Impact of Liquidity Pool Size on Trading Volume in BTC-ETH Pools

Team #111
Matias Vizcaino (avizcaino3) — Walter Jack Simmons (wsimmons35) — MingShen Wang (mwang709) — Vítor de Matos Castilho (vcastilho3)

09 July 2023

Problem Statement and Objective

Decentralized Finance (DeFi), hosting about 48.78 billion USD in total value (as of April 23, 2023[1]), relies heavily on liquidity pools in decentralized exchanges like Uniswap for efficient crypto trading. These pools, using an automated market maker model, earn from trading fees distributed among liquidity providers based on their pool share.

This study primarily **aims to examine the influence of pool size on trading volume, specifically in BTC-ETH liquidity pools in exchanges like Uniswap**. It hypothesizes a direct correlation between pool size and trading volume, supported by existing studies.

This understanding could optimize liquidity provision, enhance DeFi market efficiency, and benefit stakeholders. For instance, liquidity providers could improve fee returns and risk management, traders could refine strategies, and DeFi platforms could develop more effective liquidity pools.

Uniswap Liquidity Pools: An Overview

Various DeFi platforms cater to different needs. Uniswap, a popular choice, provides a user-friendly interface and a broad spectrum of token pairs. Other platforms like Curve Finance and Balancer offer low-slippage trades and customizable pools, respectively.

Uniswap, a decentralized exchange (DEX) on Ethereum, supports trading of Ethereum-native and wrapped non-native assets. It uses a Constant Product Market Maker (CPMM) model in which liquidity providers deposit two tokens, keeping a constant product of reserves.

The CPMM formula is given by:

$$x \cdot y = k \quad \text{and} \quad price = \frac{y}{x}$$

Here, x and y denote the quantities of Token X and Y in the pool, and k is the constant product.

Uniswap operates without a central authority, ensuring privacy, fund control, and a diverse range of tokens. Centralized exchanges (CEX), though offering faster transaction speeds and better customer support, may pose security risks as they hold user funds.

Liquidity pools in DEXs enable direct interaction with smart contracts, ensuring continuous liquidity but exposing liquidity providers to impermanent loss risks. Prices are influenced by trade size, market conditions, supply-demand dynamics, and external price changes. Swaps and liquidity provision events, such as token minting and burning, significantly impact trading volumes. Indirect effects from network correlations and arbitrage opportunities can also influence trading volumes across pools.

Dataset

Inspired by the *"DeFi modeling and forecasting trading volume" (2023)*[2] paper, we sourced and constructed trade information for at least 6 months[3]. Initial extraction work has already been performed, and you can refer to our GitHub repository[4] for further details and access to the code. The dataset consists of data obtained from the following sources:

Source	Description	Data
Uniswap's The Graph API[5]	Provides transaction details, trading volumes, and block information from Uniswap v3 liquidity pools.	Transaction IDs, timestamps, amounts, USD equivalents, and other related data.
Etherscan API[6] for Uniswap transaction hashes	Used to extract corresponding transaction data from Etherscan based on transaction hashes.	Block hashes, block numbers, sender addresses, gas details, transaction hashes, and other relevant information.
Binance[7] CEX Data for ETHBTC	Daily zip trades downloaded using provided scripts from the Binance GitHub repository.	Detailed information about each trade executed on the Binance platform, including trade prices, quantities, timestamps, and buyer/seller characteristics.

[1] *"DefiLlama"* (2023), `https://defillama.com/`

[2] *"DeFi: modeling and forecasting trading volume on Uniswap v3 liquidity pools"* (2023), `https://ssrn.com/abstract=444535`

[3] *Data Extracted and Cleansed*, `https://drive.google.com/drive/folders/1y5ZwLZK9GQYsCNYSY--4VQMg8OdnuwuU?usp=sharing`

[4] GitHub repository: `https://github.gatech.edu/MGT-6203-Summer-2023-Canvas/Team-111/tree/main/Code`

Key Variables

Target Variable: Liquidity pool trading volumes (amountUSD) over specific blocks. We consider different models for multiple time horizons to study the relationship change over time.

Independent Variables: The dependent variables are derived from a set of features and are categorized as follows [Mostly completed, refer to progress section]:

1. Direct pool features (43): volatility, rate, number of trades, average trade size, total value locked (TVL).
2. Network spillover effects (8): trade flow imbalance.
3. CEX spillover effects (6): actual coin trade volume.
4. Price divergences between the 500 and 3000 pools (2).

We will aim to include as many variables as possible, given the time constraints of our project.

Approach and Progress

Overall, significant progress has been made in each stage of our approach. However, further analysis, evaluation, and optimization are required to successfully complete the project.

Throughout this process, we aim to maintain rigorous documentation of our methodologies and findings, which will allow us to ensure the transparency and replicability of our research.

Data Collection: Sourced data from multiple APIs (Uniswap, Binance) for liquidity pool sizes, trading volumes, and relevant variables. Data has been associated with Ethereum block numbers for consistent time measurement. Despite API rate limits, most of the data has been collected successfully.
Data Preprocessing: Cleaned, formatted, and addressed missing values in the dataset. Data consistency ensured and discrepancies from multiple sources addressed. Preprocessing completed with further examination of missing values in independent variables. Records with null values temporarily removed for continued analysis.
Feature Engineering: Constructed features to capture historical patterns, spillover effects, and price divergences utilizing a block-based time-series analysis framework. Feature engineering mostly complete, handling most direct pool features with finite lag. A review is planned to reduce the number of null values, include other feature categories, and perform feature selection.
Exploratory Data Analysis: Generated descriptive statistics, visualizations, and performed correlation analyses to uncover patterns, trends, and relationships between variables. In-depth exploratory data analysis will be performed once all features have been engineered, with various correlations and trends identified among the variables.
Model Selection and Development: Implemented a multivariate regression model (OLS) to investigate the pool size-trading volume relationship. Predicted target variables at different horizons with the application of lagged variables. OLS regression models have been developed and tested. Additional model testing and optimization in progress.
Model Evaluation and Optimization: Models evaluated using R-squared metric. Initial analyses indicate that model performance can be enhanced through further feature engineering/selection and reducing multicollinearity. Conducting correlation analysis and Variance Inflation Factor (VIF) for the final features.
Final Analysis and Conclusions: In the final stages of analysis. Commencing with drawing conclusions and communicating findings. Preparing to provide quantitative-driven insights.

Table 1: Approach and Progress

Challenges and Insights

Data collection and processing required significant effort due to the advanced engineering involved. Notably, we faced challenges with the rate limits of the Etherscan free-tier API when gathering data via Python API calls.

We've adopted two time concepts to analyze liquidity pool dynamics[8]. The "trading clock", defined by a block containing a mint operation, captures data temporal dynamics. A "time horizon" concept was introduced, defined every ten blocks up to the next mint operation, providing insights into the predictive power of independent variables over different timeframes.

Combined, these concepts allow a comprehensive exploration of short and long-term trends, informing decision-making and risk management in the DeFi ecosystem. Considering Uniswap operates on the Ethereum blockchain with a 14 seconds block time, these insights are invaluable.

The complexities of liquidity pool mathematics[9][10] presented hurdles, but we've captured the formulas to our best understanding in calculations.py functions. Given more time, we aim to refine these for enhanced feature derivation.

[8] *"DeFi: modeling and forecasting trading volume on Uniswap v3 liquidity pools"* (2023), `https://ssrn.com/abstract=444535`

[9] *"Liquidity Math in Uniswap V3"* (2021), `https://atiselsts.github.io/pdfs/uniswap-v3-liquidity-math.pdf`

[10] *"A Primer on Uniswap v3 Math: As Easy As 1, 2, v3"* (2023), `https://blog.uniswap.org/uniswap-v3-math-primer`

Progress

Engineering

The primary focus has been on the design of the Target Variable and determining its temporal horizons. Simultaneously, we addressed the challenge of feature engineering for the Direct pool features with finite lag. Despite the complexity, we are confident in meeting the remaining objec

In terms of merging Uniswap transactions (txt) with Etherscan, we've observed a high match rate of transactions for the analysis period of six months. The transaction breakdown is as follows:

- Uniswap txs: ((500, 'burns'), 4000), ((500, 'mints'), 3406), ((500, 'swaps'), 217821), ((3000, 'burns'), 5918), ((3000, 'mints'), 5046), ((3000, 'swaps'), 48592)
- Etherscan txs: ((500, 'burns'), 4000), ((500, 'mints'), 3242), ((500, 'swaps'), 217821), ((3000, 'burns'), 5918), ((3000, 'mints'), 4986), ((3000, 'swaps'), 48592)

From the transaction data, we created 8,227 reference blocks that correspond to each mint operation, excluding one. These blocks were then used to engineer features to calculate metrics for the same pool and other pools. These features capture data about liquidity pools and calculate metrics such as volatility, traded volume rate, trades count, and average volume. To expand the analysis, we have also included lagged features for the previous three mint operations. Table 2 serves as our block reference table.

Before initiating the modeling phase, we need to address records with missing values in the independent variables. Notably, some null values have been identified in the volatility (same pool) and avg-USD/rate-USD (other pool) metrics. We have opted to remove these records temporarily to proceed with the analysis, with plans to reassess the calculations and mitigate the occurrence of null values likely resulting from lagged variables.

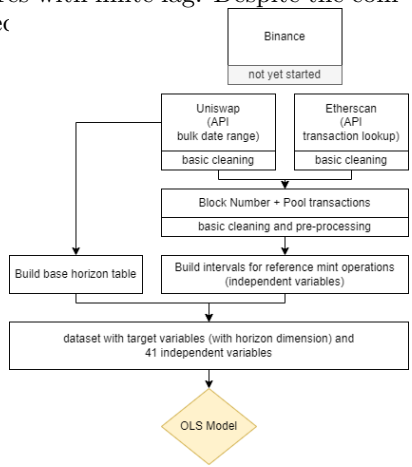

Figure 1: Data Engineering

pool	blockNumber	blockNumberChain	other_blockNumberChain
500	14498564	[14498564, nan, nan, nan]	[14498564, nan, nan, nan]
500	14498699	[14498699, 14498564, nan, nan]	[14498699, nan, nan, nan]
500	14499597	[14499597, 14498699, 14498564, nan]	[14499597, 14499560, 14499457, 14499198]
500	14499836	[14499836, 14499597, 14498699, 14498564]	[14499836, 14499560, 14499457, 14499198]
500	14500355	[14500355, 14499836, 14499597, 14498699]	[14500355, 14500043, 14499560, 14499457]
...	...	...	...
3000	15648981	[15648981, 15648330, 15648305, 15648187]	[15648981, 15648887, 15648536, 15646933]
3000	15649246	[15649246, 15648981, 15648330, 15648305]	[15649246, 15649243, 15648887, 15648536]
3000	15649545	[15649545, 15649246, 15648981, 15648330]	[15649545, 15649522, 15649347, 15649269]
3000	15649565	[15649565, 15649545, 15649246, 15648981]	[15649565, 15649522, 15649347, 15649269]
3000	15649578	[15649578, 15649565, 15649545, 15649246]	[15649578, 15649522, 15649347, 15649269]

Table 2: BlockNumber Chains

Finally, we've constructed a base table 3 with the `start_blockNumber` of each horizon and the `reference_blockNumber`, defined by mint operations in the block. This base table is joined with the mint aggregated transaction data to generate our target variables and independent variables that are lagged.

blockNumber	min_flag	reference_blockNumber	horizon_label	cum_volume_500
108757	0	15552674	9	423485.346309
108758	0	15552674	10	423485.346309
108759	1	15552772	1	328338.732259
108760	0	15552772	2	406084.780730
108761	0	15552772	3	536640.714920
...	...	...	...	...
109047	0	15555464	12	122730.731534
109048	0	15555464	13	123650.594764

Table 3: Reference and Horizon blocksn

Modelling

We have commenced the development of a framework for our prediction models and have constructed several Proof of Concept (PoC) models. For the modeling phase, we're employing an Ordinary Least Squares (OLS) regression approach to predict multiple target variables across different horizons using a set of independent lagged variables for each reference mint. This process involves iterating over the horizons, generating data subsets, fitting an OLS model for each horizon, and retrieving the R-squared value as a performance indicator. This approach allows for efficient and consistent prediction across various horizons.

$$\text{cum\_volume\_500}_{\text{horizon}} = \beta_0 + \beta_1 \cdot \widetilde{01}X + \beta_2 \cdot \widetilde{12}X + \beta_3 \cdot \widetilde{23}X + \ldots + \epsilon$$

In this formula, β_0, β_1, β_2, β_3, etc., represent the coefficients associated with each spot lagged variable, while ϵ denotes the error term or residual.

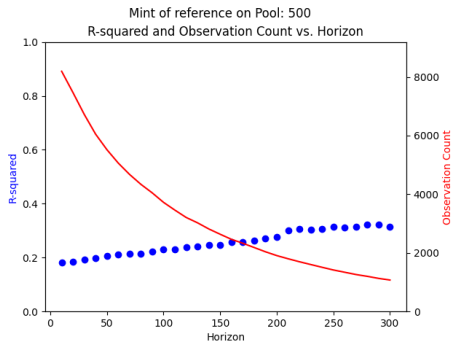

Figure 2: R2 best fit horizon

Observations indicate that around the horizon 20 (200 blocks, approximately 13 minutes), the R2 value ceases to increase significantly. Preliminary analysis of the relevant model (at horizon 20) reveals that the OLS regression model explains approximately 27.6% of the variation in the dependent variable cum_volume_3000_ref500. The significant variables, such as rate-count-isame_01, rate-count-isame_12, rate-count-isame_23, and wlother_3, have a considerable impact on the dependent variable. However, the potential collinearity among the independent variables could affect the interpretation of individual coefficients, suggesting a need for further analysis.

Model selection will be based on the optimal combination of features for each horizon. As we refine the feature and model development, considerations like autocorrelation, endogeneity, structural breaks, or multicollinearity will be addressed. These will be managed through additional sensitivity analyses, cross-validations, and the application of appropriate statistical techniques to diagnose and mitigate any complexities.

Going Forward

To summarize, the learnings from this project, particularly regarding data engineering and feature generation, are substantial. This largely unexplored field of financial technology presents vast opportunities for fresh findings and contributions to the existing body of knowledge. Our initial results are promising, and we're eager to delve into the forthcoming phases of our project, emphasizing model optimization and refining our analysis and conclusions.

Our future steps in the modeling phase include:

- Dataset Splitting: Partitioning the dataset into training and test sets, performing model metrics analysis, and interpretation.
- Analysis Scope: Examining both Pool 500 and Pool 3000, with a focus on target variables related to volume on the same pool as the reference mint, the other pool, or both.
- Experimentation: Identifying the optimal combinations of features, horizons, and target variables for accurate predictions.
- Feature Selection: Using techniques like step-wise or Principal Component Analysis to reduce feature complexity, limit overfitting, and improve interpretability.

Our planned experiments include testing different lag lengths, incorporating quadratic features or interaction terms, and accounting for structural breaks, such as the Terra-Luna collapse (May 2022) and the Merge (Sep 2022).

We aspire here or on further work to drive a comprehensive analysis and quantitative discussion on:

1. The relationship between liquidity pool size and trading volume in BTC-ETH liquidity pools.
2. The influence of the liquidity pool size on the slippage in BTC-ETH trading pairs, considering CEX spillover effects.
3. Impact of BTC-ETH price volatility on trading volume relative to liquidity pool size, focusing on price divergence.
4. Specific periods/events that significantly affect the relationship between BTC-ETH liquidity pool size and trading volume.