Delta\tau = 5$, p_T separates assets into two groups: one for the most liquids (BTC, ETH) and one altcoin (DOGE) with a lower p_T, while the less liquid group (SOL, ACM) presents a higher p_T, consistently across all models. This separation, unseen in Figure 12, correlates somewhat with the microstructural characteristics displayed in Figure 7 with less extreme LOB depth within less liquid securities, making them better suited for deep learning inputs. This structural similarity, combined with the balanced class distribution observed at $H\Delta\tau = 5$, enhances model performance. Moving to $H\Delta\tau \in \{10, 50\}$, this effect diminishes due to more pronounced class imbalances. Across less-liquid assets, p_T quickly declines with thresholds above 0.5.

We can confirm that in Figure 13 with the view of the performances across the 3 classes, showing the contrast of evolution of metrics along probability threshold depending on the class and the security. For the example we pick the most and least liquid securities: BTC vs ACM. We observe the recall increasing across the two securities along the x-axis which is explained by the fact that if the prediction is either ‘Down’ or ‘Up’ but the probability threshold is not met for that class, the prediction defaults to ‘Stable’. Since our trading strategy does not act based on the class ‘Stable’, it is considered a neutral class and defaulting to that class does not impact our simulated trading strategy. At $H\Delta\tau = 5$, the hierarchy of performance inter-class is the same for BTC and ACM with the Stable class performing much better than the extreme class as well as the trend being consistent across all classes of the two securities, given the class imbalance at that time horizon. However, the trend changes direction as the horizon increases, such that at $H\Delta\tau = 50$ we observe a decreasing prediction power across the classes along the thresholds for both BTC and ACM aligning with MCC and p_T observed earlier, except for the ‘Stable’ class with ACM, again explained by the class imbalance at $H\Delta\tau = 50$. The illustrations for the other models are available in Appendix C.

At $H\Delta\tau \in \{10, 50\}$ the pattern observed for p_T is not the same while relatively consistent across models at each time horizon. Indeed, at $H\Delta\tau = 10$ the p_T values allow us to distinguish two groups, formed by (BTC, SOL) with overall higher p_T values, and (ETH, DOGE, ACM) with lower values. At $H\Delta\tau = 50$, the groups based on similar p_T values are less clear and less consistent: DOGE has clear higher values, the SOL and ETH, followed by BTC, and last ACM. In summary, we do not observe a consistent pattern across horizons regarding p_T, which is fine. The securities we chose present different liquidity aspects as illustrated in Figures 7,8 and 9 and are traded very differently. For these reasons it may make sense that a group of securities showing a high predictive power at $H\Delta\tau = 5$ is again showing the best performances at $H\Delta\tau = 50$.

Table 10: Metrics by Probability Threshold for model DeepLOB in a coarse-grained representation, across the three prediction horizons $H\Delta\tau \in \{5{:}\,(a), 10{:}\,(b), 50{:}\,(c)\}$, on Test set. The results are organized by prediction horizon, securities and metrics (p_T, MCC and F1).

H5												
	0.3			0.5			0.7			0.9		
Ticker	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1
BTC	0.61	0.60	0.58	0.58	0.52	0.53	0.54	0.57	0.74	0.74	0.76	0.80
ETH	0.60	0.59	0.59	0.64	0.62	0.62	0.63	0.64	0.77	0.78	0.78	0.79
SOL	0.67	0.67	0.71	0.76	0.51	0.51	0.51	0.50	0.70	0.70	0.70	0.69
DOGE	0.59	0.58	0.59	0.63	0.39	0.39	0.40	0.42	0.59	0.59	0.62	0.66
ACM	0.65	0.64	0.63	0.63	0.22	0.22	0.23	0.26	0.78	0.78	0.81	0.86

(a)

H10												
	0.3			0.5			0.7			0.9		
Ticker	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1
BTC	0.67	0.66	0.64	0.65	0.59	0.59	0.60	0.63	0.75	0.75	0.76	0.79
ETH	0.70	0.69	0.69	0.72	0.64	0.64	0.63	0.62	0.76	0.76	0.75	0.74
SOL	0.78	0.78	0.81	0.84	0.51	0.51	0.49	0.47	0.67	0.67	0.66	0.63
DOGE	0.69	0.68	0.69	0.72	0.41	0.41	0.42	0.42	0.59	0.59	0.61	0.62
ACM	0.70	0.70	0.70	0.69	0.34	0.34	0.35	0.36	0.80	0.80	0.82	0.84

(b)

H50												
	0.3			0.5			0.7			0.9		
Ticker	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1	p_T	MCC	F1
BTC	0.81	0.81	0.80	0.73	0.64	0.64	0.64	0.61	0.75	0.75	0.75	0.72
ETH	0.86	0.86	0.83	0.81	0.52	0.52	0.50	0.47	0.72	0.72	0.70	0.66
SOL	0.83	0.81	0.78	0.70	0.35	0.35	0.31	0.26	0.57	0.56	0.49	0.38
DOGE	0.86	0.84	0.78	0.74	0.34	0.34	0.31	0.29	0.59	0.59	0.56	0.50
ACM	0.79	0.76	0.72	0.67	0.32	0.32	0.32	0.32	0.56	0.57	0.60	0.61

(c)

Figure 12: Metrics by Probability Threshold for model DeepLOB. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys two key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) the pattern of average performance, along with its standard deviation, represented by the grey line and shaded areas. All average values and standard deviations are calculated across all assets, covering the testing period.

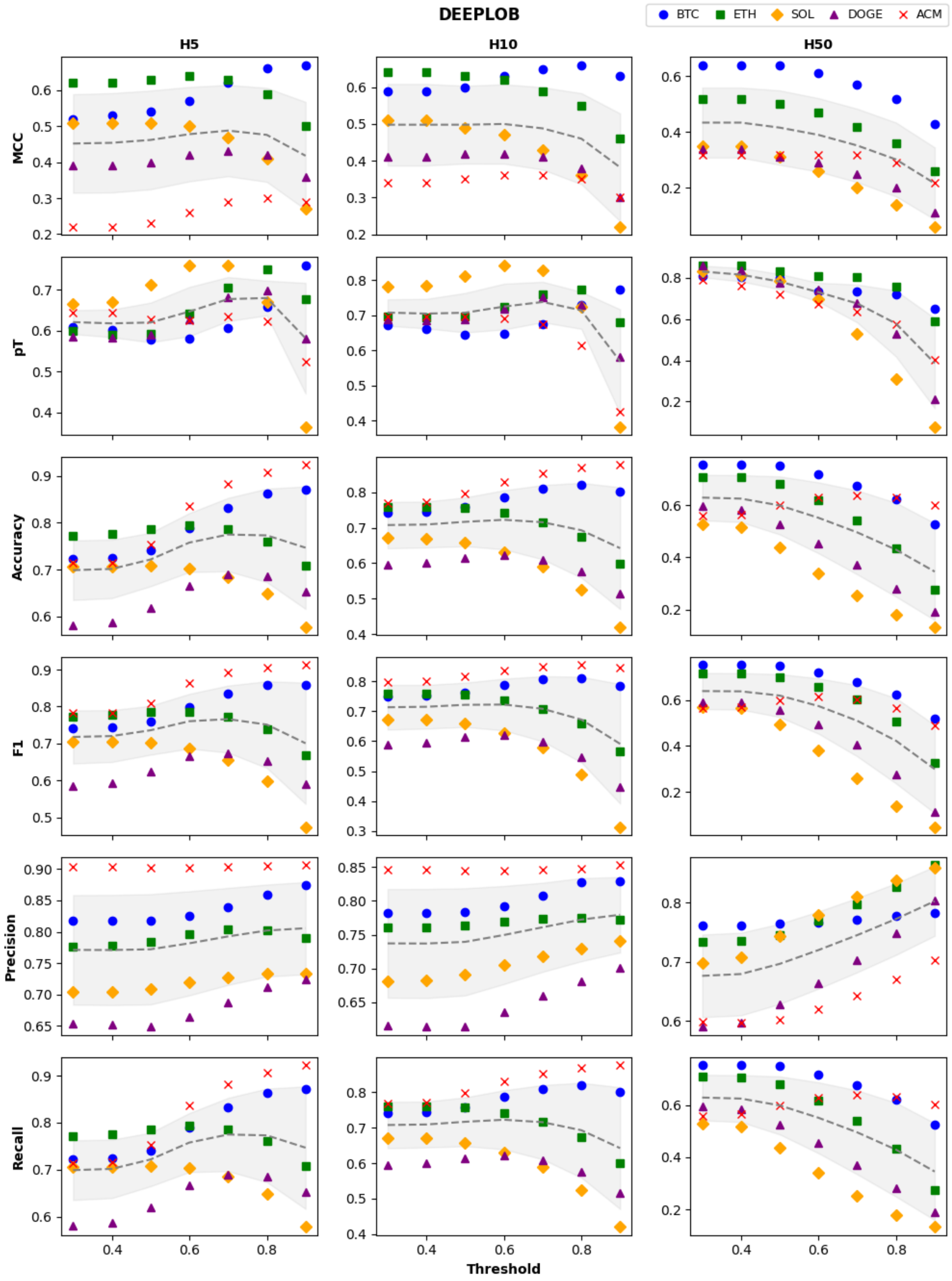
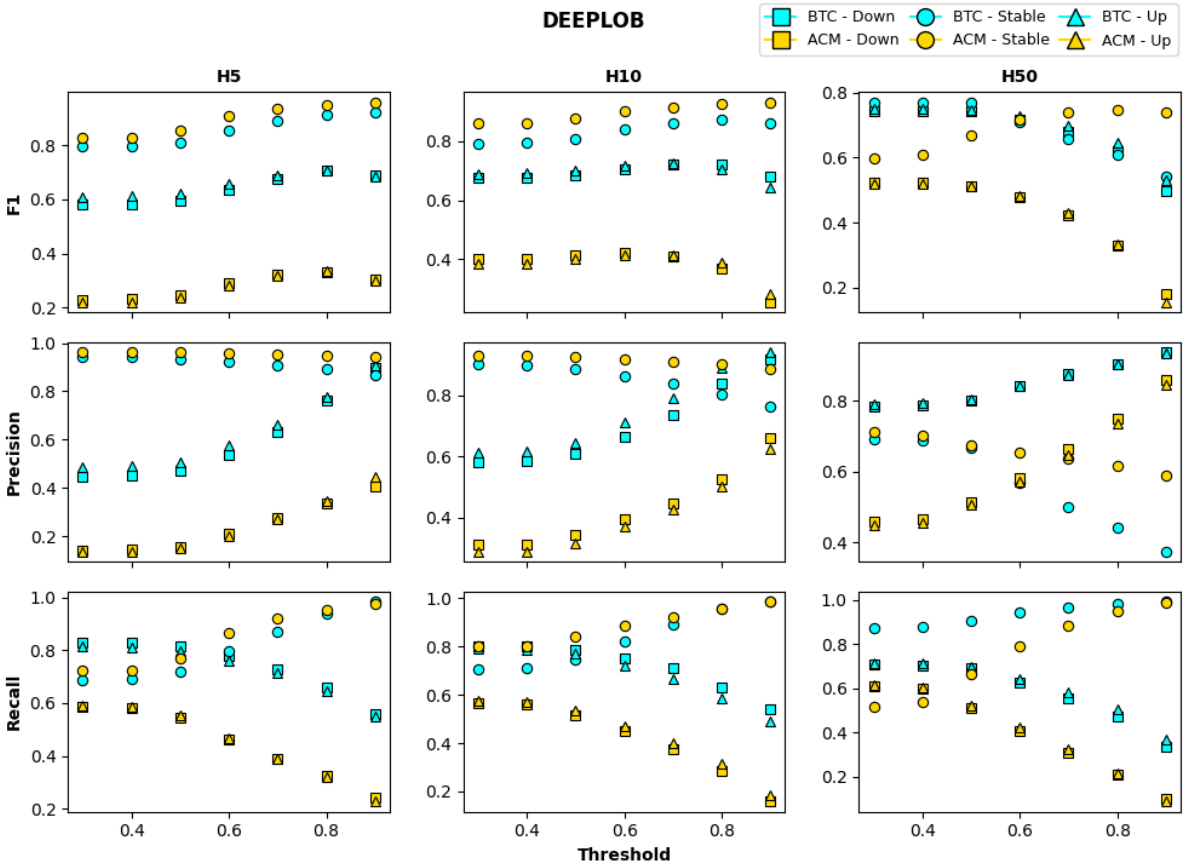


Figure 13: Metrics by classes along Probability Threshold for BTC and ACM applied on model DeepLOB. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys the following key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) across the different predicted classes ('Down', 'Stable', 'Up'); (iii) the similarities between assets at opposite sides of the liquidity spectrum.



6.3. Comparison of Model Performances based on F1 score, MCC and p_T.

We analyzed the predictive capabilities of different models— BinBTabl, BinCTabl, Transformer, and DeepLOB—in forecasting the direction of mid-price changes across the three prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. The overall metrics evaluated include (i) F1 score; (ii) Matthews Correlation Coefficient (MCC); and (iii) p_T (the probability of correctly executing a round-trip transaction). The results of these analyses are presented in Tables 11, 12 and 13, with the best-performing metrics highlighted in green. We compare the overall performances of the models based on the sum of these three metrics, as proposed by Briola et al. [8].

Table 11: Models' performances at $H\Delta\tau = 5$. For each deep learning architecture, we report three key metrics: (i) the F1 score; (ii) the MCC; and (iii) the p_T. For each asset, the best-performing model is highlighted in green. A model is considered superior if the sum of its three-performance metrics is the highest.

	H5											
	binctable			binbtable			transformer			deeplob		
Ticker	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T
BTC	**0.72**	**0.57**	0.59	0.71	**0.56**	0.59	0.67	0.53	0.60	0.66	0.52	**0.61**
ETH	**0.75**	**0.62**	**0.61**	0.74	0.60	0.60	0.74	0.61	**0.61**	**0.75**	**0.62**	0.60
SOL	0.67	0.49	0.63	0.66	0.48	0.64	0.66	0.48	0.65	**0.68**	**0.51**	**0.67**
DOGE	**0.58**	0.38	0.59	**0.58**	0.38	**0.60**	0.57	0.36	0.58	**0.58**	**0.39**	0.59
ACM	**0.45**	**0.23**	0.65	0.44	0.22	**0.69**	0.44	0.20	**0.69**	0.43	0.22	0.65

At the shortest prediction horizon, $H\Delta\tau = 5$, the performance of the models varies significantly across the different liquidity groups. For the most liquid cryptocurrencies BTC and ETH, the BinCTabl and DeepLOB models yield the best same performances. For DOGE table 11 shows balanced metrics, with BinBTabl and DeepLOB slightly on top. In contrast, the other altcoin SOL DeepLOB is the best model by a little relative margin across all metrics. Finally, for the least liquid ACM, BinCTabl yields the best F1 (0.45) and MCC (0.23) while BinBTabl shows similar values on those metrics and is ahead by a larger margin in p_T. Overall, summing the value of the three metrics, in this horizon BinCTabl stands out for the most liquid securities with BTC and ETH, then DeepLOB stands out for ETH and SOL (lower liquidity cluster), while BinBTabl is best suited for the lowest liquidity range with DOGE and ACM.

Table 12: Models' performances at $H\Delta\tau = 10$. For each deep learning architecture, we report three key metrics: (i) the F1 score; (ii) the MCC; and (iii) the p_T. For each asset, the best-performing model is highlighted in green. A model is considered superior if the sum of its three-performance metrics is the highest.

	H10											
	binctable			binbtable			transformer			deeplob		
Metric	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T
BTC	0.75	0.61	**0.68**	0.73	0.60	0.68	**0.76**	**0.63**	0.67	0.72	0.59	0.67
ETH	**0.75**	0.63	0.71	**0.75**	0.63	**0.72**	**0.75**	0.63	0.70	**0.75**	**0.64**	0.70
SOL	0.66	0.49	0.77	0.66	0.49	0.75	0.64	0.47	**0.78**	**0.67**	0.51	**0.78**
DOGE	**0.61**	**0.42**	**0.70**	0.60	0.41	**0.70**	0.59	0.38	0.67	0.60	0.41	0.69
ACM	0.51	0.30	0.69	0.50	0.29	**0.72**	0.52	0.29	0.67	**0.55**	**0.34**	0.70

In the medium horizon, $H\Delta\tau = 10$, Table 12 shows the transformer model achieves the highest combined score across F1, MCC, and p_T for BTC (F1 = 0.76, MCC = 0.63, p_T = 0.67). This highlights its stability and predictive power over mid-price changes. For ETH, BinBTabl achieves a very mild lead driven by consistently high F1, MCC and p_T values (F1 = 0.75, MCC = 0.63, p_T = 0.72). The altcoins SOL and DOGE have a different best model: the DeepLOB model comes out best for SOL thanks to the consistent lead across metrics offering a lead by a large relative margin over the other models in combined metrics terms. For DOGE, BinCTabl leads in aggregate metrics (F1 = 0.61, MCC = 0.42, p_T = 0.70), reflecting its capacity to provide balanced metrics across the board. ACM, the least liquid asset, reaches its highest aggregate with DeepLOB with a strong margin over the second-best model in F1

(0.04) and MCC (0.04), indicating superior performance and reliability with this model at this prediction horizon.

In summary, BinCTabl exhibit strong adaptability and resilience in predicting mid-price changes at the $H\Delta\tau = 10$ horizon by ending best only for DOGE but second best for all the other securities, while DeepLOB shows a significant edge in the least liquid bucket of assets (SOL and ACM).

Table 13: Models' performances at $H\Delta\tau = 50$. For each deep learning architecture, we report three key metrics: (i) the F1 score; (ii) the MCC; and (iii) the p_T. For each asset, the best-performing model is highlighted in green. A model is considered superior if the sum of its three-performance metrics is the highest.

	H50											
	binctable			binbtable			transformer			deeplob		
	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T	F1	MCC	p_T
BTC	0.75	0.64	0.82	0.76	0.64	**0.83**	**0.77**	**0.67**	**0.83**	0.75	0.64	0.81
ETH	0.64	0.51	**0.87**	0.63	0.50	**0.87**	0.65	0.51	0.84	**0.67**	**0.52**	0.86
SOL	**0.52**	0.36	**0.89**	**0.52**	**0.37**	**0.89**	0.50	0.35	0.83	0.50	0.35	0.83
DOGE	**0.56**	**0.40**	**0.91**	0.55	0.39	0.90	0.52	0.37	0.88	0.52	0.34	0.86
ACM	0.56	0.33	0.78	0.55	0.32	**0.81**	**0.57**	**0.35**	0.76	0.55	0.32	0.79

Looking at Table 13, we observe that, at $H\Delta\tau = 50$ for BTC, as it did at $H\Delta\tau = 10$, the Transformer is the best model, in this horizon by a large margin in combined metrics (0.05). For ETH, the DeepLOB model is this time the best-performing model (while ending close second at $H\Delta\tau = 10$), highlighting its stability and adaptability over extended horizons. For the altcoins SOL and DOGE, BinCTabl and BinBTabl achieve a similar stronger performance by a large margin over Transformer and DeepLOB, but they also perform second for the other currencies, providing the most robust models across the liquidity spectrum at this prediction horizon. Regarding ACM, however, the model performances are overall similar, with the Transformer only showing a small relative lead over the other models.

The p_T score highlights BinBTabl's strengths, achieving the best averaged value across all securities, which aligns with the ranking of models by p_T values recorded across other horizons. BinCTabl provides a similar lead in F1 score with the highest value among the most securities, across the three time horizons. The case is less obvious in MCC terms, where DeepLOB (and BinCTabl in a lesser fashion) achieves a lead at shorter time horizons, specifically at $H\Delta\tau = 5$.

In summary, the DeepLOB model demonstrates strong overall performance across horizons, excelling especially with ETH across horizons and performing overall well among altcoins (DOGE, SOL) in short time horizons, $H\Delta\tau \in \{5, 10\}$. Regarding BTC, it depends on the horizon: BinCTabl will be the best choice for short term horizons $- H\Delta\tau \in \{5, 10\} -$ while Transformer will perform best at longer terms $- H\Delta\tau \in \{10, 50\}$. In the case of altcoins and ACM, BinCTabl and BinBTabl are either the best across prediction horizons or at least the most stables arriving second-best, showcasing the reliability of those model architecture given a much smaller number of features.

6.4. Transfer Learning Performances based on F1 score, MCC and p_T.

As in Zhang et al. [5], we test the tractability of the models within the same prediction horizon for other securities, as mentioned in his paper, the so-called 'Transfer Learning', which in this case does not mean fine-tuning. Given the high cardinality of possibilities across models, security and prediction horizons, we will only show the cross-securities results for DeepLOB at $H\Delta\tau = 10$. We do not investigate in this paper the "Transfer Learning" across different time horizons.

Figure 14: probability of correctly executing a round-trip transaction for DeepLOB trained on a security and tested on the other securities

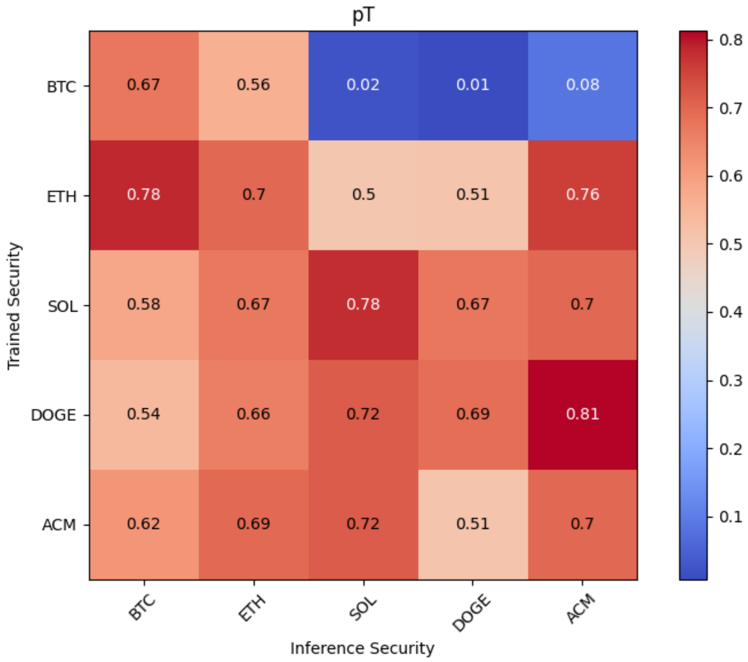


Figure 15: F1 score of DeepLOB trained on a security and tested on the other securities

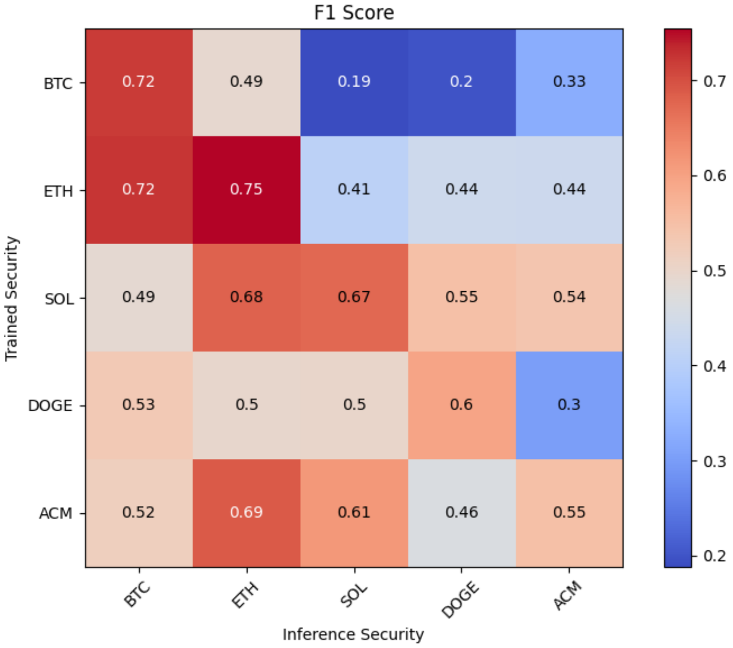
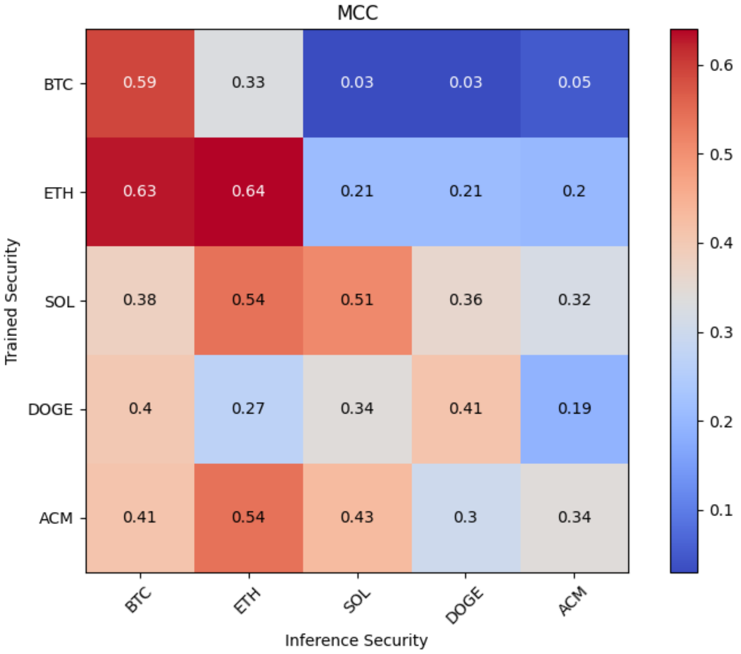


Figure 16: MCC of DeepLOB trained on a security and tested on the other securities



The tractability of the DeepLOB model at the $H\Delta\tau = 10$ prediction horizon, trained on one security and applied to others, reveals mixed results depending on the liquidity and microstructural characteristics of the securities involved. The performance metrics—F1-score, MCC, and p_T—indicate that the model's predictive power is highly sensitive to both the training security and the inference security.

Looking into the high-liquidity assets group (BTC, ETH): when the model is trained on BTC or ETH, the F1-scores remain relatively high, even when applied to other securities, demonstrating that the model generalizes well across high-liquidity pairs. For example, ETH-trained DeepLOB achieves F1-scores of 0.72 (BTC) while drops to 0.44 on DOGE for the next best (see Figure 15). Similarly, the MCC values for BTC-trained and ETH-trained models, which are equally strong compared to lower-liquidity pairs, further reflecting robust classification performance. This supports the hypothesis that high liquidity securities yield transferable learning due to consistent shared patterns in their respective LOB structures and dynamics, that are picked-up by the model.

Regarding the medium liquidity cryptos composed of the altcoins SOL and DOGE, the reasoning regarding the superior performance of transfer learning within a same liquidity cluster is less clear. DOGE shows the best cross-learning performances with BTC (MCC Figure 16) and to a lesser extent ETH (F1-score Figure 15) instead of SOL, and DOGE-trained tested on ACM is even better than tested on DOGE when measuring in p_T, showing that DOGE is a wildcard asset sharing some similarities with the high-liquidity groups and the lower liquidity altcoins (SOL, ACM). For the SOL-trained model, the same observation can be made across metrics. This suggests that the medium liquidity altcoins introduce some asset-specific variability that limits the model's ability to fully extrapolate to assets with markedly different trading dynamics.

Finally, when trained on the low-liquidity pair ACM, the model performs even better with ETH and SOL than on ACM itself (F1 and MCC, similar p_T), suggesting that although much less liquid, some patterns of the book structure and dynamics are transferable to the more liquid securities, when the reverse is not observed (comparing MCC between ACM-ETH and ETH-ACM Figure 16).

In summary, the cross-security matrixes highlight a clear cluster of consistent transferable patterns captured by the model within the high-liquidity cluster. Furthermore, we can observe a better generalization from lower liquidity assets towards higher liquidity assets, rather than the reverse. However, this interpretability of mapping results to our defined liquidity groups hits its limits regarding DOGE, SOL and ACM.

Overall, DeepLOB at $H\Delta\tau = 10$ demonstrates tractability when applied within liquidity-aligned securities but faces significant challenges when extrapolated across the liquidity spectrum. These results reinforce the necessity of liquidity cluster-based and for certain cases even asset-specific calibration for models leveraging LOB data. In production, the choices regarding clustering for calibration purposes may be influenced by other factors that are not part of this study (PnL in production, cardinality of the security universe, computing resources, etc).

7. Conclusion

This paper tackles the challenge of integrating Limit Order Book (LOB) microstructural analysis with predictive modeling. To achieve this, we collected high-quality LOB data for a diverse group of 5 crypto-currency pairs traded on BINANCE over 2023-2024, categorizing them by liquidity groups to establish quantitative benchmarks that differentiate less and more liquid pairs.

On the forecasting side, we introduce our innovative open-source framework designed to standardize the handling of crypto LOB data at a smaller scaler to allow interested parties with smaller resources (students, light hardware, etc.) the chance to apply it. Our framework needs to be run partly or fully on the platform of the cloud provider (SageMaker, Google Colab). It includes cutting-edge scientific methodologies for data transformation and processing, rapid and flexible training, validation, testing, and quality assessment through trading simulations. Its modular design ensures seamless integration of future models, enhancing adaptability for evolving advancements in LOB research. This work specifically extends a state-of-the-art model, DeepLOB, for multivariate time series forecasting, which is tailored for LOB dynamics. We also propose a refined labeling strategy that enhances the model's applicability for developing high-frequency trading strategies and establish an efficient, data-conservative pipeline to manage data imbalances. Model performance is evaluated using Accuracy, F1-score and the Matthews Correlation Coefficient (MCC) across three different prediction horizons (in terms of LOB updates) and at various confidence levels. Our findings indicate that forecasting accuracy is significantly affected by the liquidity of the asset, with higher liquidity pairs yielding stronger prediction signals and more consistent performance across horizons. Notably, as mid-price updates occur over shorter time intervals, we emphasize the critical role of low-latency hardware infrastructure for signal utility.

The study leverages the probability-based metric (p_T) introduced by Briola et al. [8] for evaluating model forecasts, offering a robust, assumption-free, and class-imbalance-immune alternative to traditional profit-and-loss (PnL) metrics by focusing on the probability of correct transaction execution while accounting for the chronological placement of prediction errors to better assess performance impact.

Finally we investigate the transferability of the learning from the DeepLOB model at $H\Delta\tau = 10$ and find strong similarities in results within the most liquid securities (BTC, ETH) and a generalization of learning that works better for illiquid to liquid assets than in the reverse order.

Overall, this paper presents a robust methodology and data pipeline that links the microstructural analysis of LOB data to predictive modeling, offering practical insights into a few representative crypto-currency pairs’ characteristics and microstructural factors influencing forecast performance. This study opens avenues for further research, particularly in cross-exchange validation and systematic testing of diverse deep learning models across the liquidity spectrum within the crypto-currency universe. Such analysis could reveal how specific model architectures can be optimized to handle challenges unique to sparser LOB structures, especially in the more liquid pairs. Future studies could explore the application of diffusion models, and graph-based models within this domain.

Bibliography

[1] Christine A. Parlour and Duane J. Seppi, “Limit order markets: A survey,” *Handbook of financial intermediation and banking*, vol. 5, pp. 63–95, 2008.

[2] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould. *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press, 2018.

[3] Eric Zivot and Jiahui Wang, “Vector autoregressive models for multivariate time series,” *Modeling Financial Time Series with S-PLUS®*, pp. 385–429, 2006.

[4] Adebiyi A. Ariyo, Adeyemi O. Adewumi, and Charles K. Ayo, “Stock price prediction using the ARIMA model,” in *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, pp. 106112.

[5] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

[6] Michael C. Münnix, Rudi Schäfer, Thomas Guhr. Impact of the tick-size on financial returns and correlations. *arXiv preprint arXiv:1001.5124v4*, 2010.

[7] Antonio Briola, Jeremy Turiel, and Tomaso Aste. Deep learning modeling of limit order book: A comparative perspective. *arXiv preprint arXiv:2007.07319*, 2020.

[8] Antonio Briola, Silvia Bartolucci, and Tomaso Aste. Deep limit order book forecasting. *arXiv preprint arXiv:2403.09267*, 2024.

[9] Antonio Briola, Silvia Bartolucci, and Tomaso Aste. HLOB: Information Persistence and Structure in Limit Order Books. *arXiv preprint arXiv:2405.18938*, 2024.

[10] Charles-Albert Lehalle and Sophie Laruelle. *Market microstructure in practice.* World Scientific, 2018.

[11] Konark Jain, Jean-Francois Muzy, Jonathan Kochems, and Emmanuel Bacry. No Tick-Size Too Small: A General Method for Modelling Small Tick Limit Order Books. *Available at SSRN 4983810*, 2024.

[12] Matteo Prata, Giuseppe Masi, Leonardo Berti, Viviana Arrigoni, Andrea Coletta, Irene Cannistraci, Svitlana Vyetrenko, Paola Velardi, and Novella Bartolini. Lob-based deep learning models for stock price trend prediction: A benchmark study. *arXiv preprint arXiv:2308.01915*, 2023.

[13] Matthew Dixon. Sequence classification of the limit order book using recurrent neural networks. *Journal of computational science*, 24:277–286, 2018.

[14] Antonio Briola, Jeremy Turiel, Riccardo Marcaccioli, Alvaro Cauderan, and Tomaso Aste. Deep reinforcement learning for active high frequency trading. *arXiv preprint arXiv:2101.07107*, 2021.

[15] Crypto Lake. Cryptocurrencies data provider. https://crypto-lake.com/. Accessed 09/12/2024

[16] Ioane Muni Toke. "Market making" behaviour in an order book model and its impact on the bid-ask spread. *arXiv preprint arXiv:1003.3796*, 2010.

[17] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*, 2022.

[18] Gabrielle La Spada, J. Doyne Farmer, and Fabrizio Lillo. Tick size and price diffusion. *arXiv preprint arXiv:1009.2329*, 2010

[19] Yufei Wu, Mahmoud Mahfouz, Daniele Magazzeni, and Manuela Veloso. Towards robust representation of limit orders books for deep learning models. *arXiv preprint arXiv:2110.05479*, 2021.

[20] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[21] Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData mining*, 10(1):35, 2017.

[22] Dat Thanh Tran, Alexandros Iosifidis, Juho Kanniainen, and Moncef Gabbouj. Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE transactions on neural networks and learning systems*, 30(5):1407–1418, 2018.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Appendix A. Additional Confusion Matrixes

Figure A.14: Confusion Matrices for model BinBTabl across the three prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. To create these summarized representations, we first calculate individual confusion matrices for each pair, each time horizon over the test set. These matrixes are subsequently normalized row-wise to transform raw counts into proportions, providing insight into predictive accuracy and class-specific performance. The final normalized matrix effectively visualizes the model's capability in classifying mid-price movement directions throughout the testing period.

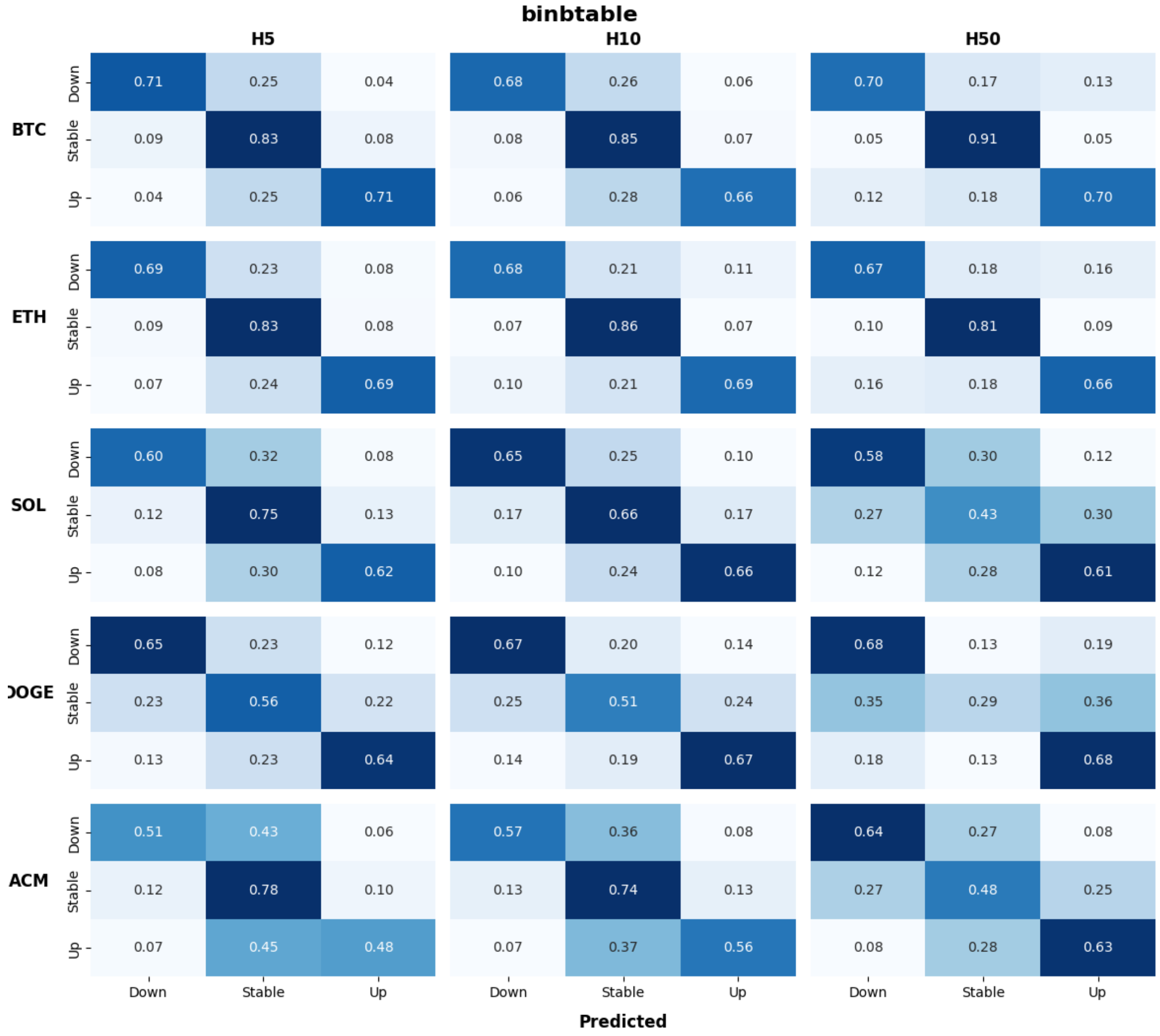


Figure A.15: Confusion Matrices for model BinCTabl across the three prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. To create these summarized representations, we first calculate individual confusion matrices for each pair, each time horizon over the test set. These matrixes are subsequently normalized row-wise to transform raw counts into proportions, providing insight into predictive accuracy and class-specific performance. The final normalized matrix effectively visualizes the model's capability in classifying mid-price movement directions throughout the testing period.

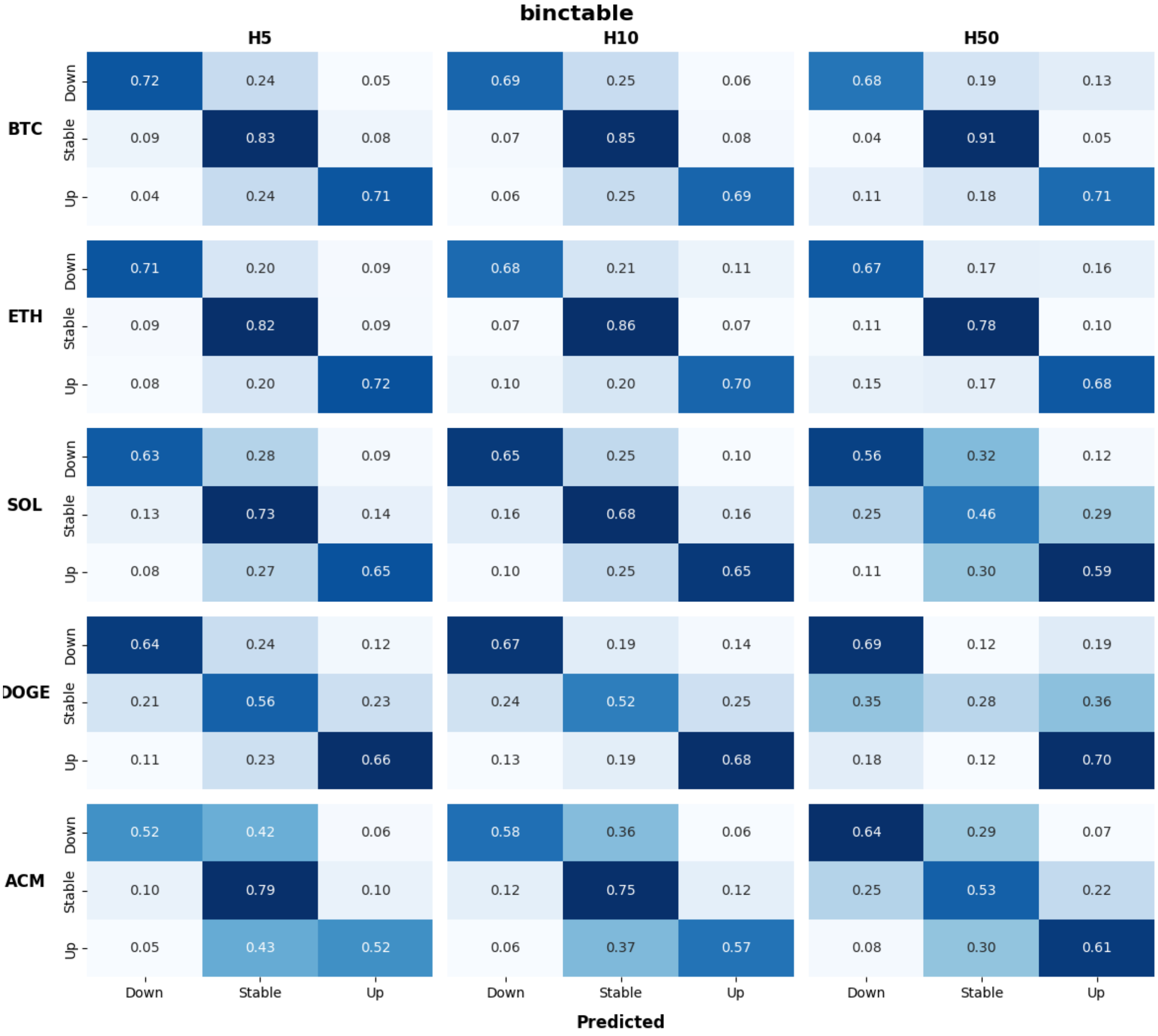
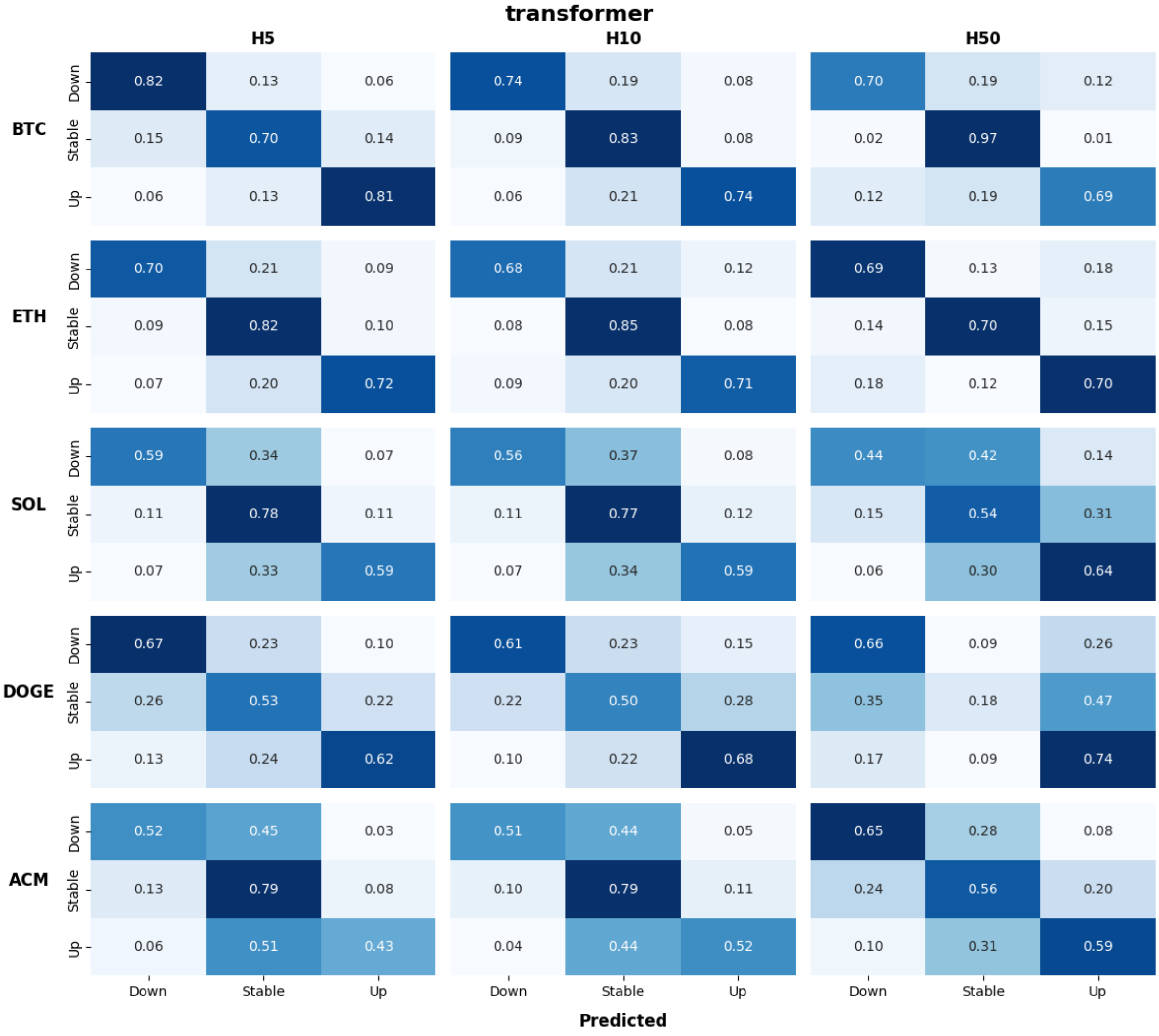


Figure A.16: Confusion Matrices for model Transformer across the three prediction horizons $H\Delta\tau \in \{5, 10, 50\}$. To create these summarized representations, we first calculate individual confusion matrices for each pair, each time horizon over the test set. These matrixes are subsequently normalized row-wise to transform raw counts into proportions, providing insight into predictive accuracy and class-specific performance. The final normalized matrix effectively visualizes the model's capability in classifying mid-price movement directions throughout the testing period.



Appendix B. Additional Analyses using Traditional Machine Learning Metrics (precision, recall, f1-score, accuracy, mcc) and p_T

Figure B.17: Trend of metrics along different probability thresholds for model BinBTabl. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys two key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) the pattern of average performance, along with its standard deviation, represented by the grey line and shaded areas. All average values and standard deviations are calculated across each asset, covering the entire analysis period.

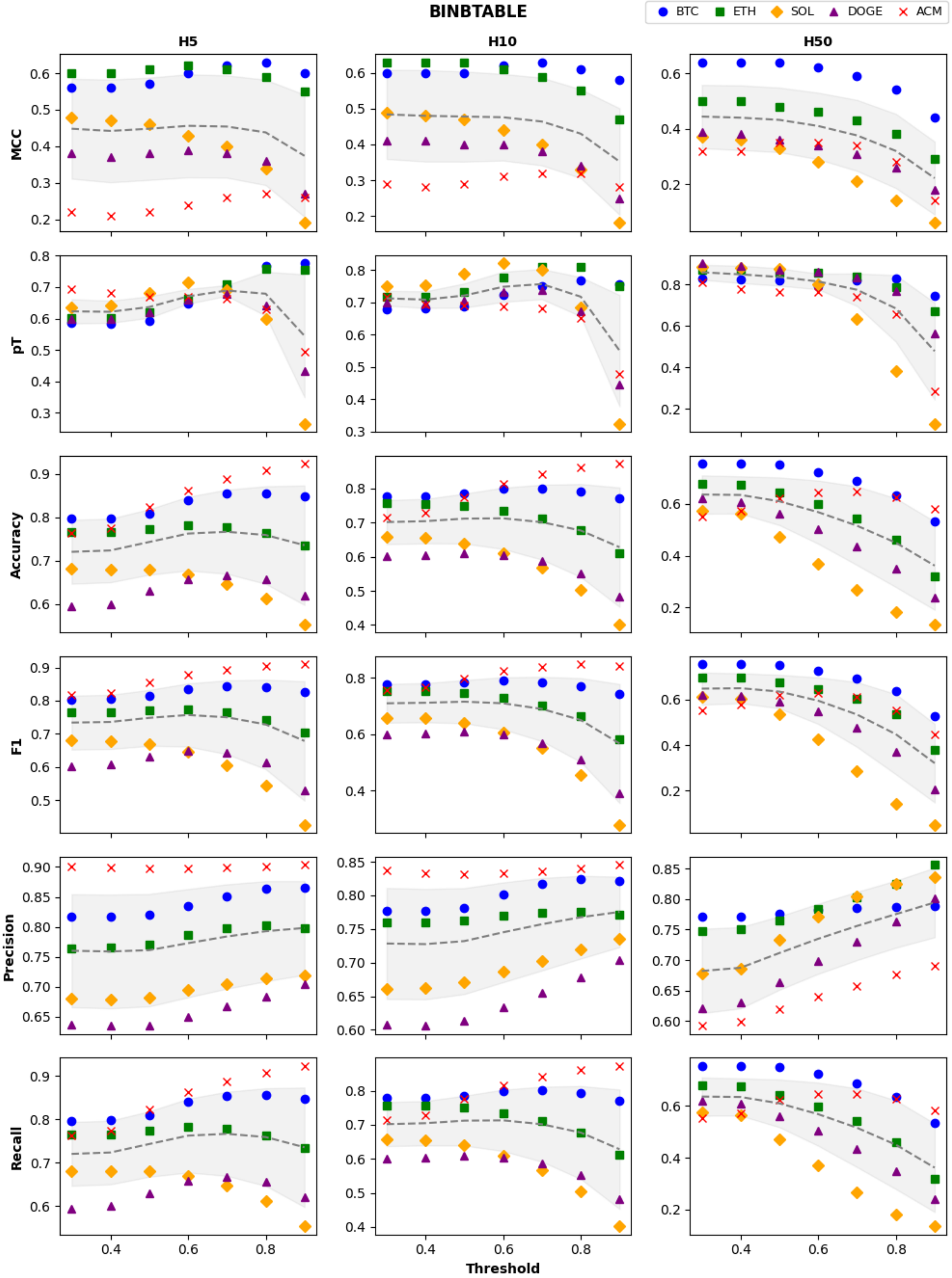


Figure B.18: Trend of metrics along different probability thresholds for BinCTabl. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys two key insights: (i)

model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) the pattern of average performance, along with its standard deviation, represented by the grey line and shaded areas. All average values and standard deviations are calculated across each asset, covering the entire analysis period.

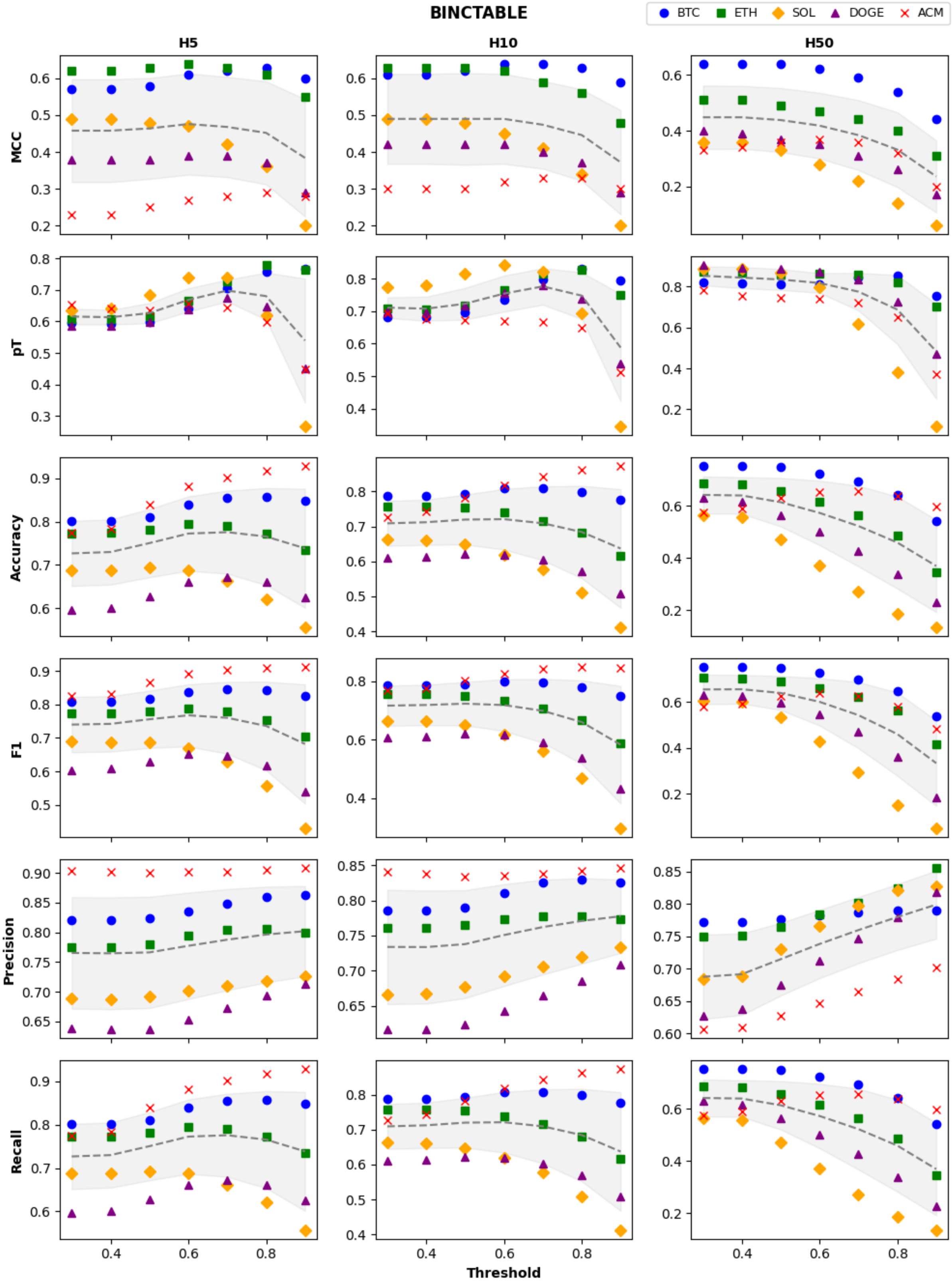
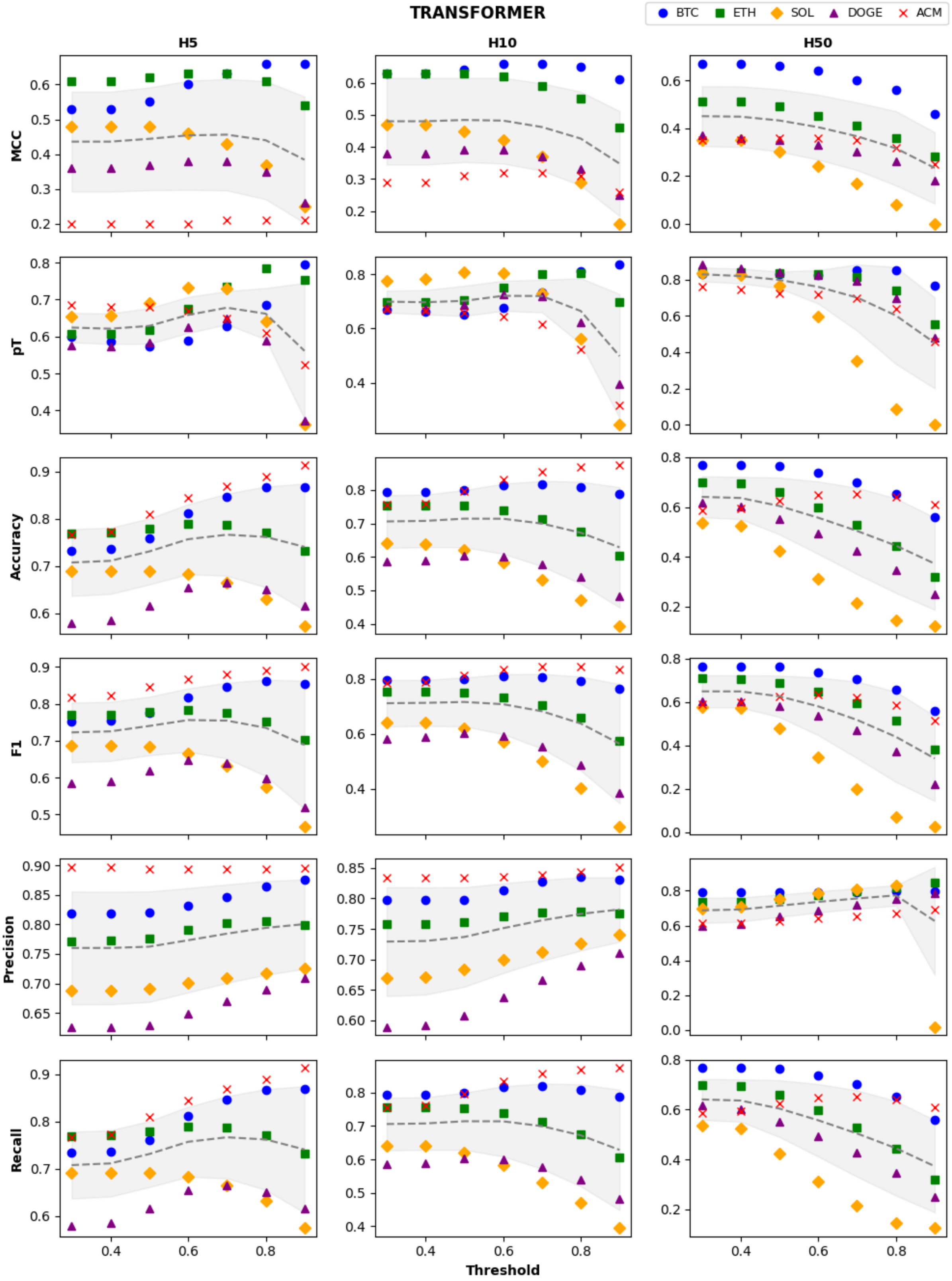


Figure B.19: Trend of metrics along different probability thresholds for Transformer. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys two key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) the pattern of average performance, along with its standard deviation, represented by the grey line and shaded areas. All average values and standard deviations are calculated across each asset, covering the entire analysis period.



Appendix C. Metrics by classes along Probability Threshold for BTC and ACM (f1-score, precision, recall)

Figure C.20: Metrics by classes along Probability Threshold for BTC and ACM applied on model BinBTabl. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys the following key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) across the different predicted classes ('Down', 'Stable', 'Up'); (iii) the similarities between assets at opposite sides of the liquidity spectrum.

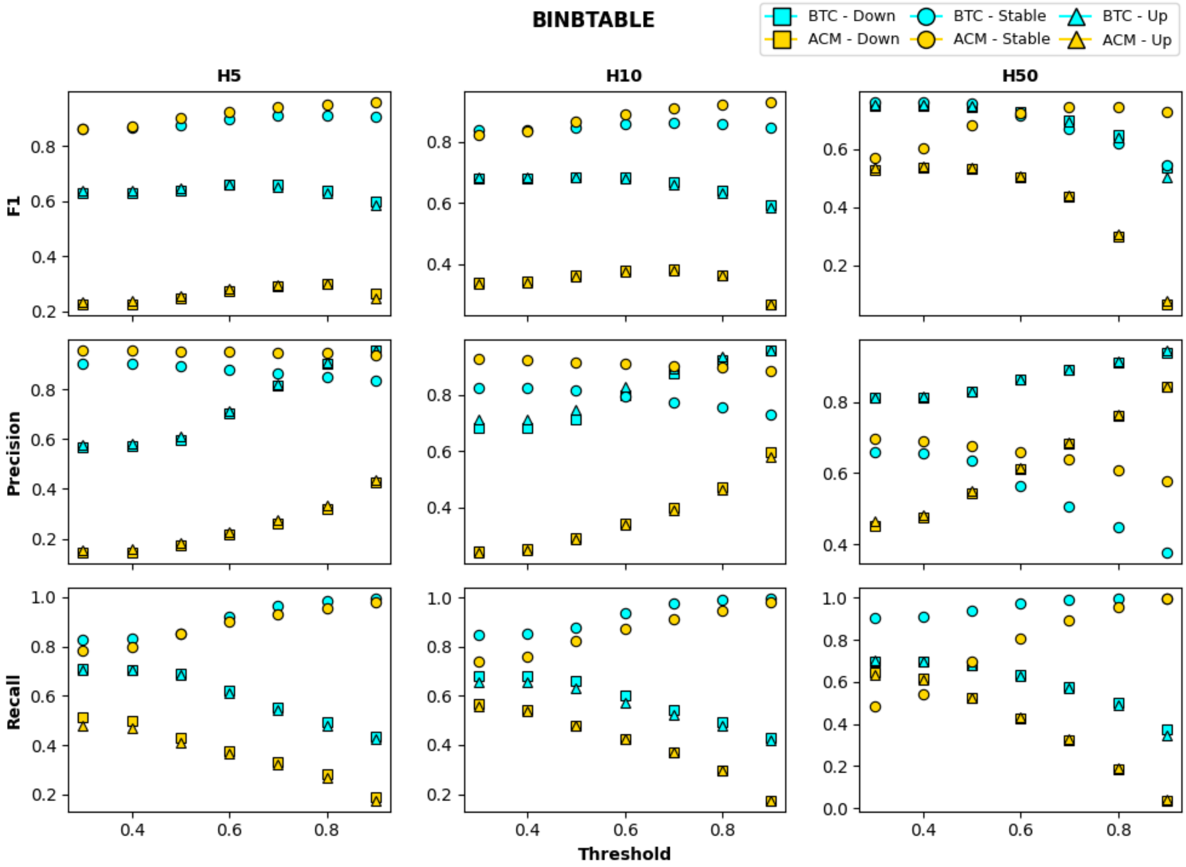


Figure C.21: Metrics by classes along Probability Threshold for BTC and ACM applied on model BinCTabl. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys the following key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) across the different predicted classes ('Down', 'Stable', 'Up'); (iii) the similarities between assets at opposite sides of the liquidity spectrum.

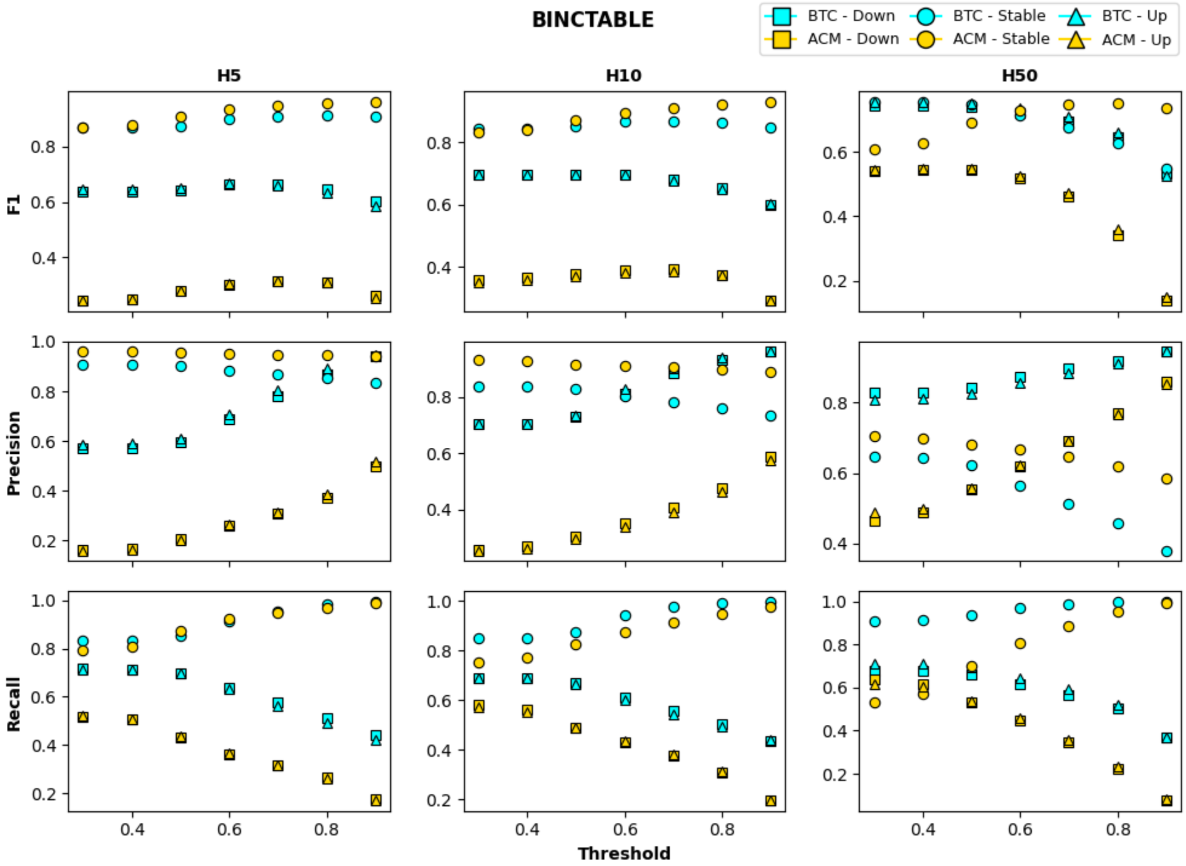
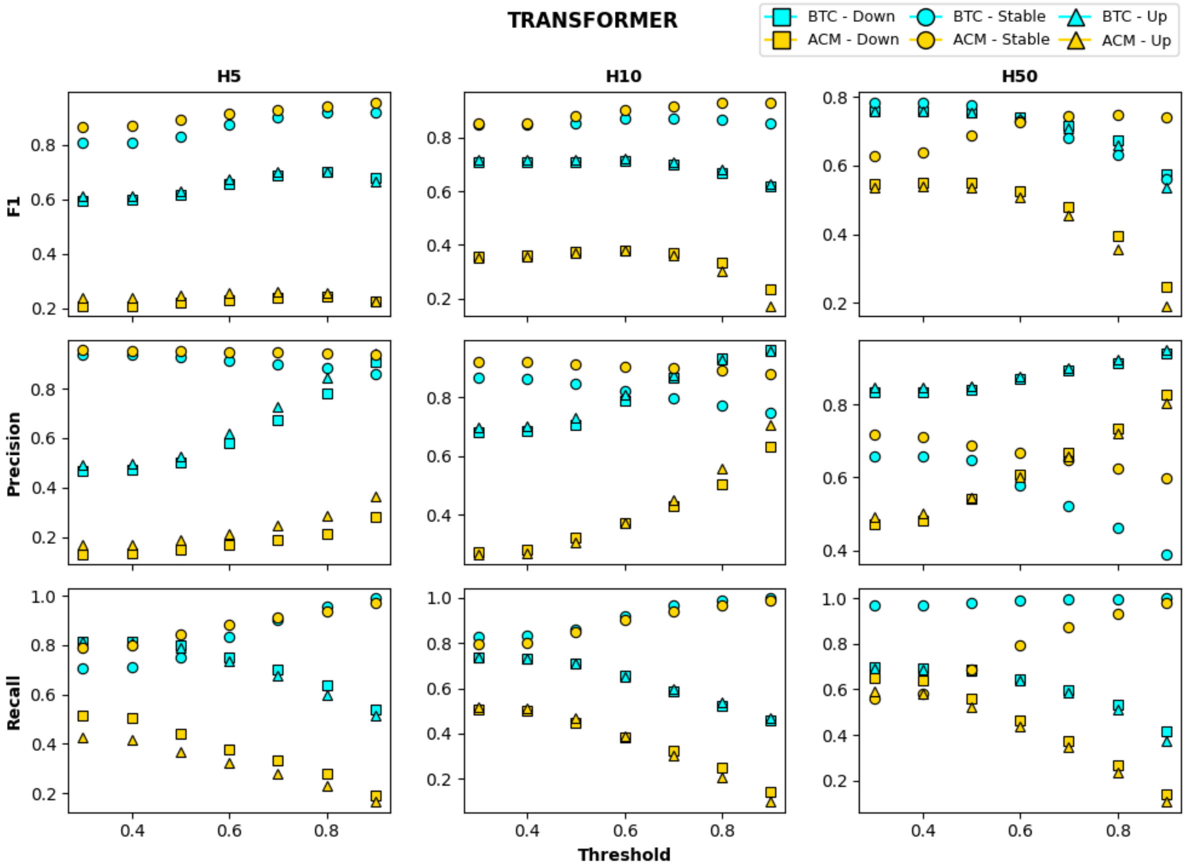


Figure C.22: Metrics by classes along Probability Threshold for BTC and ACM applied on model Transformer. The results are organized by prediction horizons (shown in columns) and metric (displayed in rows). Each plot conveys the following key insights: (i) model performance variations across different probability thresholds for forecasts (indicated at the bottom of the x-axis); (ii) across the different predicted classes (‘Down’, ‘Stable’, ‘Up’); (iii) the similarities between assets at opposite sides of the liquidity spectrum.


TRANSFORMER
BTC - Down
ACM - Down
BTC - Stable
ACM - Stable
BTC - Up
ACM - Up
H5
H10
H50
F1
Precision
Recall
Threshold