A Comparative Analysis of LSTM and Transformer Models for High-Frequency Cryptocurrency Pairs Trading on the Binance Exchange

Author : Anh-Tuan NGUYEN

Supervisor : Johnson EZE

MSc Computer Science with AI

Department of Computer Science

University of York

August 2024

Executive Summary Wordcount : 423

Main Text Wordcount : 8996

Module Name : Independent Research Project

Module Code : COM00151M

Acknowledgements

I would like to express my heartfelt thanks to my supervisor, Dr. Johnson Eze, for his invaluable guidance throughout the project. I am also profoundly grateful to my family for their unwavering support and encouragement over the past two years.

Executive Summary

The aim of this research is twofold: first, to compare the predictive capabilities of two vanilla Deep Learning models (DL), Long Short-Term Memory (LSTM) and Transformer, specifically focusing on high-frequency (HF) cryptocurrencies forming cointegrated pairs over the period from 01/01/2023 to 23/01/2023. Second, leveraging the optimal DL model, the study examines the profitability of different AI-based algorithmic pair trading (PT) strategies against traditional rule-based algorithmic PT strategies within a back-testing framework, with an emphasis on identifying the conditions under which the AI-based strategies can outperform the traditional ones.

The motivation for this work arises primarily from the high volatility observed in cryptocurrency markets compared to other financial markets, offering a unique opportunity to investigate the statistical characteristics of HF cointegrated cryptocurrencies. Additionally, while LSTM models have demonstrated success in PT applications, there is limited literature on the integration of Transformers into algorithmic PT, highlighting the need for further exploration in this area.

From an ethical perspective, this research does not involve any human or qualitative data. However, integrating complex ML models into trading decisions raises accountability issues. These models can carry biases that may influence outcomes against certain cryptocurrencies. Additionally, when combined with algorithmic trading, ML models can enhance opportunities for market manipulation. For individuals without access to advanced ML-based algorithmic trading technology, this creates economic inequalities.

The methodology follows a systematic approach involving several steps. First, the most liquid pairs are identified using the monetary volume. Then, cointegrated pairs are determined in the in-sample period and evaluated in the out-of-sample period. Based on the actual prices of the coins forming the pairs over the respective in-samples and out-of-samples, two vanilla LSTM and Transformer models are trained and evaluated to ensure acceptable processing time. Vanilla Transformers were found to significantly outperform vanilla LSTM models in terms of processing time and errors and were used to predict prices, from which the predicted spreads were calculated and compared to the actual spreads. The pairs that exhibit low prediction errors and high correlation and cointegration relationships in the predicted spreads are elected for PT.

The predicted and actual spreads were then used in algorithmic PT strategies using the rolling Z-score and different rolling windows, resulting in multiple PT strategies. The study found that the financial performance of Transformer-based PT significantly underperformed compared to rule-based algorithmic PT when evaluated using the Sharpe ratio, annualised return, and win ratio. This aligns with the fact that Transformer-based PT is significantly riskier, involving a higher number of trades, increased fees, shorter average duration, and a greater maximum drawdown.

Table of Contents

List of Tables

List of Figures

1. Introduction/Focus of Research

1.1. Research Topic

Assuming the Cryptocurrency Markets (CMs) can be inefficient based on the current literature, the current paper (CP) aims to explore the area of Financial Data Science within the field of High-Frequency Pair Trading.

The objectives are twofold: first, to evaluate the predictive capabilities of two vanilla Deep Learning (DL) models, Long Short-Term Memory (LSTM) and Transformer on CM prices; second, to compare the profitability of executing a pair trading (PT) strategy using AI-predicted signals versus rule-based algorithmic PT strategy (RB-APTS) in a back-testing environment.

1.2. Project Topic : Justification and Motivation

CMs have grown rapidly, with global trading volume expected to exceed $108 trillion by 2024[1]. This surge highlights the increasing importance of cryptocurrencies in the global financial system, affecting liquidity, market dynamics, and cross-border capital flows[2]. With over 560 million users worldwide[3], digital assets are now a key component of investment portfolios. A significant portion of crypto trading is driven by algorithmic strategies[4], which aim to exploit market inefficiencies. The CP aims to contribute by examining whether AI-based algorithmic PT strategies (AI-APTS) can reliably replace traditional RB-APTS.

Given the higher volatility[5] in CMs compared to other financial markets, it investigates the statistical characteristics of high-frequency cointegrated cryptocurrencies and explores how DL can enhance PT performance. Unlike most literature, which compares AI-based algorithmic trading against market performance, typically a buy-and-hold strategy, to challenge the Efficient Market Hypothesis (EMH), the CP focuses on direct comparisons with traditional algorithmic trading.

This paper focuses on intra-day trading, considered riskier than daily trading but offering more profit opportunities due to frequent market inefficiencies. While HFT typically involves millisecond or nanosecond transactions, this study uses minute-interval prices, assuming that minutes are representative of HFT.

The PT strategy is chosen because of its market-neutral stance[6], meaning that the predictions made by the machine learning(ML) model should remain unaffected by prevailing market conditions.

In this study, two DL models are being challenged. While LSTM networks have been effective in previous studies on account of their ability to capture time-dependent patterns[7], Transformer models, originally designed for natural language processing (NLP) tasks[8], have recently demonstrated potential in the forecasting of time-series data. This is attributed to their capacity to handle long relationships using self-attention mechanism and parallel processing[9, 10], making them more scalable compared to LSTMs. Consequently, it makes sense to use LSTM as a baseline to assess the performance of Transformers.

Predictions will be generated every minute for a 1-minute forecasting window, with order execution delayed by 1 minute to account for processing and execution time. It is assumed that execution does not experience any slippage, ensuring the full benefit of an effective price prediction. This anticipative approach contrasts with conventional RB-APTS, where trading signals are generated instantly once actual prices are known, often leading to slippage[11](Figure.1) or non-execution due to the high volume of concurrent orders.

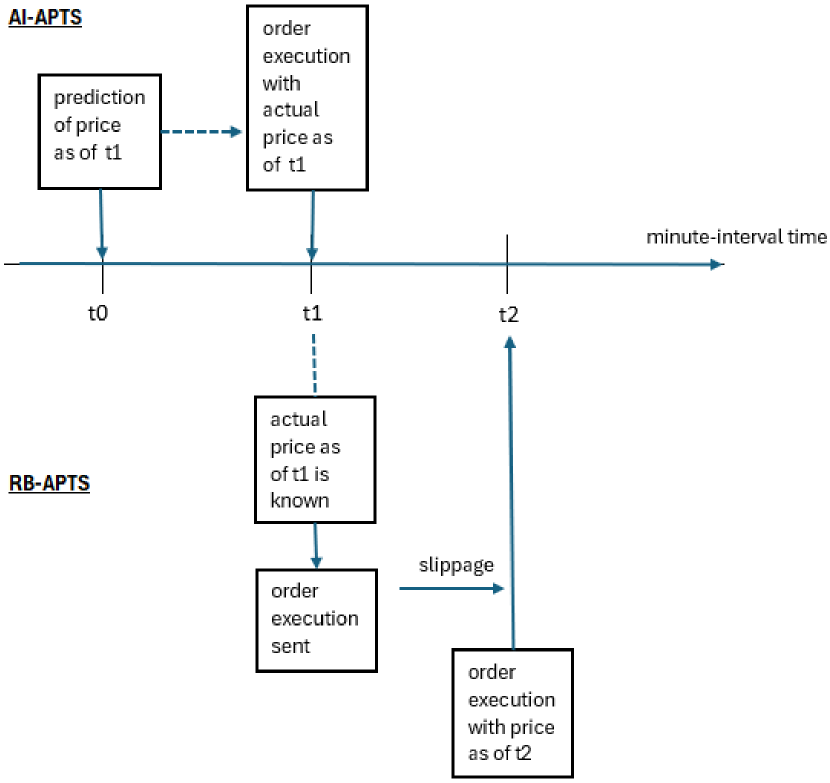

Figure.1 : Experiment's assumptions

1.3. Research Questions

This research aims to address the following questions:

- How do Transformer models compare to LSTMs in forecasting high-frequency cryptocurrency prices?
    - Null hypothesis-1 ($H_0 - 1$) : LSTM models cannot be outperformed by Transformer models.

- Alternative hypothesis-1 ($H_a - 1$) : Transformer models outperform LSTM, in a standard setting in terms of processing time and error metrics.

- How can AI-APTS outperform RB-APTS?
  - $H_0 - 2$: RB-APTS cannot be outperformed by AI-APTS.
  - $H_a - 2$: AI-APTS outperforms RB-APTS in terms of Sharpe ratio (SR), annualised returns (AR), and win ratio (WR), assuming order execution for AI-APTS occurs without slippage, unlike RB-APTS.

1.4. Aims and Objectives

The project aims to achieve the following goals:

- Identify cointegrated pairs over the study period (SP).
- Develop LSTM and Transformer models to predict prices of coins forming these pairs.
- Create an AI-based back-testing environment to generate PT signals based on predicted prices, simulate transactions, and evaluate profitability.
- Create a rule-based back-testing environment to generate PT signals from actual prices, simulate transactions, and evaluate profitability.

Specific objectives include:

- Reviewing related works to understand various approaches.
- Collecting market data for the SP.
- Identifying the most liquid coins.
- Identifying the most cointegrated pairs from the list of highly liquid coins.

- Building, training, and evaluating the 2 DL models associated with the coins in the pairs.
- Generating price predictions with the optimal DL model.
- Generating AI-APTS trading signals, simulating transactions, and evaluating financial results.
- Generating RB-APTS trading signals, simulating transactions, and evaluating financial results.
- Comparing the results between the two algorithmic trading strategies.

2. Literature Review

2.1. Cointegration method and PT strategy

PT[12] is a form of statistical arbitrage (SA)[13] that targets 2 closely related assets. According to mean-reversion theory, the difference in price between these assets will eventually return to its mean after deviating. Investors purchase the underperforming asset and sell the outperforming one, expecting their prices to converge to the historical mean(Figure.22). While PT is a well-known strategy for equities, its effectiveness is less established in the highly volatile CMs[14].

Figure.2 : PT strategy (Illustration of PT from[15])

Two papers offer conclusions that justify the choices taken by the CP.

In [16], the paper concludes that PT in CMs can yield significant profits, consistently surpassing traditional buy-and-hold strategies. This is attributed to the market inefficiencies and temporary mispricing of crypto-coins. It focuses on four well-known methods: correlation analysis[17], distance method (DM)[18], stochastic return differential approach, and cointegration analysis (CA)[19]. The study covers the period from January 2018 to December 2019, using closing prices of 4 major cryptocurrencies. Although the findings cannot be generalised to all major cryptocurrencies, the paper presents arguments supporting the use of pairs trading, that are endorsed by the CP. Furthermore, the CP extends this approach to encompass the top 25th quantile of most liquid coins over the SP.

In another study[14], the authors use two primary methods for selecting pairs: the DM[18] and the CA. The DM involves selecting pairs based on the minimum sum of squared distances between their normalized prices, whereas the CA selects pairs that exhibit a stable long-term relationship as determined by cointegration tests. The methodologies are tested across different data frequencies (daily and hourly) and various trading strategies, including fixed threshold, dynamic threshold, and trailing stop-loss. The paper finds that the frequency of the selection period (daily vs. hourly) does not significantly influence the pairs selected by either method. It also finds that CA-selected pairs generally outperform DM-selected pairs. This is attributed to the higher transaction costs associated with the DM due to a greater number of trades. Finally, it finds intraday trading strategies generally outperform daily strategies in terms of raw returns, reflecting the higher volatility and trading opportunities in shorter time frames. However, the benefits are often offset by higher transaction costs. This finding suggests a trade-off between the number of transactions executed and managing trading costs. The paper finally finds that the introduction of a stop-loss mechanism generally reduces profitability, particularly for high-frequency trading. This outcome indicates that while stop-loss orders can protect against extreme losses, they may also limit the ability to capture full mean-reversion profits.

The CP builds on the findings of [14] and employs the CA to identify cointegrated pairs[20] and uses a dynamic threshold PT strategy that updates the Z-score calculation with the most recent data. It also employs the correlation tests as suggested in [16] to back the cointegration tests. Stop-loss is not included in the trading strategy, as one of the research questions is to evaluate whether AI-APTS can outperform RB-APTS, allowing for evaluation under simplified trading conditions. Additionally, as advised by [14], transaction costs are considered in the CP, as they are a crucial component of the financial performance of the trading strategy.

2.2. LSTM-based PT Strategy

Introduced by [21], LSTM models were developed to address the shortcomings of traditional RNNs, particularly the issues of vanishing and exploding gradients. LSTMs utilize a memory cell and gating mechanisms (input, output, and forget gates) to control information flow and maintain long-term dependencies. These features make LSTMs a popular choice for cryptocurrency price prediction, and they are frequently applied in PT strategies.

The following paper has effectively integrated LSTM with a PT strategy.

In [22], the study explores daily cryptocurrency price prediction using multiple ML and DL models, focusing on a PT strategy for Bitcoin and Ethereum, selected for their high correlation and low cointegration p-values among the top 10 liquid coins at the end of the period between 01/01/2018 and 01/09/2019 according to Yahoo.com. Models like LSTM, MA, ARIMA[23], and ANN were used, with LSTM showing superior performance. A PT strategy based on LSTM predictions utilised the rolling Z-score of the predicted spread for signal generation, proving profitable with a SR of 3.05, including fees, and outperforming simple buy-and-hold strategies. The paper[22] has a robust approach to selecting the best ML/DL model based on usual error metrics and the correlation ratio. Conversely, the results of [22] being limited to a single pair and a single SP cannot be generalised. In comparison, the CP employs a systematic approach to select the most liquid coins, identify the most cointegrated pairs, and then applies multiple PT strategies to derive trading results through significance tests. Moreover, [22] used daily data with only 610 observations while the CP used minute interval data, representing 10,384,500 instances over the SP.

While [22] seeks to evaluate whether AI-APTS can beat the market, contributing to the arguments against the weak-form of EMH, the CP aims to compare it to a RB-APTS, answering the second research question.

2.3. Transformers versus LSTM for Time-Series

In recent years, the Transformer architecture has revolutionized the field of artificial intelligence (AI), becoming a central component of mainstream AI advancements.

As introduced by [24], the self-attention mechanism of the architecture enables parallel processing of words, effectively capturing long-range relationships better than earlier models such as RNNs and LSTMs. This innovation allowed Transformers to reach state-of-the-art performance across various NLP benchmarks, exemplified by models like BERT, GPT-3, and T5.

However, their potential extends far beyond NLP, showing exceptional performance in various other domains such as image recognition tasks. In [25], the authors introduced the Vision

Transformer (ViT), which applied transformer architecture to image recognition tasks, demonstrating that Transformers might surpass standard convolutional neural networks (CNNs) in image classification tasks.

Another significant application of Transformers is in time-series forecasting. Traditional models like LSTMs and RNNs were previously considered the best for sequence-to-sequence tasks in time-series data. However, [26] showed that Transformer-based models could outperform these established methods.

The definitive application of Transformers in time-series forecasting was established in 2023 with the introduction of TimeGPT-1[27], a foundational Transformer-based model specifically designed for time-series forecasting. Trained on over 100 billion data points from diverse sources, TimeGPT-1 demonstrated exceptional generalization capabilities, performing well on previously unseen time-series problems.

At the other end of the spectrum, [28] has critically examined the effectiveness of Transformer-based models in long-term time series forecasting (LTSF) and finds that statistical simple one-layer linear models (LTSF-Linear) outperform complex Transformer-based models in long-term time series forecasting(LTSF) across various real-life datasets. The study shows that Transformers struggle with maintaining temporal order in time series data. LTSF-Linear models, on the other hand, demonstrate superior performance, showing significant improvements (20% to 50%) over Transformer models. Additionally, the research reveals that increasing the look-back window size benefits the LTSF-Linear models but not the Transformer models, which tend to overfit to temporal noises. In this context, the CP believes that the Auto-Correlation function (ACF) can help find an optimal lookback period (3.2.5), within which the time series is autocorrelated with its lagged values. All in all, the paper[28] provides a thorough and critical examination of Transformer-based models for LTSF, highlighting significant limitations and challenges. The methodology is robust, with comprehensive empirical evaluations and ablation studies that offer valuable insights into the effectiveness of different model components.

In [29], the paper explores the application of Transformer architectures for High-Frequency Bitcoin-USDT log-return forecasting, comparing their performance to traditional LSTMs. It introduces a novel hybrid Transformer model, HFformer, which effectively incorporates features from various Transformer architectures, addressing issues such as noise sensitivity and discarding positional encoding. Extensive back-testing, ablation studies, and diverse trading strategies provide robust validation of the model's performance. The results indicate that HFformer outperforms LSTM models in cumulative PnL during back-testing, particularly when multiple signals are used for trading. However, the short back-testing period (two days) may not fully capture long-term market dynamics and model robustness. Additionally, the trading environment assumptions (e.g., no trading fees, fixed execution times) may not accurately reflect real-world conditions. Finally, the findings are specific to the BTC-USDT trading pair, with unknown performance on other assets or market conditions. Contrary to [29], the CP uses vanilla Transformer and LSTM models, selecting the best for forecasting performance before evaluating its financial performance with the PT strategy. Additionally, the CP uses sinusoidal functions for positional encoding, while HFformer[29] discarded them.

In [30], the study examines the effectiveness of Transformer-based models versus LSTM-based models on high-frequency financial time series data from the Binance Exchange, tackling both regression and classification tasks. While Transformer models like FEDformer and Autoformer showed lower prediction errors in mid-price predictions, they were still inadequate for practical use. Conversely, LSTM models performed better in predicting mid-price differences, demonstrating higher robustness and profitability in trading simulations. The newly developed DLSTM model excelled in predicting mid-price movements, surpassing both traditional LSTM and Transformer models. Following the approach of [30], the CP seeks to determine whether a vanilla Transformer model can outperform a vanilla LSTM model in predicting mid-prices, addressing the first research question.

2.4. The Research Project's Contribution

This research primarily contributes by exploring the use of Transformers in high-frequency cryptocurrency PT and comparing their predictive performance with LSTM models in standard settings. Additionally, it compares an AI-APTS with a RB-APTS approach, addressing a gap in existing research.

3. Methodology

3.1. Philosophical Approach and Methodology

This CP follows a positivist approach, since it is suited for deriving generalizable knowledge from empirical data, by focusing on quantitative analysis, statistical methods, and objective metrics to assess CM price predictability and PT strategy performance.

The study adopts an inductive approach, starting with the assumption of RB-APTS efficacy and then developing and testing AI-APTS to challenge this baseline within the context of high-frequency CMs.

The study does not involve the collection of qualitative or human data.

3.2. Research Methods

The quantitative methods primarily involve correlational and experimental approaches.

3.2.1. Overview

This paper's general approach(Figure.3) is described as follows. Detail and justification of the methods are provided in the sub-sequent sections.

Initially, the most correlated and cointegrated pairs of cryptocurrencies are identified over an in-sample period(3.2.4), based on a list of the most liquid assets(Table.A7)(3.2.3**Error! Reference source not found.**). The cointegration parameters are then estimated(3.2.4), allowing for the calculation of the spreads.

Cointegrated pairs(Table.A8) are selected for the PT strategy only if the cointegration and correlation relationships are verified over the out-of-sample period[31](3.2.4).

Subsequently, a vanilla LSTM model(3.2.6) and a vanilla Transformer model(3.2.7) are trained with the in-sample data and evaluated with the out-of-sample data, using the cryptocurrencies in the selected pairs. The lookback period is determined by using the ACF cut-off lag of each coin(3.2.5). The models are then compared(3.2.8) to answer the first research question.

The best-performing DL model is used to predict coin prices over the out-of-sample period. From the predicted prices and cointegration parameters, the predicted spreads are derived and compared to the actual spreads. Only the pairs that exhibit low predictive errors and demonstrate strong cointegration and correlation in the predicted spreads(3.2.10) are selected for PT strategies using Z-scores with dynamic thresholds. These strategies are back-tested(3.2.10) using the out-of-sample data. The financial performances of AI-APTS and RB-APTS are then evaluated and compared with significance tests, addressing the second research question.

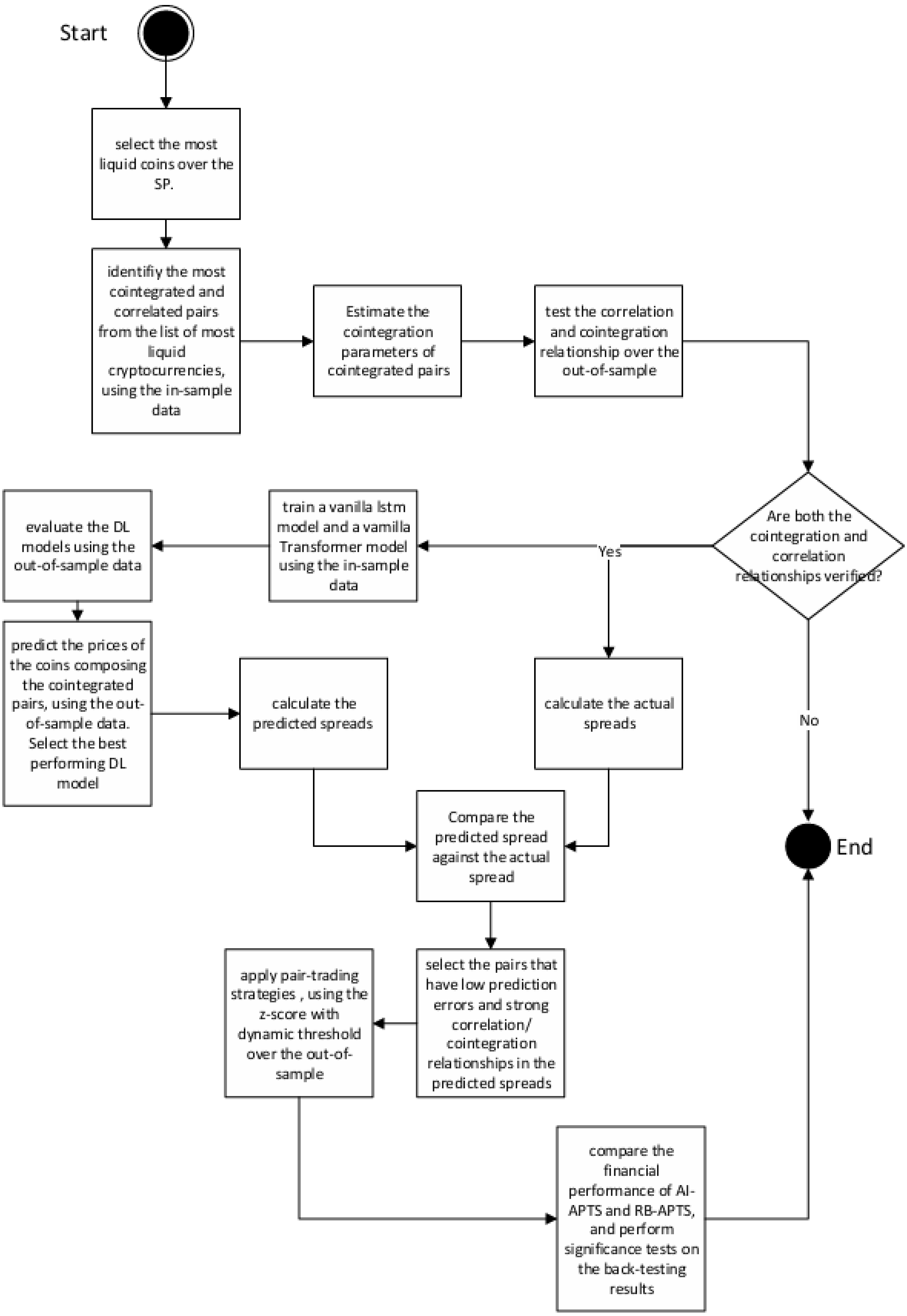

Figure.3 : Flow chart of experiment's steps

3.2.2. Building the Dataset

The dataset is provided in the form of an SQLite database(Table.B16) constructed using the Binance.com API(Table.B11).

It contains the market prices and trading volumes of cryptocurrencies between 01/01/2023 and 23/01/2023. The period from 01/01/2023 to 18/01/2023 is used as the in-sample/training set, and the remaining period is used as the out-of-sample/test set. The selection of such a brief SP is justified, considering that the CP is focused on HFT. The dataset, which is 2.2GB in size, contains 10,384,500 instances.

The cryptocurrencies are quoted against USDT, which generally enhances liquidity[32], and they are traded on Binance, a leading exchange[33].

The in-sample and out-of-sample periods are split approximately 85%-15%, which is a good split ratio[34].

Before training the DL models, the data is standardised[35] using the mean and standard deviation (SD) of the training set, and these statistics are applied to the test set to prevent data leakage.

Standardisation is chosen over normalisation[36] due to its lower sensitivity to outliers[37], which is crucial given the high volatility of CMs in HFT. Normalisation, based on minimum and maximum values, can be distorted by extreme values[38].

3.2.3. Identifying the Most Liquid Cryptocurrencies

The in-sample is used to identify the most liquid coins(Table.A7). Liquid coins have lower volatility[39], leading to more stable and predictable price movements, improving prediction accuracy. They also allow for easier trade execution at predictable prices[40], reducing slippage[41] risks and enabling accurate back-testing.

A coin's liquidity is commonly evaluated using its market capitalisation[42], which is determined by multiplying its current price by the total number of cryptocurrencies in circulation. However, this calculation usually involves data from several major exchanges, while this CP focuses solely on Binance. Additionally, historical market capitalisation data is not easily accessible through major APIs.

The CP rather adopts an approach that calculates the monetary volume for each coin and time period. The monetary volume is determined by multiplying the mid-price by the trading volume. In this study, the mid-price is defined as the average of the low and high prices.

A key question is how to identify coins that were traded during a specific past period, as this serves as the entry point to fetch prices through APIs. The Binance API provides a list of currently active coins, which doesn't help identify coins active at the end of a past in-sample period. This limitation introduces survivorship bias[43], as the analysis only includes coins that persist until the API query, excluding defunct coins from examination. However, this list is a good proxy, as the research focuses on coins active throughout the entire SP.

Several pre-processing steps are required to finalise the list of liquid coins. First, base stablecoins[44] and fiat currencies[45] are omitted as their price fluctuations do not reflect CM dynamics. Observations with null mid-prices are then removed. The 75th percentile of monetary volume is calculated for each time step across all coins, and periods below this threshold are eliminated. Coins with more than 10% missing periods during the in-sample period are excluded. Finally, the average monetary volume per coin is sorted to rank the most liquid assets(Table.A7).

3.2.4. Identifying the Cointegrated Cryptocurrency Pairs

PT strategies use various methods, such as the DM[46] and CA methods, but this CP opts for a cointegration approach to exploit mean-reverting behaviour. Cointegration[47] predicts that if the price spread deviates from the norm, it will revert, allowing profitable trades.

Among popular cointegration methods for PT, this paper uses the Engle-Granger two-step method[48, 49] over the Johansen test[50]. The Engle-Granger test is simpler and more interpretable but introduces a pre-test bias, where results depend on the chosen dependent variable.

To use the Engle-Granger test, this CP employs the log of prices[51] as recommended by [14], rather than raw prices. The log of prices stabilises the variance of the time series[52], making it more likely to exhibit stationarity properties[53] by approximating a normal distribution, crucial for CA. Logarithmic transformations[54] also make proportional price changes appear as additive changes, unlike raw prices which show multiplicative effects.

A prerequisite for the Engle-Granger test is that all coins in the pairs must exhibit non-stationarity[55], verified by performing an Augmented Dickey-Fuller (ADF) test[56] on the logarithm of the prices for each coin, expecting a p-value greater than 0.05.

Next, the cointegration relationship is assessed by performing an OLS regression(Equation.1) of the log prices of one coin on the log prices of the other, estimating the cointegration parameters alpha and beta (the hedge ratio)[57], and calculating the residuals/spreads. The ADF test is then applied to these spreads, expecting a p-value below 0.05 to confirm that the residuals are stationary and mean-reverting.

Equation.1 : Cointegration OLS regression

$$Y_t = \alpha + \beta X_t + \mu_t$$

For simplicity, this paper runs the test on each pair once, using the initial order of roles without switching dependent and independent variables.

All these steps are executed on the in-sample(Table.B12). If the cointegration test is positive, it is validated on the out-of-sample data by calculating the spreads using actual prices and previously obtained cointegration parameters, and then running the ADF test(Table.B12).

Additionally, other statistical properties are calculated for the cointegrated pairs using the in-sample and out-of-sample data(Table.B12), as they provide valuable insights into the cointegration behaviour:

- The Hurst exponent[58] calculated on the spreads: This provides insight into the degree of mean reversion. A value below 0.5 indicates the time series exhibits mean-reverting behaviour.
- The half-life[59] calculated on the spreads: This measures the time it takes for a deviation from the mean to be reduced by half.
- The Pearson correlation[60] calculated between the coins in the pair: This measure provides insight into how the two time series are correlated in the short term, whereas cointegration concerns a long-term relationship.

3.2.5. Determining the Look-Back and Look-Forward Windows

The look-back window[61] refers to the number of previous time steps used as input for the model to generate a prediction. An appropriate look-back window is crucial as it directly affects the model's ability to capture temporal relationships. For LSTMs and Transformers, lagged values as input features are not needed[62]; instead, sequences of time steps are utilised.

In this CP, the lookback period is determined using the ACF[63](Table.B12). It is applied to the mid prices of cointegrated coins to identify the correlation between the time series and its lagged values. When the ACF crosses the confidence interval(Figure.4), it indicates the point beyond which the autocorrelation is no longer statistically significant, suggesting a natural cutoff for meaningful lagged relationships. The CP believes that this method helps tailor the look-back period for each coin.

To prevent data leakage, the look-back sequence in the test set does not overlap with the training set, thereby slightly reducing test set periods due to the forward shift of the first instance.

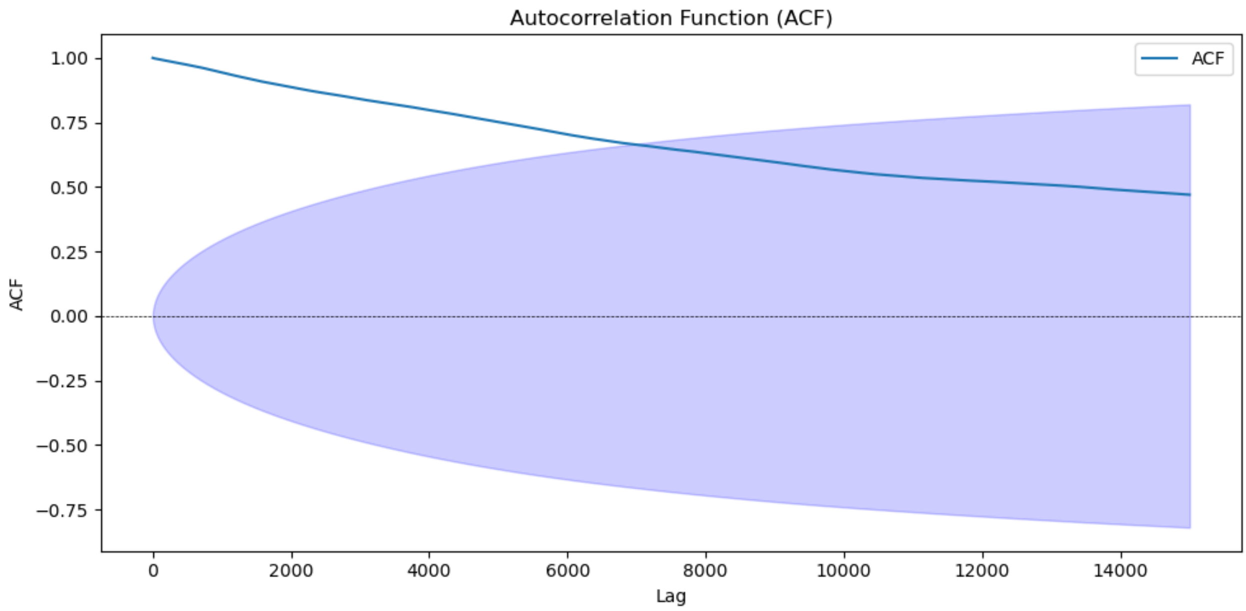

Figure.4 : ACF Correlogram

The look-forward window is the future time span for predictions. It is set to 1 minute to accommodate the time needed for prediction, trading signal generation, processing, and order execution.

3.2.6. Building the LSTM Model Architecture

LSTM key features include the cell state, which acts as long-term memory and is updated by the forget and input gates to prevent loss of important information and address the vanishing gradient problem[64]. The forget gate removes unnecessary information, the input gate adds new information, and the output gate determines the next hidden state. The hidden state functions as short-term memory, holding recent information and interacting with the output gate to produce the network's output.

Given the need to construct an LSTM model for each coin in a cointegrated pair, achieving reasonable processing time is crucial. Therefore, a vanilla LSTM[65] is used across all coins(Table.B13). It has the following architecture:

- A first LSTM layer which consists of 50 units, returning a full sequence of outputs for each input, which is necessary when stacking LSTM layers. The input shape has dimensions of (ACF-based lookback window, 1 input feature) at each time step.
- The second layer contains 3 units and returns the last output in the sequence.
- The dropout layer[66] following each LSTM layer has a rate of 0.2, to help prevent overfitting by randomly setting a portion of input units to zero during training.
- The dense layer is a fully connected layer with a single output, typically used for regression tasks where the output is a single continuous value.

The model is compiled using the Adam optimizer[67] with a moderate learning rate of 0.001.

As part of the training process, the following features are included:

- The training process will run for a maximum of 100 epochs[68].
- A batch size of 32[69] is used, which is the number of samples per gradient update.
- 20% of the training data is set aside for validation.
- The validation loss metric used is Mean Squared Error(MSE) because it provides a smoother and more continuous gradient compared to Mean Absolute Error (MAE).
- Early stopping[70] : training stops if the validation loss does not improve after 10 epochs, which helps prevent overfitting.

3.2.7. Building the Transformer Architecture

The Transformer's self-attention mechanism[71] allows it to assess the importance of elements in a sequence, making it adept at identifying temporal patterns.

As with LSTM, the architecture of a vanilla[72] Transformer model is applied to all coins(Table.B13). The main features[28] are the following :

- The input embedding layer: The embedding dimension is set to 64 for simplicity. Increasing the dimension further would require more computational resources.
- Positional Encoding[73]: Positional encoding is incorporated into the input embeddings to convey information regarding the position of each element within the sequence.

  The CP opts to use a common technique for generating positional encodings, involving sinusoidal functions[74, 75]. For a given position p in the sequence, an embedding dimension d, and the dimension index i, the positional encoding is calculated as follows(Equation.2***Error! Reference source not found.***):

Equation.2 : Sinusoidal function for position encoding

$$PE(2,i) = sin(\frac{p}{10000^{2i/d}})$$

$$PE(2,i+1) = cos(\frac{p}{10000^{2i/d}})$$

This technique has two significant advantages[76]. First, these functions generate values ranging from -1 to 1, which aligns well with the typical value ranges of embeddings. Second, their repetitive patterns allow the model to generalise to positions outside the sequence lengths experienced during training.

This function is preferred over other approaches, such as temporal encodings[77], where temporal information is directly incorporated into the model using features like timestamps, day of the week, and time of day.

- Multi-Head Attention[78]: The model uses 4 heads, which is a reasonable choice and does not heavily impact computational resources. With the input embeddings dimension of 64, each head will handle a 16-dimensional subspace.
- The number of stacked transformer layers is set to 2, aligning with the number of layers in the baseline LSTM model.
- Feed-Forward Neural Network[79]: This network follows each attention layer and applies non-linear transformations to the attention outputs, enabling more complex mappings.
- A dropout layer with a rate of 0.2 is included in the forward pass to prevent overfitting.

Regarding the training process, the same values for early stopping, the number of epochs, and the batch size will be used as in the LSTM model. The validation loss function is also MSE.

3.2.8. Metrics for Evaluating the Prediction Performance

The following metrics[80] are utilised to compare the predictive performance of the LSTM and Transformer models, after the price values were inverse-transformed from the scaled data, in the test set. The metrics are chosen because they are commonly used for regression tasks.

- Mean Squared Error (MSE)[81].
- Root Mean Squared Error (RMSE)[82].
- Mean Absolute Error (MAE)[82].
- R-squared (Coefficient of Determination)[83].
- Mean Absolute Percentage Error (MAPE).

The best-performing model is then used to predict the prices over the out-of-sample.

3.2.9. Eligibility for PT

The predicted prices are used to calculate the predicted spreads using the cointegration parameters from the in-sample. These predicted spreads are then compared to the actual spreads in terms of prediction errors(3.2.8), and the cointegration and correlation relationships of the pairs are assessed using the predicted prices. The objective is twofold: to assess the accuracy of the predicted spreads and to determine whether the cointegration relationship remains valid with these predicted values. Without these two conditions, PT is not profitable(Table.B14).

It is important to note that R-squared and MAPE are not suitable for evaluating prediction errors in this context, as spreads typically fluctuate around 0, encompassing both negative and positive values. Instead, low MAE and RMSE are preferred, with MAE providing an equal weighting to all errors by averaging them, while RMSE penalises larger errors more heavily[84].

For the cointegration relationship to be considered valid, the pairs must exhibit certain characteristics: a strong correlation exceeding 70%, and a robust cointegration relationship, evidenced by an ADF p-value below 0.05 and a Hurst ratio below 0.5.

3.2.10. Back-Testing the PT Strategies

As this CP aims to evaluate the profitability of PT strategies based on predicted spreads compared to actual spreads, the back-testing framework[85] will independently consider these two scenarios(Table.B14).

In the AI-APTS scenario, the predicted spreads contribute to the calculation of a predicted Z-score[86] with a dynamic threshold[14].

Here, this predicted Z-score(Equation.3) Z_t is the predicted spread S_t , adjusted by subtracting the rolling mean $\mu_{S,t,rolling}$, which considers both the predicted spread and the previous actual spreads within the rolling window. This result is then divided by the rolling SD of the same spreads, $\sigma_{S,t,rolling}$. Unlike the fixed threshold approach[46], where the statistical properties of the actual spreads are calculated over the entire in-sample data and employed throughout the out-of-sample period, the rolling window approach captures recent market dynamics by continuously updating the mean and SD.

Equation.3 : rolling Z-score

$$Z_t = \frac{S_t - \mu_{S,t,rolling}}{\sigma_{S,t,rolling}}$$

The last actual Z-score is calculated by using the last actual spread and the same rolling window.

20 rolling windows are used, expressed as multiples of one day, ranging from 0.25 days to 5 days in 0.25-day increments.

Based on the values of the predicted Z-score and the last actual Z-score, a multi-value signal is generated to guide the trading strategy[87]:

- If the predicted Z-score surpasses a positive threshold and the last actual Z-score is below it, a signal (1) is generated to initiate short positions if none currently exist (buy asset 2, sell asset 1) and to close all long positions.
- If the last actual Z-score falls below 0 and its value lagged by one period is positive, a signal (3) is generated to exit a short position.
- If the predicted Z-score falls below a negative threshold and the last actual Z-score is above it, a signal (2) is generated to initiate long positions if none currently exist and to close all short positions.
- If the last actual Z-score surpasses 0 while its value lagged by one period is negative, a signal (4) is generated to exit a long position.

Typically, the threshold is set at 2 SDs. However, during periods of high volatility, this threshold is adjusted to 2.5 SDs. Volatility is measured by comparing the rolling SD of the spread to the 75th percentile of the spread's SD, using a reasonable rolling window of 60 minutes. Failing to adjust the threshold could result in multiple trades triggered by noise, increasing transaction costs and reducing overall performance.

The execution of the AI-based opening signal, based on the predicted Z-score, is delayed by 1 minute to simulate execution at the time when the actual prices are known, accounting for prediction and processing times.

The AI-based closing signals, relying on the actual Z-score, are systematically delayed by one minute, similar to the RB-APTS, to approximate the bid-ask spread and account for the risk of slippage, as noted in [48, 88]. In other papers[29], price slippage is generally considered through a penalty applied to the amounts bought and sold.

In summary, the AI-APTS scenario comprises 20 sub-scenarios (corresponding to 20 rolling windows), representing various PT strategies applied to the same predicted spreads.

In the RB-APTS scenario, actual spreads are used to calculate the actual rolling Z-scores. All trading signals are delayed by 1 minute. The same rolling windows are used.

The initial portfolio is assumed to be 10,000 USDT, equally distributed between the two coins. However, there are two adjustments to consider. First, the actual number of coin units bought or sold is determined so that no position in either of the paired coins exceeds 10% of their respective period volumes. This approach aligns with current liquidity conditions and aims to minimise the impact on market prices[89]. Second, to comply with the cointegration relationship of the two coins, the hedge ratio is used as a relative volume ratio between both coins.

As noted in[48], Binance's standard fees are reduced as monetary volumes increase. However, the CP opts for a fixed 0,05% round-trip transaction cost, accounting for both implicit and explicit costs such as bid-ask spreads [90].

The evaluation metrics used to compare both scenarios include:

- The Sharpe ratio[91](Equation.4). In this experiment, it is based on the annualised return R_a, a risk-free rate of 1% per annum R_f and the σ_a the annualised volatility (AV). A SR of 2 is considered excellent and means that for every unit of risk (as measured by volatility), the portfolio is earning 2 units of excess return over the risk-free rate.

Equation.4 : Sharpe ratio

$$SR = \frac{R_a - R_f}{\sigma_a}$$

- The annualised return(Equation.5). R_t is the cumulative return(Equation.6) at the end of the trading period T. N is the total number of minute intervals in a year. d_i is the duration of each trade interval in minutes. r_t(Equation. 7) is 1-minute return at time t. $V_t$is the portfolio value at time t.

Equation.5:Annualised return

$$(1 + R_T)^{\frac{N}{\sum_{i=1}^{T} d_i}} - 1$$

Equation.6: cumulative return

$$R_T = \prod_{i=1}^{t}(1 + r_t) - 1$$

Equation.7 : 1-minute return

$$r_t = \frac{V_t - V_{t-1}}{V_{t-1}}$$

- The annualised volatility measures the degree of variation in the portfolio's returns (Equation.8). σ_r is the SD of r_t .

Equation.8 : Annualised volatility

$$\sigma_r * \sqrt{N}$$

- The Net Pnl.
- The fees.
- The maximum drawdown[92].
- The average number of transactions.
- The average trade duration.
- The average winning trade ratio.

It is important to note that for a given pair, strategy, and rolling window, if there are no transactions or only a single transaction, the tuple is excluded from the analysis, as the SR and AV calculations require at least two transactions.

3.2.11. Significance Tests

The exploratory data analysis (EDA) employs significance tests to identify significant outcomes throughout this experiment. The first approach involves comparing two independent series (AI-APTS vs. RB-APTS, or LSTM vs Transformer) using a two-sample test. The second approach examines each series separately, employing a one-sample test to determine the highest threshold at which each metric is significantly greater.

For the two-sample test, it is essential to first check if the data for the metric is normally distributed using the Shapiro-Wilk test[93]. If both series are normally distributed, the next step is to verify if they have equal variances using Levene's test[94]. Based on the results of these tests, the two-sample t-test[95] is used to compare the means of the two groups. If either dataset is not normally distributed, the non-parametric Mann-Whitney U test[96] is used instead.

For the one-sample test, if the data is not normally distributed, the non-parametric Wilcoxon signed-rank test[97] is applied. The Wilcoxon signed-rank test is appropriate for testing whether the median of a single sample significantly differs from a specified value. The one-sample test is iteratively repeated to determine the highest threshold at which significance is achieved.

In all cases, the p-value is compared against a threshold of 0.05 for decision-making.

3.2.12. Data Sources

The primary source of public market data is Binance.com, accessed for free via its API.

3.2.13. Software and Tools

Python and its libraries, including pandas, numpy, scipy, torch, are used for data collection via API and preprocessing tasks. For EDA and visualization, matplotlib and seaborn are utilized, along with statsmodels for statistical analysis. DL models are built, trained, and assessed using TensorFlow, PyTorch, and sklearn within the Google Colab environment, using an AT100 GPU. Back-testing is conducted in Python. The dataset is stored in an SQLite database (

Table.B16: Dataset

Table.B16).

The following Jupyter notebooks are run sequentially in Google Colab:

- Table.B11: Fetches market data with the Binance API and stores them in an SQLite database.
- Table.B12: Identifies cointegrated pairs.
- Table.B13: Builds 34 LSTM and Transformer models.
- Table.B14: Manages AI-APTS and RB-APTS back-testing processes.
- Table.B15: Conducts significance tests on the experiment's results.

3.3. Ethical Implications

The primary ethical concerns involve the transparency and accountability[98] of ML-based trading decisions. Ensuring clarity in the decision-making processes of trading algorithms, especially those using complex DL models like Transformers and LSTMs, is crucial to avoid accountability issues, particularly when trades fail. It is important for traders and beneficiaries to communicate these processes clearly to regulators and stakeholders.

ML models trained on historical data may also perpetuate existing biases[99] in financial markets, potentially leading to unfair treatment of certain cryptocurrencies. With high transaction volumes, these biases could disrupt the market.

Additionally, ML models combined with algo-trading can increase the risk of market manipulation[100]. Advanced algorithms might be used to exploit low-volume assets for pump-and-dump schemes[101, 102] or create false signals by placing large orders with no intention of executing them. Furthermore, small investors and those without access to advanced ML-based trading technology may be excluded from certain market opportunities, worsening economic inequalities.

4. Results

4.1. List of most liquid pairs per SP

Table.A7 highlights the 24 most liquid cryptocurrencies(3.2.3) based on the monetary volume. Unsurprisingly, Bitcoin is at the top of the ranking. EURUSDT is removed as it is a fiat currency. Based on these coins quoted in USDT, pairs are formed over the in-sample period and a cointegration test is performed to identify the most cointegrated pairs (3.2.4).

4.2. List of Cointegrated Pairs

Table.A8 presents 27 cointegrated pairs(3.2.4) and their statistical characteristics over the in-sample and out-of-sample, including the ADF p-value of the spreads, the cointegration parameters alpha and beta, the half-life, the Hurst exponent, the correlation ratio and its p-value, and the look-back window for each coin.

It is observed that both the ADF p-values and the correlation p-values are below 0.05, correlation ratios are close to 1, Hurst exponents are below 0.5, all of which indicate strong correlation and cointegration relationships.

The short half-life (max value is 1414 minutes) confirms a rapid mean-reverting pattern[103].

The ACF cut-off lag(3.2.5) for each coin, utilised as the lookback windows for the DL models, is around 2300 minutes. This indicates a reasonable level of autocorrelation and provides suitable lookback sequences for training the DL models.

Some pairs have a cointegration beta greater than 1 (ex: NEARUSDT/ LUNCUSDT), indicating a strong correlation between the underlying coins. This involves taking larger positions in the dependent coin relative to the independent coin to exploit the mean-reversion dynamics(Equation.1).

4.3. Evaluating the Performance of LSTM and Transformer Models

Table.A9

Table.A9 presents the performance of 34 vanilla LSTM and Transformer models(3.2.6, 3.2.7) based on the error metrics(3.2.8) and the training times.

The results from two-sample significance tests(3.2.11) comparing the 2 models(Table.1) indicate that the distribution of each metric is non-normal(p-value for Shapiro-Wilk normality test <0.05). Therefore, the Mann-Whitney U test is performed. The tests reveal that Transformer models significantly outperform LSTM models in terms of MAPE and R-squared metrics. Additionally, LSTM models significantly have longer training times.

Table.1 : Two-sample t-tests for models 'performance

metric	H0	Ha	p_value_mannwhitney	decision
mae	lstm>=transformer	transformer>lstm	0.805405294	Fail to reject H0
mae	transformer>=lstm	lstm>transformer	0.204219218	Fail to reject H0
mape	lstm>=transformer	transformer>lstm	0.999032128	Fail to reject H0
mape	transformer>=lstm	lstm>transformer	0.001086625	Reject H0

mse	lstm>=transformer	transformer>lstm	0.795780782	Fail to reject H0
mse	transformer>=lstm	lstm>transformer	0.214121674	Fail to reject H0
r2	lstm>=transformer	transformer>lstm	0.000765335	Reject H0
r2	transformer>=lstm	lstm>transformer	0.999320567	Fail to reject H0
rmse	lstm>=transformer	transformer>lstm	0.795780782	Fail to reject H0
rmse	transformer>=lstm	lstm>transformer	0.214121674	Fail to reject H0
training_time	lstm>=transformer	transformer>lstm	0.997873854	Fail to reject H0
training_time	transformer>=lstm	lstm>transformer	0.002368686	Reject H0

Table.2 presents the statistical properties of models' performance. It indicates that the R-squared values are almost negative for the LSTM models, whereas this is less pronounced for the Transformer models, which have a median R-squared of 88.13% but a negative mean R-squared of -27.33%, suggesting that both models are struggling to capture the variance. Conversely, the MAPE shows low values, indicating its reduced sensitivity to outliers due to the averaging of percentage errors. This suggests that while both models might perform well for most data points, they fail for a few points with high variance, resulting in a low average error (low MAPE) but a poor overall fit (negative R-squared).

Table.2 : Statistical properties of models' performance

metric	model	mean	std	min	25%	50%	75%	max
training time(seconds)	lstm	1307.29109	528.4105212	825.3484249	948.6503	**1176.546**	1435.343	2932.920434
	transformer	840.331243	467.7753415	338.0597179	570.9362	**742.3583**	1018.385	2248.831492

r2(%)	lstm	-386.83097	736.1921149	-2002.793406	-121.527	-28.7078	18.29767	84.33154076
	transformer	-27.33385	270.639074	-908.7786165	42.4244	88.12958	93.35678	99.11051249
Mape(%)	lstm	**3.46504917**	3.124527897	0.730097265	1.310618	**1.97306**	5.030924	11.64303965
	transformer	**1.33962969**	1.782403731	0.184272268	0.361321	**0.592011**	1.334242	5.887485919

Table.3, utilising the one-sample t-test(3.2.11), indicates that the training time is significantly higher than 1048 seconds for LSTM models and 642 seconds for Transformer models. Additionally, the MAPE is significantly higher than 1.64% for LSTM models and 0.42% for Transformer models.

Table.3 : One-sample t-tests for model's performance

metric	model	Hypothesized population mean (max_threshold)	Ha	P-value for the Wilcoxon test
training time in seconds	lstm	1048	metric>max_threshold	0.039840698
	transformer	642		0.049186707
r2 (%)	lstm	0		0.960159302
	transformer	0		0.095046997

mape(%)	lstm	1.64		0.039840698
	transformer	0.42		0.025268555

4.4. Evaluating the Predicted Spreads

Based on these previous results, Transformer models are utilised to predict prices over the out-of-sample period. These predictions are then used to create the predicted spreads, using the respective cointegration parameters of the pairs(Table.A8). These predicted spreads are compared with the actual spreads using MAE and RMSE(3.2.8, 3.2.9), and the correlation and cointegration relationships are assessed using the predicted spreads.

Subsequently, the eligibility criteria for PT (3.2.9) are applied, reducing the initial list of 27 cointegrated pairs down to 11 pairs. Table.4 shows these pairs with their cointegration features along with their error metrics.

By ranking the pairs in ascending order based on their MAE and RMSE errors, all pairs following LTCUSDT/LUNCUSDT in Table.4 are excluded due to their high RMSE. Furthermore, an examination of the plots for the top four pairs reveals that the predicted prices of XRPUSDT in the XRPUSDT/TRXUSDT pair have a low R-squared of 11.3%, indicating that the Transformer model struggles to capture the variance, as shown in Figure.8, Figure.9, and Figure.10. As a result, this pair is not retained, leaving the final list of pairs eligible for PT as :

- LINKUSDT/ATOMUSDT,
- DOTUSDT/JASMYUSDT,

- and LTCUSDT/LUNCUSDT.

In the LINKUSDT/ATOMUSDT pair, as shown in Figure.5, Figure.6, and Figure.7, there is significant variance in the actual spreads around the period near 2023-01-22, which the model failed to capture. This issue is due to inaccurate price predictions for the underlying coin ATOMUSDT during that time.

For the DOTUSDT/JASMYUSDT pair, the predicted spreads consistently fall below the actual spread, as illustrated in Figure.11,Figure.12, Figure.13. This discrepancy is caused by the predicted prices of DOTUSDT (Figure.12) following a similar pattern.

For the LTCUSDT/LUNCUSDT pair, as depicted in Figure.14, Figure.15, and Figure.16, the opposite pattern is observed, with the predicted spread being higher than the actual spreads. This is due to the same pattern in the predicted prices of the underlying coin LTCUSDT.

Table.4 : Cointegrated pairs eligible for PT.

coin1	coin2	mae	rmse	mse	adf_p_value	correlation	hurst	half_life
LINKUSDT	ATOMUSDT	0.002787963	0.003777898	1.43E-05	0.016438239	0.8954291	0.387788	88
XRPUSDT	TRXUSDT	0.005799341	0.007272301	5.29E-05	0.035363868	0.733775456	0.306808	63
DOTUSDT	JASMYUSDT	0.005905579	0.006801403	4.63E-05	0.046110426	0.946571314	0.308578	46
LTCUSDT	LUNCUSDT	0.006185705	0.006995315	4.89E-05	0.019298448	0.720286231	0.388774	128
NEARUSDT	LUNCUSDT	0.009991793	0.015951756	0.000254459	0.002456193	0.834205867	0.420271	75
XRPUSDT	LTCUSDT	0.010811159	0.012498127	0.000156203	0.001955146	0.851345148	0.396877	75
SOLUSDT	DOTUSDT	0.011384544	0.014070483	0.000197978	0.003826173	0.882424479	0.352842	80
ADAUSDT	LUNCUSDT	0.011721281	0.015873295	0.000251961	0.000996134	0.752580037	0.419122	81
SANDUSDT	DOTUSDT	0.0534231	0.057841006	0.003345582	0.049352928	0.765246769	0.382799	139
SOLUSDT	SANDUSDT	0.060448405	0.067020135	0.004491699	0.007713953	0.930538578	0.416277	177
SANDUSDT	JASMYUSDT	0.0618024	0.067925335	0.004613851	0.024956304	0.743031059	0.412692	191

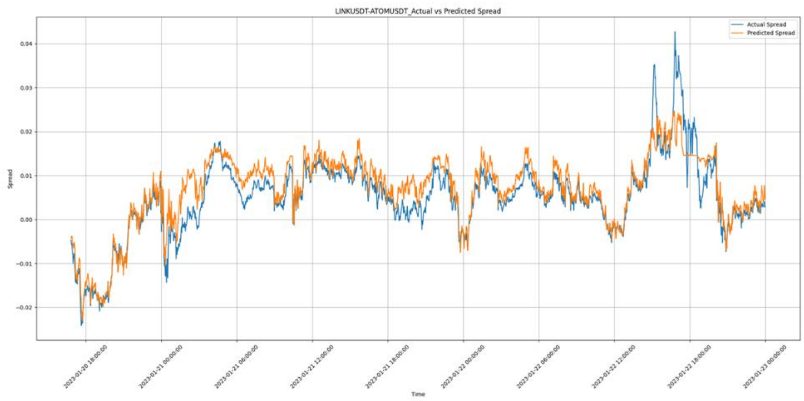

Figure.5 : LINKUSDT/ATOMUSDT pair - actual vs predicted spreads

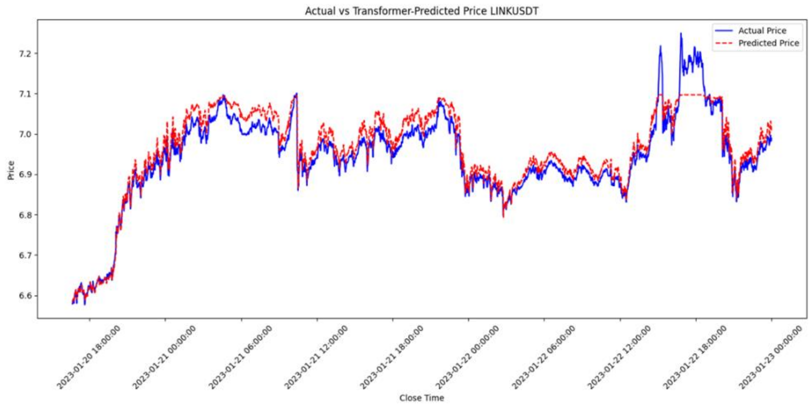

Figure.6 : LINKUSDT - actual vs predicted prices

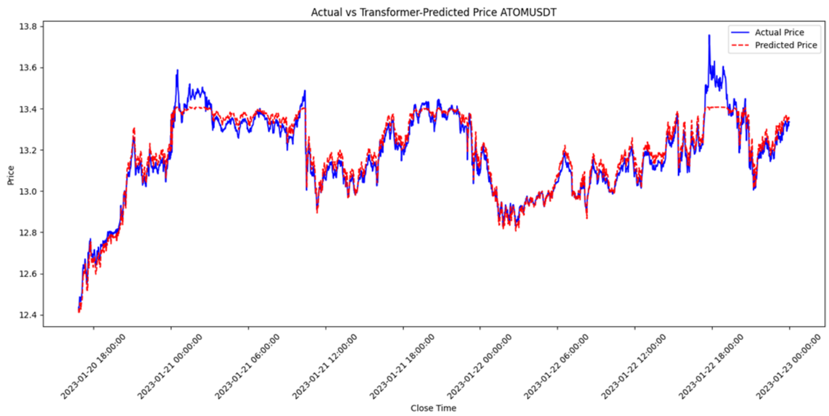

Figure.7 : ATOMUSDT - actual vs predicted prices

Figure.8 : XRPUSDT/TRXUSDT pair - actual vs predicted spreads

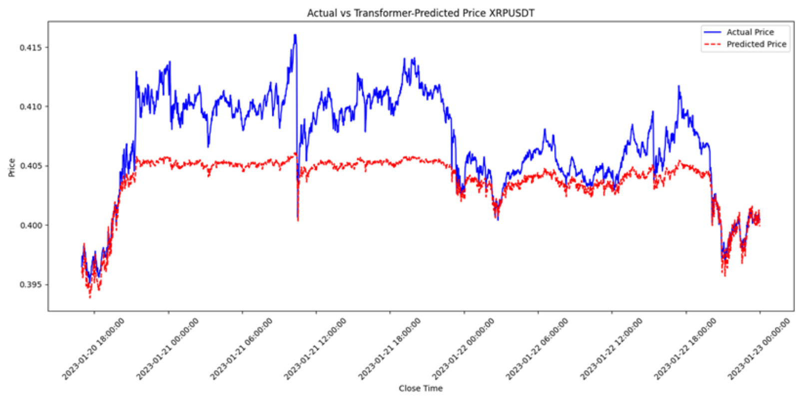

Figure.9 : XRPUSDT - actual vs predicted prices

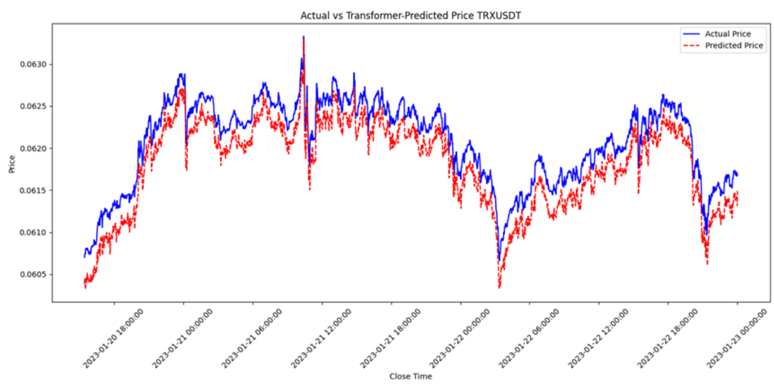

Figure.10 : TRXUSDT - actual vs predicted prices

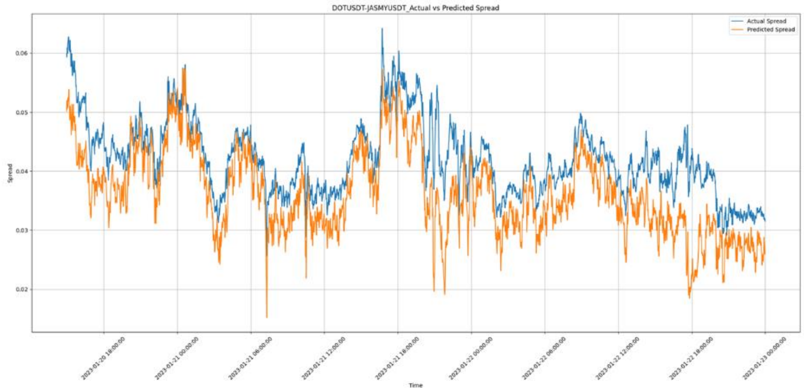

Figure.11 : DOTUSDT/JASMYUSDT pair - actual vs predicted spreads

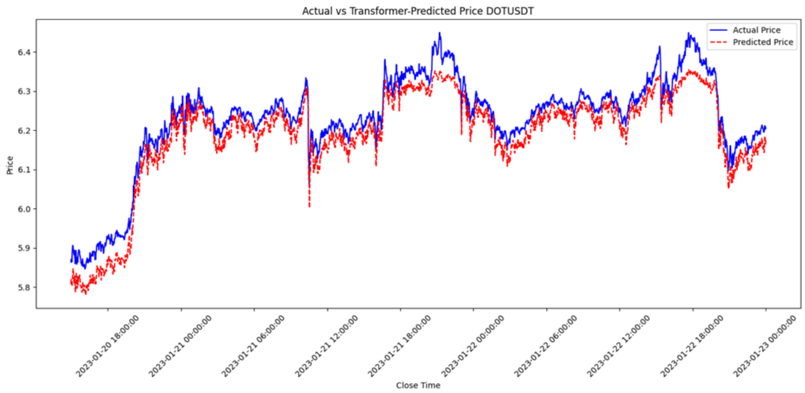

Figure.12 : DOTUSDT - actual vs predicted prices

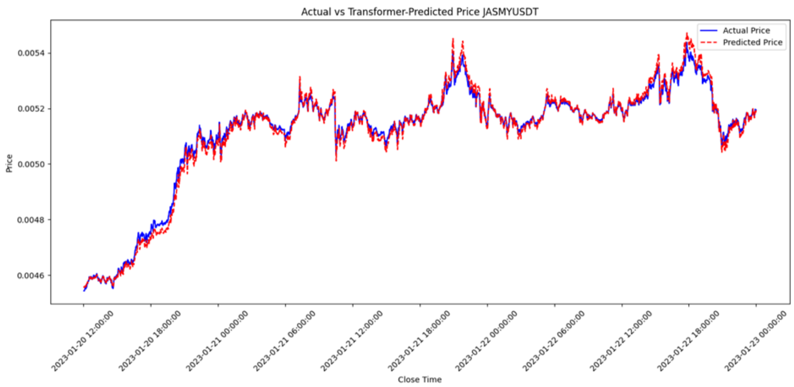

Figure.13 : JASMYUSDT - actual vs predicted prices

Figure.14 : LTCUSDT/LUNCUSDT pair - actual vs predicted spreads

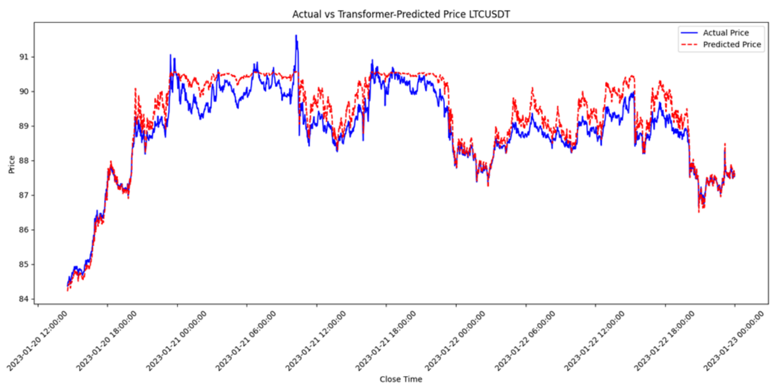

Figure.15 : LTCUSDT - actual vs predicted prices

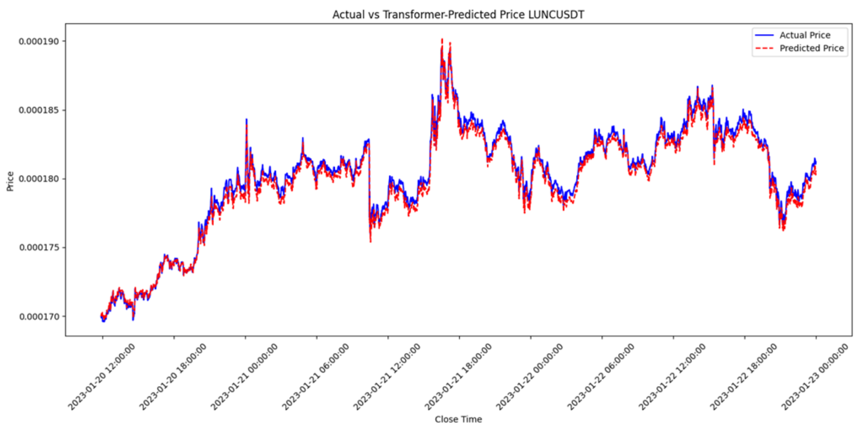

Figure.16 : LUNCUSDT - actual vs predicted prices

4.5. Comparing the Financial Performance between AI-APTS vs RB-APTS.

For each pair, two sets of PT strategies are designed: one using AI-APTS and the other using RB-APTS. With 20 rolling windows, this results in 60 PT strategies per set, as there are 3 pairs selected. However, after removing null SR values from AI-APTS, the number is reduced to 26 per set.

Table.A10 presents the descriptive statistics of back-testing results for both strategies. It is noted that AI-APTS and RB-APTS show exceptionally high SRs, with median values of 9.77 and 29.87, respectively. This is attributed to the very short duration of transactions, leading to unusually high ARs.

Given that AI-APTS has higher SR and AR mean values and lower median values than RB-APTS, this suggests that AI-APTS is a riskier strategy. This pattern is also evident in Figure.17, where the SR values for AI-APTS are right-skewed compared to RB-APTS. Both strategies exhibit similar levels of AV.

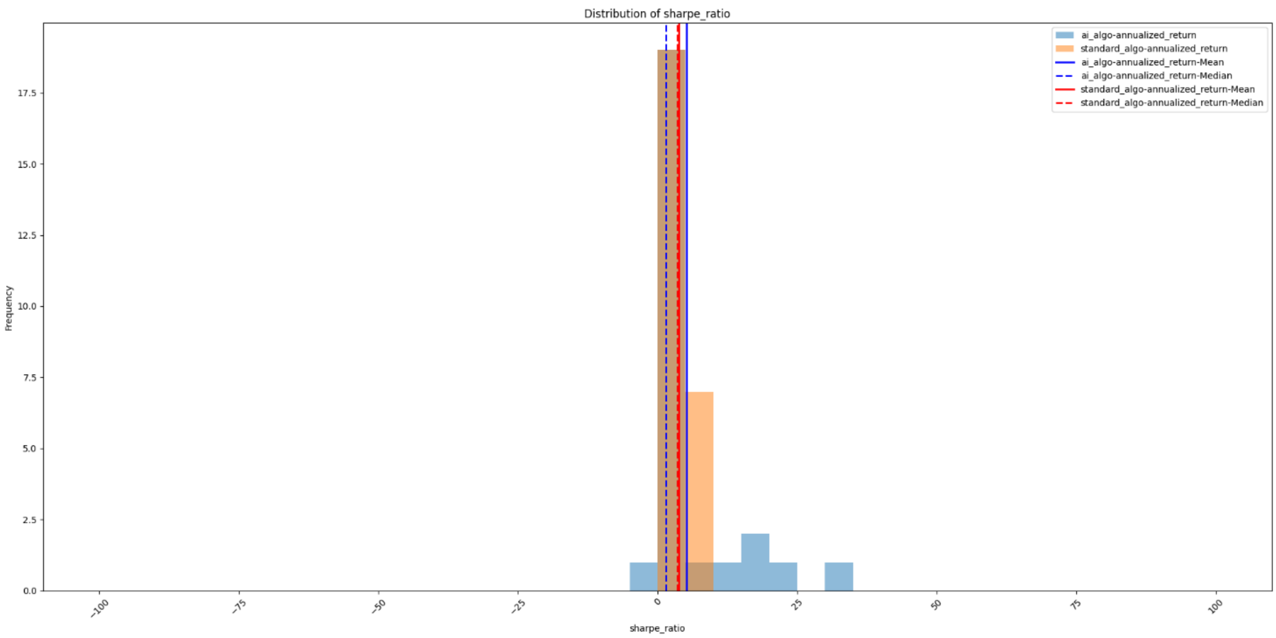

Figure.17 : Distribution of Sharpe ratio values

To understand the significance of the results of the 2 back-tested strategies, 2 one-tailed two-sample t-tests are conducted(3.2.11) for each metric(3.2.10), testing successively the alternative hypotheses “ai-algo>standard-algo” and “ai-algo<=standard-algo”.

Table.5 presents the results of these tests. As the Shapiro-Wilk test finds that the distributions are non-normal, p-values from Mann-Whitney U test are used to make decisions about H_0.

It can be concluded that the SR, AR, and WR of AI-APTS are significantly lower than those of RB-APTS. Additionally, the number of trades, fees, and maximum drawdown are significantly higher for AI-APTS, supporting the assumption that lower returns are caused by more transactions, resulting in higher fees and greater losses. This is consistent with the significantly shorter average trade duration. The figures also indicate that there are no significant differences in terms of AV and net PnL.

More insight are provided into the performance of each strategy by determining the maximum threshold value at which each metric becomes significantly greater. This process involves conducting multiple one-sample t-tests(3.2.11), by incrementally adjusting the threshold value until the point is found where the test shows significance.

Table.6 presents the results of these tests, showing that AI-APTS achieves a SR over 8, an AR exceeding 128%, and a WR above 73%. While AI-APTS does not perform as well as RB-APTS, it remains highly profitable, assuming these high returns maintain consistency over the course of the year.

Table.5 : Two-sample t-tests for back-testing results

metric	H0	Ha	p_value_mannwhitney	decision
annualised_return	ai_algo>=standard_algo	standard_algo>ai_algo	0.034305827	Reject H0
annualised_volatility			0.770713367	Fail to reject H0
avg_trade_duration			0.001560094	Reject H0
fees			0.997219738	Fail to reject H0
max_drawdown			0.999994608	Fail to reject H0
net_pnl			0.083525793	Fail to reject H0
num_trades			0.997799199	Fail to reject H0
sharpe_ratio			0.029101011	Reject H0
win_ratio			0.002027401	Reject H0
annualised_return	standard_algo>=ai_algo	ai_algo>standard_algo	0.967062232	Fail to reject H0
annualised_volatility			0.234871577	Fail to reject H0
avg_trade_duration			0.998530004	Fail to reject H0
fees			0.002940629	Reject H0
max_drawdown			5.96E-06	Reject H0
net_pnl			0.91924949	Fail to reject H0
num_trades			0.002331661	Reject H0
sharpe_ratio			0.972092313	Fail to reject H0
win_ratio			0.99809225	Fail to reject H0

Table.6 : One-sample t-tests for back-testing results

metric	Ha	threshold	p_value_ttest	p_value_wilcoxon
annualised_return(%)	ai_algo>threshold	128%		0.049667478
annualised_volatility(%)		14.4%	0.04695037	
avg_trade_duration		124.24		0.049667478
fees		5.89	0.04917863	
max_drawdown		0.0044%		0.047036514
net_pnl		17.6	0.04982272	
num_trades		7.45	0.04933504	
sharpe_ratio		8		0.049667478
win_ratio(%)		73%		0.042098656
annualised_return	standard_algo>threshold	309%	0.04823092	
annualised_volatility		12.2%		0.047036514
avg_trade_duration		218.85		0.049667478
fees		2.99		0.047036514
max_drawdown		0		0.158655254
net_pnl		22.79		0.049667478
num_trades		3.5		0.007566065
sharpe_ratio		21.32		0.049667478
win_ratio		87%		0.044514507

5. Discussion

5.1. Interpretation of Results

5.1.1. Performance of LSTM/Transformer Models

The comparison between vanilla LSTM and Transformer models(4.3) shows that Transformers significantly outperform LSTMs in terms of prediction errors (MAPE and R-squared) and training time. Transformer's lower training time is due to their parallel processing capabilities, unlike LSTM's sequential nature. LSTM models, more than Transformers, often exhibit negative or low R-squared values due to high variance, possibly caused by the fairly long lookback sequence tailored to each cryptocurrency. Using a shorter sequence, such as the half-life, could give more weight to recent movements and improve performance. Fine-tuning the hyperparameters could be considered, but it might lead to overfitting[104] and extended training times[105], thus negating the aim of maintaining reasonably low training times, especially since the models need to be frequently rebuilt in a production setting. Here lies the bias-variance dilemma[106, 107].

Consequently, these results support that $H_0 - 1$ expressed in 1.3 is rejected in favour of $H_a - 1$.

5.1.2. Evaluation of Back-Testing Results

The experiment’s results show that AI-APTS performs significantly worse than RB-APTS in terms of SR, AR, and WR. Although the SR is unusually high due to extraordinary high AR , the comparison remains valid. This underperformance is linked to more trades, higher fees, greater maximum drawdown, and shorter trading durations, indicating a riskier strategy. Prediction errors likely resulted in the generation of more trading signals, many of which were probably unjustified, during back-testing.

These results fail to reject $H_0 - 2$ expressed in 1.3.

5.2. Implications

This CP has demonstrated that Transformer models are more suitable than LSTM models for predicting prices in high-frequency CMs. However, like LSTM models, they struggle to capture high variance. Since PT profitability depends on effectively capturing volatility, it is essential to find solutions that enhance these models' ability to do so.

Moreover, 2 pairs (Figure.11, Figure.14) showed consistent lags in the predicted spreads, which were either higher or lower than the actual spreads. This could be problematic, as the predicted Z-score would also lag behind the actual Z-score, potentially leading AI-APTS to generate outdated trading signals.

Nevertheless, the results also showed that the financial underperformance of AI-APTS is somewhat offset by the fact that a PT position comprises both long and short positions in two cointegrated cryptocurrencies, making the net position almost delta neutral. Therefore, despite high errors in price predictions, the existence of a genuine cointegration relationship is crucial.

In summary, the initial goal was to leverage the early insights from price predictions to counteract the slippage of RB-APTS. However, this objective remains an unfulfilled aspiration as long as the price predictions lack satisfactory accuracy.

Unlike [22], which analysed a single pair and found AI-APTS outperforming the market's buy-and-hold strategy, this CP's results are less optimistic. It finds that AI-APTS does not outperform RB-APTS, despite taking a more comprehensive approach across multiple pairs. Statistical tests were also employed to ensure the results' significance.

5.3. Limitations and Future Work

There are a few limitations that may challenge the validity of the experiment:

- It is observed that some cryptocurrencies are found in multiple cointegrated pairs, suggesting baskets of interconnected cryptocurrencies. If a single coin is part of many pairs, adverse movements in that coin could impact multiple pairs simultaneously, increasing risk. For a more comprehensive analysis, the Johansen test[108] could be considered in future work (FW), which can handle multiple time series simultaneously, and PT could be replaced by a risk management strategy considering multiple cointegrated coins.
- The mid-prices used in this CP may not accurately reflect the true volume of activity. In a production setting, this might have adverse consequences. FW could consider using Limit Order Book (LOB) prices instead of mid-prices, as they can more effectively capture the bid-ask spread.
- This CP does not exclude cryptocurrencies that cannot be short-sold on the Binance exchange, and the fee ratio of 0.05% does not accurately reflect real-world conditions where fees vary with transaction volume. FW could incorporate these considerations.
- The CP assumes that orders based on predicted prices can be executed seamlessly once the actual price is known, thereby avoiding slippage. However, it does not explore how this could

be achieved. FW could investigate this area, potentially involving the use of bots and advanced strategies.

- The study was conducted on a single time period and exclusively on the Binance exchange. FW could extend the analysis to different time periods to account for varying market conditions and to other exchanges to improve generalisability.

6. Conclusion

The primary goal of the CP was to determine the optimal DL model, between LSTM and Transformer, to gain early insights into future prices, allowing for order execution when the actual price is announced, thus providing an edge over RB-APTS, which often suffers from slippage. Two research questions were posed: How do vanilla Transformer models compare to vanilla LSTMs in forecasting high-frequency cryptocurrency prices? And how can AI-APTS outperform RB-APTS?

Key findings were derived through a systematic process, starting from identifying cointegrated pairs to predicting prices with the optimal model and back-testing various PT strategies. Three evaluations were performed in the out-of-sample to select the best cointegrated pairs : the cointegration relationship, the price predictions, and the spread predictions.

Transformer models outperformed LSTM in error predictions and training time under vanilla settings, rejecting $H_0 - 1$. However, Transformers struggle to capture the high variance needed in volatile HFT, causing their PT results to fall short of RB-APTS, despite remaining very profitable due to valid cointegration relationships. These PT results suggest that as long as such inaccuracies persist, RB-APTS will remain the preferred choice, failing to reject $H_0 - 2$.

However, the study has limitations. It used mid-price as a shortcut, did not exclude cryptocurrencies that cannot be short-sold, applied a fixed cost ratio, and assumed AI-based order execution without slippage. The study's generalizability is limited as it focuses on a single 3-week period and only on the Binance exchange. FW should replicate these findings across different periods and exchanges.

In conclusion, AI should be applied cautiously in algorithmic pair trading systems, limited to specific areas with safeguards. In this CP, AI was used only for entry-point signals, while exit points relied on standard algorithms.

REFERENCES

[1] coinwire.com. "WHICH COUNTRIES TRADE CRYPTO THE MOST? A Comprehensive Analysis." [Online]. Available: https://coinwire.com/crypto-trading-report-2024/ (accessed Aug. 10, 2024).

[2] "Cryptocurrency Trends 2024 & The Impact on Global Economics and Finance." [Online]. Available: https://www.bulbapp.io/p/e286ec43-3902-449a-8992-880960ec6eba/cryptocurrency-trends-2024-the-impact-on-global-economics-and-finance (accessed Aug. 10, 2024).

[3] "Global cryptocurrency adoption." [Online]. Available: https://www.triple-a.io/cryptocurrency-ownership-data (accessed Aug. 10, 2024).

[4] F. Fang *et al.*, "Cryptocurrency trading: a comprehensive survey," *Financial Innovation,* vol. 8, no. 1, p. 13, 2022, doi: https://doi.org/10.1186/s40854-021-00321-6 (accessed Jul. 8, 2024)

[5] J. Liu and A. Serletis, "Volatility in the cryptocurrency market," *Open Economies Review,* vol. 30, no. 4, pp. 779-811, 2019, doi: https://doi.org/10.1007/s11079-019-09547-5 (accessed Jul. 1, 2024)

[6] G. Orfanidis, "Mathematical aspects of Pairs Trading," ed, 2020 https://www.diva-portal.org/smash/get/diva2:1477748/FULLTEXT01.pdf (accessed Aug. 1, 2024)

[7] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 17-20 Dec. 2018 2018, pp. 1394-1401, doi: https://doi.org/10.1109/ICMLA.2018.00227. (accessed Jul. 6, 2024)

[8] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled, "Overview of the Transformer-based Models for NLP Tasks," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, 2020: IEEE, pp. 179-183, doi: https://doi.org/10.15439/2020F20. (accessed Jul. 8, 2024)

[9] Q. Wen *et al.*, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125,* 2022, doi: https://doi.org/10.48550/arXiv.2202.07125 (accessed Jul. 9, 2024)

[10] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool, "Transformers in time-series analysis: A tutorial," *Circuits, Systems, and Signal Processing,* vol. 42, no. 12, pp. 7433-7466, 2023, doi: https://doi.org/10.1007/s00034-023-02454-8 (accessed Aug. 9, 2024)

[11] B. Abou Tanos and G. Badr, "Price Delay and Market Efficiency of Cryptocurrencies: The Impact of Liquidity and Volatility during the COVID-19 Pandemic," *Journal of Risk and Financial Management,* vol. 17, no. 5, p. 193, 2024, doi: https://doi.org/10.3390/jrfm17050193 (accessed May 22, 2024)

[12] C. Krauss, "Statistical arbitrage pairs trading strategies: Review and outlook," *Journal of Economic Surveys,* vol. 31, no. 2, pp. 513-545, 2017, doi: https://doi.org/10.1111/joes.12153 (accessed Apr. 3, 2024)

[13] A. Pole, *Statistical arbitrage: algorithmic trading insights and techniques*. John Wiley & Sons, 2011. https://books.google.pt/books?hl=fr&lr=&id=xSjXTnKqIKoC&oi=fnd&pg=PT11&dq=Statistical+arbitrage:+algorithmic+trading+insights+and+techniques&ots=Jv1DNMZELt&sig=4RXsPFk4jOvh38ZRgbP1NkVeVc8&redir_esc=y#v=onepage&q=Statistical%20arbitrage%3A%20algorithmic%20trading%20insights%20and%20techniques&f=false (accessed Jun. 8, 2024)

[14] C. Lesa and R. Hochreiter, "Cryptocurrency Pair Trading," *Available at SSRN 4433530,* 2023, doi: https://dx.doi.org/10.2139/ssrn.4433530 (accessed Apr. 9, 2024)

[15] N. Fu, M. Kang, J. Hong, and S. Kim, "Enhanced Genetic-Algorithm-Driven Triple Barrier Labeling Method and Machine Learning Approach for Pair Trading Strategy in Cryptocurrency Markets," *Mathematics,* vol. 12, no. 5, p. 780, 2024, doi: https://doi.org/10.3390/math12050780 (accessed Jun. 9, 2024)

[16] S. T. G. Nair, "Pairs trading in cryptocurrency market: A long-short story," *Investment Management and Financial Innovations,* vol. 18, no. 3, pp. 127-141, 2021, doi: http://dx.doi.org/10.21511/imfi.18(3).2021.12 (accessed Jun. 7, 2024)

[17] R. K. Prematunga, "Correlational analysis," *Australian Critical Care,* vol. 25, no. 3, pp. 195-199, 2012/08/01/ 2012, doi: https://doi.org/10.1016/j.aucc.2012.02.003 (accessed Jul. 13, 2024)

[18] A. Basu, H. Shioya, and C. Park, *Statistical inference: the minimum distance approach*. CRC press, 2011. https://books.google.pt/books?hl=fr&lr=&id=C-lOIgDp5_0C&oi=fnd&pg=PP1&dq=A.+Basu,+H.+Shioya,+and+C.+Park,+Statistical+inference:+the+mi

nimum+distance+approach.+CRC+press,+2011.&ots=nMQVwmYT4b&sig=FfHlF5HA7-6Dwq3SHgEkVMMKywk&redir_esc=y#v=onepage&q=A.%20Basu%2C%20H.%20Shioya%2C%20and%20C.%20Park%2C%20Statistical%20inference%3A%20the%20minimum%20distance%20approach.%20CRC%20press%2C%202011.&f=false (accessed May 16, 2024)
[19] H. J. Bierens, "Cointegration analysis," ed: Citeseer, 1997, pp. 217-246 https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9b171500978443e1c20e1a7c142887cb2dc6ec09 (accessed Jul. 24, 2024)
[20] D. d. S. Carvalho, "Pairs trading: cointegration-based methods: applied to the cryptocurrency market," 2021. [Online]. Available: https://repositorio.ucp.pt/handle/10400.14/36905 (accessed Jun. 15, 2024)
[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6795963 (accessed Apr. 9, 2024)
[22] E. Osifo and R. Bhattacharyya, "Cryptocurrency trading-pair forecasting, using machine learning and deep learning technique," *Using Machine Learning and Deep Learning Technique (March 5, 2020),* 2020, doi: https://dx.doi.org/10.2139/ssrn.3610340 (accessed Jul. 5, 2024)
[23] R. H. Shumway, D. S. Stoffer, R. H. Shumway, and D. S. Stoffer, "ARIMA models," *Time series analysis and its applications: with R examples,* pp. 75-163, 2017, doi: https://doi.org/10.1007/978-3-319-52452-8_3 (accessed Jun. 25, 2024)
[24] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017. [Online]. Available: https://cir.nii.ac.jp/crid/1370849946232757637 (accessed Jun. 19, 2024)
[25] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2020, doi: https://doi.org/10.48550/arXiv.2010.11929 (accessed Jul. 7, 2024)
[26] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," *arXiv preprint arXiv:2001.08317,* 2020, doi: https://doi.org/10.48550/arXiv.2001.08317 (accessed Aug. 6, 2024)
[27] A. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," *arXiv preprint arXiv:2310.03589,* 2023, doi: https://doi.org/10.48550/arXiv.2310.03589 (accessed Jul. 17, 2024)
[28] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, no. 9, pp. 11121-11128, doi: https://doi.org/10.1609/aaai.v37i9.26317. (accessed Aug. 5, 2024)
[29] F. Barez, P. Bilokon, A. Gervais, and N. Lisitsyn, "Exploring the advantages of transformers for high-frequency trading," *arXiv preprint arXiv:2302.13850,* 2023, doi: https://doi.org/10.48550/arXiv.2302.13850 (accessed Aug. 5, 2024)
[30] P. Bilokon and Y. Qiu, "Transformers versus LSTMs for electronic trading," *arXiv preprint arXiv:2309.11400,* 2023, doi: https://doi.org/10.48550/arXiv.2309.11400 (accessed Jun. 12, 2024)
[31] G. Keilbar and Y. Zhang, "On cointegration and cryptocurrency dynamics," *Digital Finance,* vol. 3, pp. 1-23, 2021, doi: https://doi.org/10.1007/s42521-021-00027-5 (accessed 11 July 2024)
[32] A. Jobst, D. Atzberger, R. Henker, J. O. Vollmer, W. Scheibel, and J. Döllner, "Examining liquidity of exchanges and assets and the impact of external events in centralized crypto markets: a 2022 study," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2023: IEEE, pp. 1-6, doi: https://doi.org/10.1109/ICBC56567.2023.10174905. (accessed Aug. 12, 2024)
[33] V. K. Manda and L. K. Nihar, "Lessons From the FTX Cryptocurrency Exchange Collapse," in *Cases on the Resurgence of Emerging Businesses*: IGI Global, 2023, pp. 19-36. 10.4018/978-1-6684-8488-3.ch002. (accessed Aug. 12, 2024)
[34] I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts," in *7th international Mardin Artuklu scientific research conference*, 2022, pp. 496-504. [Online]. Available: https://www.researchgate.net/profile/Ismail-Muraina/publication/358284895_IDEAL_DATASET_SPLITTING_RATIOS_IN_MACHINE_LEARNING_ALGORITHMS_GENERAL_CONCERNS_FOR_DATA_SCIENTISTS_AND_DATA_ANALYSTS/links/61fb97e711a1090a79cc1a8b/IDEAL-DATASET-SPLITTING-RATIOS-IN-MACHINE-LEARNING-ALGORITHMS-

GENERAL-CONCERNS-FOR-DATA-SCIENTISTS-AND-DATA-ANALYSTS.pdf. [Online]. Available: https://www.researchgate.net/profile/Ismail-Muraina/publication/358284895_IDEAL_DATASET_SPLITTING_RATIOS_IN_MACHINE_LEARNING_ALGORITHMS_GENERAL_CONCERNS_FOR_DATA_SCIENTISTS_AND_DATA_ANALYSTS/links/61fb97e711a1090a79cc1a8b/IDEAL-DATASET-SPLITTING-RATIOS-IN-MACHINE-LEARNING-ALGORITHMS-GENERAL-CONCERNS-FOR-DATA-SCIENTISTS-AND-DATA-ANALYSTS.pdf (accessed Aug. 12, 2024)

[35] B. Kumari and T. Swarnkar, "Importance of data standardization methods on stock indices prediction accuracy," in *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 1*, 2020: Springer, pp. 309-318, doi: https://doi.org/10.1007/978-981-15-1081-6_26. (accessed Aug. 01 2024)

[36] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The impact of data normalization on the accuracy of machine learning algorithms: a comparative analysis," in *International Conference on Soft Computing Models in Industrial and Environmental Applications*, 2023: Springer, pp. 344-353, doi: https://doi.org/10.1007/978-3-031-42536-3_33. (accessed Aug. 01, 2024)

[37] X. Ma, "The Investigation of LSTM-Random Search with Various Standardization and Normalization Technologies," *Highlights in Science, Engineering and Technology,* vol. 85, pp. 1087-1094, 2024, doi: https://doi.org/10.54097/xgeyhr93 (accessed Aug. 01, 2024)

[38] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462,* 2015, doi: https://doi.org/10.48550/arXiv.1503.06462 (accessed Aug. 12, 2024)

[39] P. R. Hansen, C. Kim, and W. Kimbrough, "Periodicity in cryptocurrency volatility and liquidity," *Journal of Financial Econometrics,* vol. 22, no. 1, pp. 224-251, 2024, doi: https://doi.org/10.1093/jjfinec/nbac034 (accessed Aug. 01, 2024)

[40] K. H. Al-Yahyaee, W. Mensi, H.-U. Ko, S.-M. Yoon, and S. H. Kang, "Why cryptocurrency markets are inefficient: The impact of liquidity and volatility," *The North American Journal of Economics and Finance,* vol. 52, p. 101168, 2020, doi: https://doi.org/10.1016/j.najef.2020.101168 (accessed Aug. 01,2024)

[41] J. Shah, P. Javare, and D. Khetan, "SlipSwap: Reduce the slippage that is incurred during the swap of tokens using Algorithmic analysis," in *2020 IEEE Bombay Section Signature Conference (IBSSC)*, 2020: IEEE, pp. 131-134, doi: https://doi.org/10.1109/IBSSC51096.2020.9332185. (accessed May 24, 2024)

[42] O. Norberg and V. Boukov, "The relationship between trading volume, market capitalization, and volatility: Normal versus abnormal market conditions," ed, 2024 https://www.diva-portal.org/smash/get/diva2:1867641/FULLTEXT01.pdf (accessed Aug. 1,2024)

[43] X. Liang, "Survivorship bias in mutual fund performance: evidence in Canadian mutual funds," Concordia University, 2000. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/1154/ (accessed May 6, 2024)

[44] J. Fernandez-Mejia, "Extremely stablecoins," *Finance Research Letters,* vol. 63, p. 105268, 2024. [Online]. Available: https://doi.org/10.1016/j.frl.2024.105268 (accessed Aug. 01, 2024)

[45] C. Pfister, "Monetary sovereignty in the digital currency era," in *Digital Assets and the Law*: Routledge, 2024, pp. 35-66. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.4324/9781003258261-5/monetary-sovereignty-digital-currency-era-christian-pfister. (accessed Aug. 01, 2024)

[46] A. Sharma, "Digital Assets Pair Trading." [Online]. Available: https://www.researchgate.net/profile/Aviral-Sharma-18/publication/382267993_Digital_Assets_Pair_Trading/links/66956bda4a172d2988a26956/Digital-Assets-Pair-Trading.pdf (accessed Aug. 01, 2024)

[47] N. Huck and K. Afawubo, "Pairs trading and selection methods: is cointegration superior?," *Applied Economics,* vol. 47, no. 6, pp. 599-613, 2015, doi: https://doi.org/10.1080/00036846.2014.975417 (accessed Aug. 01, 2024)

[48] M. Fil and L. Kristoufek, "Pairs trading in cryptocurrency markets," *IEEE Access,* vol. 8, pp. 172644-172651, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3024619 (accessed Jun. 7, 2024)

[49] H. Sebastião and P. Godinho, "Forecasting and trading cryptocurrencies with machine learning under changing market conditions," *Financial Innovation,* vol. 7, pp. 1-30, 2021, doi: https://doi.org/10.1186/s40854-020-00217-x (accessed Jul. 5, 2024)
[50] K. S. Lai and M. Lai, "A cointegration test for market efficiency," *Journal of Futures Markets,* vol. 11, no. 5, 1991. [Online]. Available: https://openurl.ebsco.com/EPDB%3Agcd%3A9%3A25985225/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A5915751&crl=c (accessed Aug. 12, 2024)
[51] S.-H. Sung, J.-M. Kim, B.-K. Park, and S. Kim, "A Study on Cryptocurrency Log-Return Price Prediction Using Multivariate Time-Series Model," *Axioms,* vol. 11, no. 9, p. 448, 2022, doi: https://doi.org/10.3390/axioms11090448 (accessed Aug. 01 , 2024)
[52] L. C. G. Rogers and S. E. Satchell, "Estimating variance from high, low and closing prices," *The Annals of Applied Probability,* pp. 504-512, 1991, doi: https://www.jstor.org/stable/2959703 (accessed Aug. 01, 2024)
[53] A. Witt, J. Kurths, and A. Pikovsky, "Testing stationarity in time series," *physical Review E,* vol. 58, no. 2, p. 1800, 1998, doi: https://doi.org/10.1103/PhysRevE.58.1800 (accessed Aug. 1, 2024)
[54] O. N. Keene, "The log transformation is special," *Statistics in medicine,* vol. 14, no. 8, pp. 811-819, 1995, doi: https://doi.org/10.1002/sim.4780140810 (accessed Aug. 01, 2024)
[55] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with koopman predictors," *Advances in Neural Information Processing Systems,* vol. 36, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/hash/28b3dc0970fa4624a63278a4268de997-Abstract-Conference.html (accessed Aug. 10, 2024)
[56] D. Maheta and M. Mehta, "Crypto Dynamics: Analyzing The Price Interplay Between Bitcoin And Ethereum," *Educational Administration: Theory and Practice,* vol. 30, no. 5, pp. 13747-13754, 2024. [Online]. Available: https://kuey.net/index.php/kuey/article/view/6016 (accessed Aug. 5, 2024)
[57] S.-S. Chen, C. F. Lee, F.-L. Lin, and K. Shrestha, "Hedge Ratios: Theory and Applications," in *Handbook of Investment Analysis, Portfolio Management, and Financial Derivatives: In 4 Volumes*: World Scientific, 2024, pp. 1277-1328. https://doi.org/10.1142/9789811269943_0038. (accessed Aug. 11, 2024)
[58] M. J. Villena and A. A. Araneda, "On sectoral market efficiency," *Finance Research Letters,* vol. 61, p. 104949, 2024, doi: https://doi.org/10.1016/j.frl.2023.104949 (accessed Aug. 9, 2024)
[59] J. Ramos Jungblut, "Multivariate Cointegration in Statistical Arbitrage," *Available at SSRN,* 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4906546 (accessed Jul. 3, 2024)
[60] H. Yu and A. D. Hutson, "Inferential procedures based on the weighted Pearson correlation coefficient test statistic," *Journal of Applied Statistics,* vol. 51, no. 3, pp. 481-496, 2024, doi: https://doi.org/10.1080/02664763.2022.2137477 (accessed Jun. 8, 2024)
[61] Q. Liu, A. Singh, J. Liu, C. Mu, Z. Yan, and J. Pedersen, "Long or Short or Both? An Exploration on Lookback Time Windows of Behavioral Features in Product Search Ranking," 2024. [Online]. Available: https://sigir-ecom.github.io/eCom24Papers/paper_5.pdf (accessed May 19, 2024)
[62] B. Vega-Márquez, J. Solís-García, I. A. Nepomuceno-Chamorro, and C. Rubio-Escudero, "A comparison of time series lags and non-lags in Spanish electricity price forecasting using data science models," *Logic Journal of the IGPL,* p. jzae034, 2024, doi: https://doi.org/10.1093/jigpal/jzae034 (accessed Aug. 7, 2024)
[63] H. Bjørnstad, Ø. L. Halvorsen, and V. C. J. Rognås, "Forecasting the price of salmon using machine learning algorithms," NTNU, 2023. [Online]. Available: https://hdl.handle.net/11250/3071433 (accessed Jun. 6, 2024)
[64] H. Yadav and A. Thakkar, "NOA-LSTM: An efficient LSTM cell architecture for time series forecasting," *Expert Systems with Applications,* vol. 238, p. 122333, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122333 (accessed Jun. 4, 2024)
[65] R. Qamar, R. Asif, L. F. Naz, A. Mannan, and A. Hussain, "FlightForecast: A Comparative Analysis of Stack LSTM and Vanilla LSTM Models for Flight Prediction," *VFAST Transactions on Software*

Engineering, vol. 12, no. 1, pp. 13-24, 2024, doi: https://doi.org/10.21015/vtse.v12i1.1740 (accessed May 25, 2024)
[66] Z. Zhang and Z.-Q. J. Xu, "Implicit regularization of dropout," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2024, doi: https://doi.org/10.1109/TPAMI.2024.3357172 (accessed Jun 30, 2024)
[67] K. Ahn, Z. Zhang, Y. Kook, and Y. Dai, "Understanding Adam optimizer via online learning of updates: Adam is FTRL in disguise," *arXiv preprint arXiv:2402.01567,* 2024, doi: https://doi.org/10.48550/arXiv.2402.01567 (accessed Aug. 10,2024)
[68] B. M. Hussein and S. M. Shareef, "An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning," in *ITM Web of Conferences*, 2024, vol. 64: EDP Sciences, p. 01003, doi: https://doi.org/10.1051/itmconf/20246401003. (accessed May 4, 2024)
[69] T. T.-K. Lau, H. Liu, and M. Kolar, "AdAdaGrad: Adaptive Batch Size Schemes for Adaptive Gradient Methods," *arXiv preprint arXiv:2402.11215,* 2024, doi: https://doi.org/10.48550/arXiv.2402.11215 (accessed Jun 27, 2024)
[70] T. Miseta, A. Fodor, and Á. Vathy-Fogarassy, "Surpassing early stopping: A novel correlation-based stopping criterion for neural networks," *Neurocomputing,* vol. 567, p. 127028, 2024, doi: https://doi.org/10.1016/j.neucom.2023.127028 (accessed May 30, 2024)
[71] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao, and C. Zheng, "Synthesizer: Rethinking Self-Attention for Transformer Models," presented at the Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2021. [Online]. Available: https://proceedings.mlr.press/v139/tay21a.html (accessed Jul. 7, 2024)
[72] M. Yao *et al.*, "Spike-driven transformer," *Advances in neural information processing systems,* vol. 36, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/hash/ca0f5358dbadda74b3049711887e9ead-Abstract-Conference.html (accessed Jul. 31, 2024)
[73] E. Eldele, M. Ragab, Z. Chen, M. Wu, and X. Li, "Tslanet: Rethinking transformers for time series representation learning," *arXiv preprint arXiv:2404.08472,* 2024, doi: https://doi.org/10.48550/arXiv.2404.08472 (accessed Jun. 5, 2024)
[74] J. Zhang *et al.*, "Intriguing Properties of Positional Encoding in Time Series Forecasting," *arXiv preprint arXiv:2404.10337,* 2024, doi: https://doi.org/10.48550/arXiv.2404.10337 (accessed May 6, 2024)
[75] D. Cao and S. Zhang, "AD-autoformer: decomposition transformers with attention distilling for long sequence time-series forecasting," *The Journal of Supercomputing,* pp. 1-21, 2024, doi: https://doi.org/10.1007/s11227-024-06266-8 (accessed Jul 31, 2024)
[76] T. Schmieg and C. Lanquillon, "Time Series Representation Learning: A Survey on Deep Learning Techniques for Time Series Forecasting," in *International Conference on Human-Computer Interaction*, 2024: Springer, pp. 422-435, doi: https://doi.org/10.1007/978-3-031-60606-9_25. (accessed May 30, 2024)
[77] Y. Zhang, L. Ma, S. Pal, Y. Zhang, and M. Coates, "Multi-resolution Time-Series Transformer for Long-term Forecasting," in *International Conference on Artificial Intelligence and Statistics*, 2024: PMLR, pp. 4222-4230. [Online]. Available: https://proceedings.mlr.press/v238/zhang24l.html. [Online]. Available: https://proceedings.mlr.press/v238/zhang24l.html (accessed Apr. 30,2024)
[78] R. Fu *et al.*, "Transformer-BLS: An efficient learning algorithm based on multi-head attention mechanism and incremental learning algorithms," *Expert Systems with Applications,* vol. 238, p. 121734, 2024, doi: https://doi.org/10.1016/j.eswa.2023.121734 (accessed May 30, 2024)
[79] J. Jiang *et al.*, "Transformer technology in molecular science," *Wiley Interdisciplinary Reviews: Computational Molecular Science,* vol. 14, no. 4, p. e1725, 2024, doi: https://doi.org/10.1002/wcms.1725 (accessed Aug. 10, 2024)
[80] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports,* vol. 14, no. 1, p. 6086, 2024, doi: https://doi.org/10.1038/s41598-024-66611-y (accessed Jun. 17, 2024)

[81] V. Kamble and S. Deshmukh, "Comparision between accuracy and MSE, RMSE by using proposed method with imputation technique," *Oriental journal of computer science and technology,* vol. 10, no. 4, pp. 773-779, 2017, doi: http://dx.doi.org/10.13005/ojcst/10.04.11 (accessed Aug. 12, 2024)
[82] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)," *Geoscientific model development discussions,* vol. 7, no. 1, pp. 1525-1534, 2014, doi: https://doi.org/10.5194/gmdd-7-1525-2014 (accessed 2024-08-02)
[83] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science,* vol. 7, p. e623, 2021, doi: https://doi.org/10.7717/peerj-cs.623 (accessed Apr. 27, 2024)
[84] W. Wang and Y. Lu, "Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model," in *IOP conference series: materials science and engineering*, 2018, vol. 324: IOP Publishing, p. 012049, doi: 10.1088/1757-899X/324/1/012049. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/324/1/012049 (accessed Aug. 11, 2024)
[85] K. Kostiainen, "Development of Trading Algorithm Backtest Environment," 2016. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/106897/Kostiainen_Kari.pdf (accessed Aug. 11, 2024)
[86] Y. Hwang, S. Zohren, and Y. Lee, "Temporal Representation Learning for Stock Similarities and Its Applications in Investment Management," *arXiv preprint arXiv:2407.13751,* 2024, doi: https://doi.org/10.48550/arXiv.2407.13751 (accessed Jun. 30, 2024)
[87] J. Sen, "Designing efficient pair-trading strategies using cointegration for the Indian stock market," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, 2022: IEEE, pp. 1-9, doi: https://doi.org/10.1109/ASIANCON55314.2022.9909455. (accessed Aug. 10, 2024)
[88] T. G. Fischer, C. Krauss, and A. Deinert, "Statistical arbitrage in cryptocurrency markets," *Journal of Risk and Financial Management,* vol. 12, no. 1, p. 31, 2019, doi: https://doi.org/10.3390/jrfm12010031 (accessed Aug. 8, 2024)
[89] S. Yao, A. Sensoy, D. K. Nguyen, and T. Li, "Investor attention and cryptocurrency market liquidity: a double-edged sword," *Annals of Operations Research,* vol. 334, no. 1, pp. 815-856, 2024, doi: https://doi.org/10.1007/s10479-022-04915-w (accessed Jun. 22, 2024)
[90] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," *Complexity,* vol. 2018, pp. 1-16, 2018, doi: https://doi.org/10.1155/2018/8983590 (accessed May 4, 2024)
[91] A. W. Lo, "The statistics of Sharpe ratios," *Financial analysts journal,* vol. 58, no. 4, pp. 36-52, 2002. [Online]. Available: https://www.tandfonline.com/doi/abs/10.2469/faj.v58.n4.2453 (accessed 2024-08-03)
[92] M. Magdon-Ismail and A. F. Atiya, "Maximum drawdown," *Risk Magazine,* vol. 17, no. 10, pp. 99-102, 2004. [Online]. Available: https://ssrn.com/abstract=874069 (accessed Jul. 31, 2024)
[93] E. González-Estrada and W. Cosmes, "Shapiro–Wilk test for skew normal distributions based on data transformations," *Journal of Statistical Computation and Simulation,* vol. 89, no. 17, pp. 3258-3272, 2019, doi: https://doi.org/10.1080/00949655.2019.1658763 (accessed Jun. 2, 2024)
[94] D. W. Nordstokke and B. D. Zumbo, "A new nonparametric Levene test for equal variances," *Psicológica,* vol. 31, no. 2, pp. 401-430, 2010. [Online]. Available: https://www.redalyc.org/pdf/169/16917017011.pdf (accessed May 15, 2024)
[95] N. Cressie and H. Whitford, "How to use the two sample t-test," *Biometrical Journal,* vol. 28, no. 2, pp. 131-148, 1986, doi: https://doi.org/10.1002/bimj.4710280202 (accessed Apr 30, 2024)
[96] R. Wall Emerson, "Mann-Whitney U test and t-test," vol. 117, ed: SAGE Publications Sage CA: Los Angeles, CA, 2023, pp. 99-100 https://doi.org/10.1177/0145482X221150592 (accessed Jun. 18, 2024)
[97] I. C. A. Oyeka and G. U. Ebuh, "Modified Wilcoxon signed-rank test," *Open Journal of Statistics,* vol. 2, no. 2, pp. 172-176, 2012, doi: http://dx.doi.org/10.4236/ojs.2012.22019 (accessed Jun. 12, 2024)

[98] B. Kim and F. Doshi-Velez, "Machine learning techniques for accountability," *AI Magazine,* vol. 42, no. 1, pp. 47-52, 2021. [Online]. Available: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/7481 (accessed Aug. 10, 2024)

[99] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR),* vol. 54, no. 6, pp. 1-35, 2021, doi: https://doi.org/10.1145/3457607 (accessed May 10, 2024)

[100] S. Koehler, N. Dhameliya, B. Patel, and S. K. R. Anumandla, "AI-Enhanced Cryptocurrency Trading Algorithm for Optimal Investment Strategies," *Asian Accounting and Auditing Advancement,* vol. 9, no. 1, pp. 101-114, 2018. [Online]. Available: https://www.researchgate.net/profile/Samuel-Koehler/publication/380710990_AI-Enhanced_Cryptocurrency_Trading_Algorithm_for_Optimal_Investment_Strategies/links/664a40ed0b0d28457447f2c4/AI-Enhanced-Cryptocurrency-Trading-Algorithm-for-Optimal-Investment-Strategies.pdf (accessed Aug. 3, 2024)

[101] T. Li, D. Shin, and B. Wang, "Cryptocurrency pump-and-dump schemes," *Available at SSRN 3267041,* 2021, doi: https://dx.doi.org/10.2139/ssrn.3267041 (accessed Aug. 5, 2024)

[102] J. Xu and B. Livshits, "The anatomy of a cryptocurrency {Pump-and-Dump} scheme," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1609-1625. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/xu-jiahua. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/xu-jiahua (accessed Aug. 10, 2024)

[103] R. R. Ahmed, J. Vveinhardt, D. Streimikiene, and Z. A. Channar, "Mean reversion in international markets: evidence from GARCH and half-life volatility models," *Economic research-Ekonomska istraživanja,* vol. 31, no. 1, pp. 1198-1217, 2018, doi: https://doi.org/10.1080/1331677X.2018.1456358 (accessed Apr. 5, 2024)

[104] M. Kuhn, K. Johnson, M. Kuhn, and K. Johnson, "Over-fitting and model tuning," *Applied predictive modeling,* pp. 61-92, 2013, doi: https://doi.org/10.1007/978-1-4614-6849-3_4 (accessed May 4, 2024)

[105] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences,* vol. 44, no. 1, pp. 1-12, 2004, doi: https://doi.org/10.1021/ci0342472 (accessed Jun. 11, 2024)

[106] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation,* vol. 4, no. 1, pp. 1-58, 1992, doi: https://doi.org/10.1162/neco.1992.4.1.1 (accessed Aug. 11, 2024)

[107] E. Briscoe and J. Feldman, "Conceptual complexity and the bias/variance tradeoff," *Cognition,* vol. 118, no. 1, pp. 2-16, 2011, doi: https://doi.org/10.1016/j.cognition.2010.10.004 (accessed Aug. 11, 2024)

[108] G. P. Dwyer, "The Johansen tests for cointegration," *White Paper,* vol. 516, 2015. [Online]. Available: https://web.archive.org/web/20180415022127id_/http://www.jerrydwyer.com/pdf/Clemson/Cointegration.pdf (accessed Aug. 8, 2024)

APPENDICES

Appendix.A

Table.A7 : List of the most liquid cryptocurrencies over the in-sample period

symbol	Monetary volume average	rank
BTCUSDT	3,087,738.72	1
ETHUSDT	437,444.05	2
SOLUSDT	119,946.97	3
XRPUSDT	100,256.32	4
GALAUSDT	82,885.32	5
DOGEUSDT	70,266.02	6
APTUSDT	63,536.92	7
BNBUSDT	58,567.10	8
SHIBUSDT	53,277.00	9
LTCUSDT	48,905.16	10
MATICUSDT	46,387.96	11
ADAUSDT	35,694.63	12
LDOUSDT	28,689.75	13
NEARUSDT	28,062.21	14
SANDUSDT	27,694.82	15
APEUSDT	24,234.00	16
FILUSDT	24,024.31	17
LINKUSDT	21,224.81	18
DOTUSDT	21,174.60	19
TRXUSDT	20,977.74	20
LUNCUSDT	19,202.31	21
ATOMUSDT	18,642.82	22
EURUSDT	16,023.09	23
JASMYUSDT	12,640.24	24

Table.A8 : Most cointegrated pairs

coin1	coin2	In-sample correlation	In-sample residuals adf p-value	In-sample alpha	In-sample beta	In-sample half-file	In-sample hurst	In-sample coin1 lookback	In-sample coin2 lookback	out-of-sample correlation	out-of-sample residuals adf p-value	out-of-sample half-file	out-of-sample hurst
BTCUSDT	LUNCUSDT	0.8893589	0.036822069	21.45578	1.33617584	1411	0.462864	2423	2152	0.905502	0.016188404	302	0.4892858
ETHUSDT	XRPUSDT	0.9546593	0.009304027	8.852473	1.60840137	850	0.467788	2443	2458	0.972693	0.003564245	196	0.365960943
ETHUSDT	SHIBUSDT	0.9610084	0.038224584	16.79414	0.82601954	1309	0.496638	2443	2262	0.976451	0.000358122	127	0.402914519
ETHUSDT	MATICUSDT	0.9937062	5.36E-05	7.351893	0.97023953	650	0.452173	2443	2409	0.978566	0.00585765	222	0.399800748
ETHUSDT	DOTUSDT	0.9896919	0.000211171	5.812364	0.86475747	627	0.457338	2443	2337	0.969781	0.047043943	265	0.384159393
SOLUSDT	SANDUSDT	0.9757009	0.01340567	3.581883	1.236073	1249	0.49093	2189	2348	0.972303	0.01674422	264	0.476676166
SOLUSDT	DOTUSDT	0.9754893	0.00921417	-1.05408	2.36107823	1024	0.456901	2189	2337	0.976617	0.006299494	203	0.448364459

XRPUSDT	LTCUSDT	0.9307645	0.009466734	-4.33233	0.75543985	887	0.467573	2458	2313	0.963166	0.000371804	143	0.41290319
XRPUSDT	MATICUSDT	0.9607277	0.005334074	-0.93972	0.55676921	718	0.461978	2458	2409	0.941651	0.019040362	305	0.396966554
XRPUSDT	SANDUSDT	0.9216145	0.033234406	-0.85518	0.25017108	1230	0.489312	2458	2348	0.893451	0.046807336	404	0.41314064
XRPUSDT	TRXUSDT	0.9228993	0.027425794	1.327833	0.81747858	1076	0.470953	2458	2366	0.960965	0.045964935	327	0.3609319
XRPUSDT	LUNCUSDT	0.9206609	0.000746128	6.2122	0.82970622	628	0.455164	2458	2152	0.911157	0.047501638	329	0.487713314
XRPUSDT	ATOMUSDT	0.9441097	0.015340669	-2.32569	0.54288203	843	0.482699	2458	2449	0.95057	0.038457694	288	0.424238225
DOGEUSDT	LINKUSDT	0.9894507	1.04E-05	-4.43101	1.02999131	346	0.43716	2461	2438	0.970357	0.020902141	274	0.476062287
SHIBUSDT	SANDUSDT	0.9554354	0.025113699	-11.2655	0.50836078	1281	0.499558	2262	2348	0.923862	0.018399461	306	0.476265149
LTCUSDT	LUNCUSDT	0.9439346	6.11E-06	13.52052	1.04810896	459	0.45575	2313	2152	0.898585	0.035002366	344	0.460178374

LTCUSDT	ATOMUSDT	0.9860043	6.48E-06	2.704504	0.69855683	308	0.445008	2313	2449	0.972147	0.036934736	247	0.433644579
ADAUSDT	LUNCUSDT	0.9069897	0.006078144	14.63134	1.81587011	976	0.460059	2448	2152	0.911301	0.029749061	315	0.483108852
NEARUSDT	DOTUSDT	0.9734777	0.020706084	-2.0762	1.61108142	1173	0.417172	2291	2337	0.97534	0.025362123	250	0.497927788
NEARUSDT	LUNCUSDT	0.8924869	0.015921581	22.90384	2.56666814	1161	0.452416	2291	2152	0.912679	0.036495069	334	0.483746637
NEARUSDT	JASMYUSDT	0.949622	0.017424605	6.752257	1.12852407	1414	0.461892	2291	2161	0.969908	0.005821125	232	0.478717711
SANDUSDT	DOTUSDT	0.9788357	0.010580473	-3.68584	1.87012334	1159	0.479766	2348	2337	0.969481	0.005071048	211	0.473043503
SANDUSDT	JASMYUSDT	0.9629483	0.004931443	6.623405	1.32108871	1176	0.467717	2348	2161	0.977519	0.003550831	198	0.454071303
LINKUSDT	LUNCUSDT	0.9479339	3.66E-05	11.09253	1.06470439	499	0.445872	2438	2152	0.896215	0.042820385	335	0.489877269

LINKUSDT	ATOMUSDT	0.9744389	0.005925526	0.132382	0.69833526	829	0.458095	2438	2449	0.976225	0.006245486	198	0.430383092
DOTUSDT	JASMYUSDT	0.9692598	0.000242036	5.455146	0.69599921	789	0.45428	2337	2161	0.976171	0.002365505	166	0.375040959
LUNCUSDT	ATOMUSDT	0.9480971	2.72E-06	-10.1716	0.6049385	423	0.449313	2152	2449	0.905348	0.029097134	334	0.489058257

Table.A9 : Models' performance per cryptocurrency

coin	model	training_time(seconds)	rmse	mse	mae	r2n(%)	mape (%)
ADAUSDT	transformer	371.8471	0.006858	4.70E-05	0.004887	47.51584975	1.294610894
ADAUSDT	lstm	2042.551	0.011161	0.000125	0.010631	-39.01348866	2.854574325
ATOMUSDT	transformer	2248.831	0.045757	0.002094	0.035109	94.81650735	0.265466567
ATOMUSDT	lstm	1179.941	0.272416	0.07421	0.260962	-83.73076401	1.97305992
BTCUSDT	transformer	366.683	1361.802	1854504	1303.288	-908.7786165	5.706865664
BTCUSDT	lstm	1636.799	1766.513	3120568	1726.288	-1597.468862	7.571858385
DOGEUSDT	transformer	742.3583	0.000559	3.12E-07	0.000514	89.83818424	0.59201052
DOGEUSDT	lstm	948.6503	0.001089	1.19E-06	0.000979	61.39297715	1.123216633
DOTUSDT	transformer	917.8116	0.041912	0.001757	0.038612	88.12958191	0.621676716

DOTUSDT	lstm	1435.343	0.109957	0.012091	0.103177	18.29767319	1.645508276
ETHUSDT	transformer	398.3177	43.01192	1850.025	38.89923	-463.673456	2.3622271
ETHUSDT	lstm	863.9059	83.07556	6901.549	81.67836	-2002.793406	4.975385921
JASMYUSDT	transformer	703.8028	1.80E-05	3.23E-10	1.36E-05	99.11051249	0.266356324
JASMYUSDT	lstm	1197.686	0.000135	1.82E-08	0.00013	49.77615027	2.52116777
LINKUSDT	transformer	1082.315	0.030971	0.000959	0.0255	92.4465907	0.364892165
LINKUSDT	lstm	1176.546	0.105747	0.011182	0.102145	11.94131779	1.466421641
LTCUSDT	transformer	570.9362	0.425353	0.180925	0.354218	88.63039802	0.397182559
LTCUSDT	lstm	825.3484	0.710305	0.504534	0.652035	68.29442483	0.730097265
LUNCUSDT	transformer	892.9332	4.95E-07	2.45E-13	4.23E-07	98.36548391	0.233823676
LUNCUSDT	lstm	845.485	1.53E-06	2.35E-12	1.45E-06	84.33154076	0.803174341
MATICUSDT	transformer	1239.571	0.002288	5.24E-06	0.00185	97.83231479	0.184272268
MATICUSDT	lstm	1783.227	0.016544	0.000274	0.015661	-13.3022724	1.5497403
NEARUSDT	transformer	964.1125	0.045789	0.002097	0.030326	75.08581195	1.248720452
NEARUSDT	lstm	1115.441	0.136537	0.018642	0.126913	-121.5270465	5.290063581
SANDUSDT	transformer	602.1865	0.052014	0.002705	0.047272	-214.0716979	5.887485919
SANDUSDT	lstm	1138.385	0.095359	0.009093	0.092773	-955.6305229	11.64303965
SHIBUSDT	transformer	1226.114	5.46E-08	2.98E-15	4.33E-08	93.35677833	0.361320903

SHIBUSDT	lstm	2932.92	9.52E-07	9.06E-13	9.37E-07	-1920.6949	7.74646568
SOLUSDT	transformer	572.6611	0.515995	0.266251	0.431809	79.32015302	1.729559172
SOLUSDT	lstm	997.7354	1.325182	1.756108	1.257084	-36.39796094	5.0309237
TRXUSDT	transformer	1018.385	0.000269	7.24E-08	0.000263	71.1156126	0.424139065
TRXUSDT	lstm	916.4727	0.000566	3.20E-07	0.000559	-27.76526693	0.901098837
XRPUSDT	transformer	338.0597	0.003913	1.53E-05	0.003249	11.37599438	0.793464067
XRPUSDT	lstm	1892.469	0.00561	3.15E-05	0.00535	-82.14167818	1.310617891

Table.A10 : Statistical properties of back-testing results

metric	algo_type_col	count	mean	std	min	25%	50%	75%	max
win_ratio	ai_algo	26	0.786916075	0.188129213	0.333333333	0.694231	0.83216783	0.9125	1
win_ratio	standard_algo	26	0.917742674	0.130818332	0.666666667	0.867857	1	1	1
max_drawdown	ai_algo	26	0.000141837	0.000190171	0	0	5.66E-05	0.00019116	0.000687535
max_drawdown	standard_algo	26	2.48E-06	1.26E-05	0	0	0	0	6.44E-05
avg_trade_duration	ai_algo	26	186.9721061	144.5621272	87.11764706	111.0195	129.681818	182.65	680
avg_trade_duration	standard_algo	26	257.1211303	113.4772048	95.8	144.3949	270.75	323	487.3333333
fees	ai_algo	26	7.099740548	3.592732441	1.520197034	4.826771	7.27315032	9.22085025	13.93312962
fees	standard_algo	26	4.347780893	2.878399417	1.470120949	1.978066	3.4213385	5.753294512	11.18761919
num_trades	ai_algo	26	9.153846154	5.065114472	2	5	9	12.75	21
num_trades	standard_algo	26	5.346153846	3.486457316	2	3	4	7	16
net_pnl	ai_algo	26	22.07498409	13.34365492	-6.540344124	15.51113	22.2738613	30.43688612	48.39823684
net_pnl	standard_algo	26	26.57854802	8.126199284	14.70445937	20.716	23.2708888	33.36233855	42.63518296
annualised_volatility	ai_algo	26	0.164060693	0.058738722	0.04128059	0.125055	0.15283488	0.209236847	0.271365408

annualised_volatility	standard_algo	26	0.165440971	0.104439615	0.043124795	0.09572	0.13751553	0.185762854	0.448391907
annualised_return	ai_algo	26	5.173285842	8.135526932	-0.157606224	0.677788	1.49640613	4.627953766	32.64229057
annualised_return	standard_algo	26	3.85323638	2.253166019	1.041733742	2.09222	3.61803707	4.987920882	9.092278832
sharpe_ratio	ai_algo	26	43.62205313	91.31844611	-1.329601266	4.592546	9.7725881	37.41247821	395.3895189
sharpe_ratio	standard_algo	26	32.52959506	24.06665197	3.953164669	9.181631	29.8799447	53.87908887	75.64214569

Appendix.B

Table.B11: Script_1

Jupyter notebook name	script_1
Script location	./SCRIPTS/
Input	- Binance API. - time interval = '1m' . - start and end dates of the period for which the market data are fetched.
functions	- Get market data from Binance API. - Save the data into an SQLite database
Run time	1 hour.
Output	- The database binance_prices3.db at the root of the project folder.

Table.B12: Script_2

Jupyter notebook name	script_2
Script location	./SCRIPTS/
functions	- Use the database binance_prices3.db to get market prices. - Produce the list of the most liquid coins in the in-sample period. - Produce the list of cointegrated pairs during this period. - Extract the statistical properties of the cointegrated pairs during the in-sample and out-of-sample periods. - Output and plot the actual prices for the coins forming the cointegrated pairs in the training/test sets. They will be used to train/test the DL models. - Calculate and output the corresponding actual spreads.
Run time	30 min
output	File containing the most liquid coins for the in-sample. Results were used in Table.A7 in the report. ex: - MOST_LIQUID_COINS/top_25th_market_caps_from_2023-10-01_to_2023-11-30.csv
	Temporary file containing all potential cointegrated pairs: Ex ./MOST_COINTEGRATED_PAIRS/cointegrated_pairs_2023-01-01_2023-06-30_2023-07-31.csv
	Refined list of cointegrated pairs used in Table.A8 in the report : ./MOST_COINTEGRATED_PAIRS/confirmed_cointegrated_pairs.csv

	Files representing the train/test sets and containing the actual mid prices of coins forming the pairs and their plots. They are used to train/test the DL models. Ex: - PRICES/ACTUAL/MID_plot_DOGEUSDT_MATICUSDT.png - PRICES/ACTUAL/training_set_MATICUSDT_2023-03-01_2023-08-31.csv - PRICES/ACTUAL/test_set_MATICUSDT_2023-08-31_2023-09-30.csv
	Files containing the actual spreads of the pairs and their plots. They are used to evaluate the predicted spreads. Ex: -SPREADS/ACTUAL/actual_spread_DOGEUSDT_MATICUSDT_2023-03-01_2023-09-30.csv -SPREADS/ACTUAL/actual_spread_plot_DOGEUSDT_MATICUSDT.png

Table.B13: Script_3

Jupyter notebook name	script_3
Script location	./SCRIPTS/
Functions	- Use the train/test sets produced by script_2. - Build the Transformer and LSTM models for each coin involved in a cointegrated pair. Save the models with h5 and pth extensions. - Predict and plot the prices of the coins forming the pairs over the out-of-sample periods.
Run time	12 hours
output	34 files containing the DL models. Ex: ./MODELS/transformer_model_TRXUSDT_2023-06-01_2023-11-30.pth ./MODELS/lstm_model_XRPUSDT_2023-06-01_2023-11-30.h5
	- Csv files containing Price predictions. - Plots displaying the predicted vs actual prices over the out-of-sample. They are used for the plots Figure.6, Figure.7, Figure.9, Figure.10, Figure.12, Figure.13, Figure.15, Figure.16 in the report. - The models' performance file (models_performance.csv) used in Table.A9 in the report. Ex: - PRICES/PREDICTED/transformers_predictions_DOGEUSDT_2023-08-31_2023-09-30.csv

	- PRICES/PREDICTED/transformers_predictions_DOGEUSDT_2023-08-31_2023-09-30.png - PRICES/PREDICTED/lstm_predictions_DOGEUSDT_2023-08-31_2023-08-31.png - PRICES/PREDICTED/lstm_predictions_DOGEUSDT_2023-08-31_2023-09-30.csv -PRICES/PREDICTED/models_performance.csv

Table.B14: Script_4

Jupyter notebook name	script_4
Script location	./SCRIPTS/
Functions	- Generate pair-trading signals based on predicted spreads, open and close transactions. - Generate pair-trading signals based on actual spreads, open and close transactions. - calculate portfolio results from the 2 trading signals. - calculate the predicted spreads. - compare and plots the actual vs predicted spreads. - calculate cointegration and correlation properties for the predicted spreads.
Run time	3 hours
Output	Backtesting results are output in sub-folders per cointegrated pair: - ai_algo_trading_data_* files contain the AI-APTS trading signals. - standard_algo_trading_data_* files contain the RB-APTS trading signals. - portfolio_ai_algo_* files contain the financial results for back-tested AI-APTS. - portfolio_standard_algo_* files contain the financial results for back-tested RB-APTS. Ex: ./BACKTESTING/DOGEUSDT_MATICUSDT_2023-03-01_2023-08-31_2023-09-30/ai_algo_trading_data_DOGEUSDT_MATICUSDT_2023-08-31_2023-09-30_360min_window_3min_delay.csv ./BACKTESTING/DOGEUSDT_MATICUSDT_2023-03-01_2023-08-31_2023-09-30/standard_algo_trading_data_DOGEUSDT_MATICUSDT_2023-08-31_2023-09-30_360min_window_3d_delay.csv ./BACKTESTING/DOGEUSDT_MATICUSDT_2023-03-01_2023-08-31_2023-09-30/portfolio_ai_algo_DOGEUSDT_MATICUSDT_2023-08-31_2023-09-30_360_min_window_3d_delay.csv

	./BACKTESTING/DOGEUSDT_MATICUSDT_2023-03-01_2023-08-31_2023-09-30/portfolio_standard_algo_DOGEUSDT_MATICUSDT_2023-08-31_2023-09-30_360_min_window_3d_delay.csv
	Aggregated results from the back-testing : ./BACKTESTING/back_testing_results.csv
	Outputs related to the actual and predicted spreads: - ./SPREADS/PREDICTED/predicted_spreads-statistical_properties.csv used to produce the final list of pairs eligible for PT. - ./SPREADS/PREDICTED/pred_vs_actual_error_metrics.csv used to produce the final list of pairs eligible for PT. - ./SPREADS/ACTUAL/actual_spreads-statistical_properties.csv. Plots of the predicted and actual spreads. They are used to produce Figure.5, Figure.8, Figure.11, and Figure.14 in the report. - Ex : ./SPREADS/PREDICTED//PREDICTED_VS_ACTUAL_SPREAD_DOGEUSDT_MATICUSDT_2023-08-31_2023-09-30_plot.png

Table.B15: Script_5

Jupyter notebook name	Script_5
Script location	./SCRIPTS/
Functions	- Get statistical properties from the back-testing results. - Get the statistical properties of models' performance. - Apply significance tests to back-testing and models' performance results.
Run time	5 min
Output	Csv files from significance and statistical tests are output in the folder ./EXPLORATIVE_DATA_ANALYSIS: - all_stats.csv is used to produce Table.A10 in the report. - all_stats_models.csv is used to produce Table.2 in the report. - average_trade_duration_distribution_plot.png - fees_distribution_plot.png - net_pnl_distribution_plot.png - number_of_trades_distribution_plot.png - one_sample_result.csv is used to produce Table.6 in the report. - one_sample_result_models.csv is used to produce Table.3 in the report. - scope.csv is used to produce Table.4 in the report. - sharpe_ratio_distribution_plot.png is used to produce Figure.17 in the report. - two_samples_result.csv is used to produce Table.5 in the report.

	- two_samples_result_models.csv is used to produce Table.1 in the report.

Table.B16: Dataset

Dataset name	SQLite database binance_prices.db
Dataset type	SQLite
Size	2.2 Go
Location in project folder	Project root.

The dataset can be found on Kaggle :

https://www.kaggle.com/datasets/anhtuanng/binance-prices

The Jupyter notebooks and other artefacts can be found on Github:

Datapyaddict/Comparative_Analysis_of_LSTM_and_Transformer_Models_for_HF_Cryptocurrency_Pairs_Trading_on_Binance: University of York (UK) / Msc in Computer Science With AI